Rudolph Frederick Stapelberg

Handbook of Reliability, Availability, Maintainability and Safety in Engineering Design

Springer

Handbook of Reliability, Availability, Maintainability and Safety in Engineering Design

Rudolph Frederick Stapelberg

Handbook of Reliability, Availability, Maintainability and Safety in Engineering Design

Springer

Rudolph Frederick Stapelberg, BScEng, MBA, PhD, DBA, PrEng
Adjunct Professor
Centre for Infrastructure and Engineering Management
Griffith University
Gold Coast Campus
Queensland
Australia

ISBN 978-1-84800-174-9 e-ISBN 978-1-84800-175-6

DOI 10.1007/978-1-84800-175-6

British Library Cataloguing in Publication Data
Stapelberg, Rudolph Frederick
Handbook of reliability, availability, maintainability and
safety in engineering design
1. Reliability (Engineering) 2. Maintainability
(Engineering) 3. Industrial safety
I. Title
620'.0045
ISBN-13: 9781848001749

Library of Congress Control Number: 2009921445

© 2009 Springer-Verlag London Limited

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms of licences issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publishers.

The use of registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant laws and regulations and therefore free for general use.

The publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions that may be made.

Cover design: eStudio Calamar S.L., Girona, Spain

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

Preface

In the past two decades, industry—particularly the process industry—has witnessed the development of several large 'super-projects', most in excess of a billion dollars. These large super-projects include the exploitation of mineral resources such as alumina, copper, iron, nickel, uranium and zinc, through the construction of huge complex industrial process plants. Although these super-projects create many thousands of jobs resulting in a significant decrease in unemployment, especially during construction, as well as projected increases in the wealth and growth of the economy, they bear a high risk in achieving their forecast profitability through maintaining budgeted costs. Most of the super-projects have either exceeded their budgeted establishment costs or have experienced operational costs far in excess of what was originally estimated in their feasibility prospectus scope. This has been the case not only with projects in the process industry but also with the development of infrastructure and high-technology projects in the petroleum and defence industries. The more significant contributors to the cost 'blow-outs' experienced by these projects can be attributed to the *complexity of their engineering design*, both in technology and in the complex integration of systems. These systems on their own are usually adequately designed and constructed, often on the basis of previous similar, though smaller designs.

It is the critical combination and complex integration of many such systems that give rise to *design complexity* and consequent frequent failure, where high risks of the integrity of engineering design are encountered. Research into this problem has indicated that large, expensive engineering projects may have quite superficial *design reviews*. As an essential control activity of engineering design, design review practices can take many forms. At the lowest level, they consist merely of an examination of engineering drawings and specifications before construction begins. At the highest level, they consist of comprehensive evaluations to ensure *due diligence*. Design reviews are included at different phases of the engineering design process, such as conceptual design, preliminary or schematic design, and final detail design. In most cases, though, a structured basis of measure is rarely used against which designs, or design alternatives, should be reviewed. It is obvious from many

examples of engineered installations that most of the problems stem from a lack of proper evaluation of their *engineering integrity*.

In determining the complexity and consequent frequent failure of the critical combination and complex integration of large engineering processes and systems, both in their level of technology as well as in their integration, the integrity of their design needs to be determined. This includes *reliability*, *availability*, *maintainability* and *safety* of the inherent process and system functions and their related equipment. Determining engineering design integrity implies determining reliability, availability, maintainability and safety *design criteria* of the design's inherent systems and related equipment. The tools that most design engineers resort to in determining integrity of design are techniques such as hazardous operations (HazOp) studies, and simulation. Less frequently used techniques include hazards analysis (HazAn), fault-tree analysis, failure modes and effects analysis (FMEA) and failure modes effects and criticality analysis (FMECA). Despite the vast amount of research already conducted, many of these techniques are either misunderstood or conducted incorrectly, or not even conducted at all, with the result that many high-cost super-projects eventually reach the construction phase without having been subjected to a rigorous and correct evaluation of the integrity of their designs.

Much consideration is being given to general engineering design, based on the theoretical expertise and practical experience of chemical, civil, electrical, electronic, industrial, mechanical and process engineers, from the point of view of '*what should be achieved*' to meet the design criteria. Unfortunately, it is apparent that not enough consideration is being given to '*what should be assured*' in the event the design criteria are not met. It is thus on this basis that many high-cost super-projects eventually reach the construction phase without having been subjected to a proper rigorous evaluation of the integrity of their designs. Consequently, research into a methodology for determining the integrity of engineering design has been initiated by the contention that not enough consideration is being given, in engineering design and design reviews, to *what should be assured* in the event of design criteria not being met. Many of the methods covered in this handbook have already been thoroughly explored by other researchers in the fields of reliability, availability, maintainability and safety analyses. What makes this compilation unique, though, is the combination of these methods and techniques in probability and possibility modelling, mathematical algorithmic modelling, evolutionary algorithmic modelling, symbolic logic modelling, artificial intelligence modelling, and object oriented computer modelling, in a logically structured approach to determining the integrity of engineering design.

This endeavour has encompassed not only a *depth of research* into the various methods and techniques—ranging from quantitative probability theory and expert judgement in Bayesian analysis, to qualitative possibility theory, fuzzy logic and uncertainty in Markov analysis, and from reliability block diagrams, fault trees, event trees and cause-consequence diagrams, to Petri nets, genetic algorithms and artificial neural networks—but also a *breadth of research* into the concept of integrity

in engineering design. Such breadth is represented by the topics of reliability and performance, availability and maintainability, and safety and risk, in an overall concept of *designing for integrity* during the engineering design process. These topics cover the integrity of engineering design not only for complex industrial processes and engineered installations but also for a wide range of engineering systems, from mobile to installed equipment.

This handbook is therefore written in the best way possible to appeal to:

1. Engineering design lecturers, for a comprehensive coverage of the subject theory and application examples, sufficient for addition to university graduate and postgraduate award courses.
2. Design engineering students, for sufficient theoretical coverage of the different topics with insightful examples and exercises.
3. Postgraduate research candidates, for use of the handbook as overall guidance and reference to other material.
4. Practicing engineers who want an easy readable reference to both theoretical and practical applications of the various topics.
5. Corporate organisations and companies (manufacturing, mining, engineering and process industries) requiring standard approaches to be understood and adopted throughout by their technical staff.
6. Design engineers, design organisations and consultant groups who require a 'best practice' handbook on the integrity of engineering design practice.

The topics covered in this handbook have proven to be much more of a research challenge than initially expected. The concept of design is both complex and complicated—even more so with engineering design, especially the design of engineering systems and processes that encompass all of the engineering disciplines. The challenge has been further compounded by focusing on applied and current methodology for determining the *integrity* of engineering design. Acknowledgement is thus gratefully given to those numerous authors whose techniques are presented in this handbook and also to those academics whose theoretical insight and critique made this handbook possible. The proof of the challenge, however, was not only to find solutions to the integrity problem in engineering design but also to be able to deliver some means of implementing these solutions in a practical computational format. This demanded an in-depth application of very many subjects ranging from mathematical and statistical modelling to symbolic and computational modelling, resulting in the need for research beyond the basic engineering sciences. Additionally, the solution models had to be tested in those very same engineering environments in which design integrity problems were highlighted. No one looks kindly upon criticism, especially with regard to allegations of shortcomings in their profession, where a high level of resistance to change is inevitable in respect of implementing new design tools such as AI-based blackboard models incorporating collaborative expert systems. Acknowledgement is therefore also gratefully given to those captains of industry who allowed this research to be

conducted in their companies, including all those design engineers who offered so much of their valuable time. Last but by no means least was the support and encouragement from my wife and family over the many years during which the topics in this handbook were researched and accumulated from a lifetime career in consulting engineering.

Rudolph Frederick Stapelberg

Contents

Part I Engineering Design Integrity Overview

1 Design Integrity Methodology 3
1.1 Designing for Integrity 4
1.1.1 Development and Scope of Design Integrity Theory 12
1.1.2 Designing for Reliability, Availability, Maintainability and Safety 14
1.2 Artificial Intelligence in Design 21
1.2.1 Development of Models and AIB Methodology 22
1.2.2 Artificial Intelligence in Engineering Design 25

2 Design Integrity and Automation 33
2.1 Industry Perception and Related Research 34
2.1.1 Industry Perception 34
2.1.2 Related Research 35
2.2 Intelligent Design Systems 37
2.2.1 The Future of Intelligent Design Systems 37
2.2.2 Design Automation and Evaluation Design Automation 38

Part II Engineering Design Integrity Application

3 Reliability and Performance in Engineering Design 43
3.1 Introduction 43
3.2 Theoretical Overview of Reliability and Performance in Engineering Design 45
3.2.1 Theoretical Overview of Reliability and Performance Prediction in Conceptual Design 60
3.2.2 Theoretical Overview of Reliability Assessment in Preliminary Design 72
3.2.3 Theoretical Overview of Reliability Evaluation in Detail Design 90

3.3 Analytic Development of Reliability and Performance in Engineering Design 107
3.3.1 Analytic Development of Reliability and Performance Prediction in Conceptual Design 107
3.3.2 Analytic Development of Reliability Assessment in Preliminary Design 133
3.3.3 Analytic Development of Reliability Evaluation in Detail Design 190
3.4 Application Modelling of Reliability and Performance in Engineering Design 241
3.4.1 The RAMS Analysis Application Model 242
3.4.2 Evaluation of Modelling Results 271
3.4.3 Application Modelling Outcome 285
3.5 Review Exercises and References 288

4 Availability and Maintainability in Engineering Design 295
4.1 Introduction 296
4.2 Theoretical Overview of Availability and Maintainability in Engineering Design 302
4.2.1 Theoretical Overview of Availability and Maintainability Prediction in Conceptual Design 308
4.2.2 Theoretical Overview of Availability and Maintainability Assessment in Preliminary Design 349
4.2.3 Theoretical Overview of Availability and Maintainability Evaluation in Detail Design 385
4.3 Analytic Development of Availability and Maintainability in Engineering Design 415
4.3.1 Analytic Development of Availability and Maintainability Prediction in Conceptual Design 416
4.3.2 Analytic Development of Availability and Maintainability Assessment in Preliminary Design 436
4.3.3 Analytic Development of Availability and Maintainability Evaluation in Detail Design 456
4.4 Application Modelling of Availability and Maintainability in Engineering Design 486
4.4.1 Process Equipment Models (PEMs) 486
4.4.2 Evaluation of Modelling Results 500
4.4.3 Application Modelling Outcome 518
4.5 Review Exercises and References 520

5 Safety and Risk in Engineering Design 529
5.1 Introduction 530
5.2 Theoretical Overview of Safety and Risk in Engineering Design 537
5.2.1 Forward Search Techniques for Safety in Engineering Design 541
5.2.2 Theoretical Overview of Safety and Risk Prediction in Conceptual Design 588
5.2.3 Theoretical Overview of Safety and Risk Assessment in Preliminary Design 607
5.2.4 Theoretical Overview of Safety and Risk Evaluation in Detail Design 627
5.3 Analytic Development of Safety and Risk in Engineering Design . . . 676
5.3.1 Analytic Development of Safety and Risk Prediction in Conceptual Design 678
5.3.2 Analytic Development of Safety and Risk Assessment in Preliminary Design 687
5.3.3 Analytic Development of Safety and Risk Evaluation in Detail Design 702
5.4 Application Modelling of Safety and Risk in Engineering Design 725
5.4.1 Artificial Intelligence-Based (AIB) Blackboard Model 726
5.4.2 Evaluation of Modelling Results 776
5.4.3 Application Modelling Outcome 790
5.5 Review Exercises and References 791

A Design Engineer's Scope of Work 799

B Bibliography of Selected Literature 807

Index 811

List of Figures

1.1 Layout of the RAM analysis model 24
1.2 Layout of part of the OOP simulation model 25
1.3 Layout of the AIB blackboard model 26

3.1 Reliability block diagram of two components in series 48
3.2 Reliability of a high-speed self-lubricated reducer 49
3.3 Reliability block diagram of two components in parallel 50
3.4 Combination of series and parallel configuration 51
3.5 Reduction of combination system configuration 51
3.6 Power train system reliability of a haul truck (Komatsu Corp., Japan) 53
3.7 Power train system diagram of a haul truck 53
3.8 Reliability of groups of series components 55
3.9 Example of two parallel components 56
3.10 Reliability of groups of parallel components 57
3.11 Slurry mill engineered installation 57
3.12 Total cost versus design reliability 61
3.13 Stress/strength diagram 66
3.14 Interaction of load and strength distributions (Carter 1986) 68
3.15 System transition diagram 74
3.16 Risk as a function of time and stress 77
3.17 Criticality matrix (Dhillon 1999) 83
3.18 Simple fault tree of cooling water system 87
3.19 Failure hazard curve (life characteristic curve or risk profile) 92
3.20 Shape of the Weibull density function, $F(t)$, for different values of β 100
3.21 The Weibull graph chart for different percentage values of the failure distribution 101
3.22 Parameter profile matrix 108
3.23 Determination of a data point: two limits 109
3.24 Determination of a data point: one upper limit 109
3.25 Determination of a data point: one lower limit 110
3.26 Two-variable parameter profile matrix 112

3.27 Possibility distribution of *young* 152
3.28 Possibility distribution of *somewhat young* 152
3.29 Values of linguistic variable *pressure* 160
3.30 Simple crisp inference 167
3.31 **a** Basic property $A' = A$. **b** Basic property $B' = B$ 168
3.32 **a**, **b** Total indeterminance 169
3.33 **a**, **b** Subset property 169
3.34 Effects of λ on the probability density function 199
3.35 Effects of λ on the reliability function 199
3.36 Example exponential probability graph 203
3.37 Weibull p.d.f. with $0 < \beta < 1$, $\beta = 1$, $\beta > 1$ and a fixed μ (ReliaSoft Corp.) 205
3.38 Weibull c.d.f. or unreliability vs. time (ReliaSoft Corp.) 206
3.39 Weibull 1–c.d.f. or reliability vs. time (ReliaSoft Corp.) 206
3.40 Weibull failure rate vs. time (ReliaSoft Corp.) 207
3.41 Weibull p.d.f. with $\mu = 50$, $\mu = 100$, $\mu = 200$ (ReliaSoft Corp.) 208
3.42 Plot of the Weibull density function, $F(t)$, for different values of β . . 210
3.43 Minimum life parameter and true MTBF 212
3.44 Revised Weibull chart 213
3.45 Theories for representing uncertainty distributions (Booker et al. 2000) 217
3.46 Methodology of combining available information 225
3.47 Baselines of an engineering design project 230
3.48 Tracking reliability uncertainty (Booker et al. 2000) 239
3.49 Component condition sets for membership functions 240
3.50 Performance-level sets for membership functions 240
3.51 Database structuring of SBS into dynasets 245
3.52 Initial structuring of plant/operation/section 247
3.53 Front-end selection of plant/operation/section: RAMS analysis model spreadsheet, process flow, and treeview 248
3.54 Global grid list (spreadsheet) of systems breakdown structuring 249
3.55 Graphics of selected section PFD 251
3.56 Graphics of selected section treeview (cascaded systems structure) . . 252
3.57 Development list options for selected PFD system 253
3.58 Overview of selected equipment specifications 254
3.59 Overview of the selected equipment technical data worksheet 255
3.60 Overview of the selected equipment technical specification document 256
3.61 Analysis of development tasks for the selected system 257
3.62 Analysis of selected systems functions 258
3.63 Functions analysis worksheet of selected component 259
3.64 Specifications of selected major development tasks 260
3.65 Specifications worksheet of selected equipment 261
3.66 Diagnostics of selected major development tasks 262
3.67 Hazards criticality analysis assembly condition 263

3.68 Hazards criticality analysis component condition 264
3.69 Hazards criticality analysis condition diagnostic worksheet 265
3.70 Hazards criticality analysis condition spreadsheet 266
3.71 Hazards criticality analysis criticality worksheet 267
3.72 Hazards criticality analysis criticality spreadsheet 268
3.73 Hazards criticality analysis strategy worksheet 269
3.74 Hazards criticality analysis strategy spreadsheet 270
3.75 Hazards criticality analysis costs worksheet 271
3.76 Hazards criticality analysis costs spreadsheet 272
3.77 Hazards criticality analysis logistics worksheet 273
3.78 Hazards criticality analysis logistics spreadsheet 274
3.79 Typical data accumulated by the installation's DCS 275
3.80 Design specification FMECA—drying tower 280
3.81 Design specification FMECA—hot gas feed 281
3.82 Design specification FMECA—reverse jet scrubber 282
3.83 Design specification FMECA—final absorption tower 283
3.84 Weibull distribution chart for failure data 285
3.85 Monte Carlo simulation spreadsheet results for a gamma distribution best fit of TBF data 287

4.1 Breakdown of total system's equipment time (DoD 3235.1-H 1982) where UP TIME = operable time, DOWN TIME = inoperable time, OT = operating time, ST = standby time, ALDT = administrative and logistics downtime, TPM = total preventive maintenance and TCM = total corrective maintenance . . . 297
4.2 Regression equation of predicted repair time in nomograph form . . . 308
4.3 Three-system parallel configuration system 311
4.4 Life-cycle costs structure 318
4.5 Cost minimisation curve for non-recurring and recurring LCC 321
4.6 Design effectiveness and life-cycle costs (Barringer 1998) 327
4.7 Markov model state space diagram 350
4.8 Multi-state system transition 352
4.9 Operational availability time-line model—generalised format (DoD 3235.1-H 1982) 389
4.10 Operational availability time-line model—recovery time format (DoD 3235.1-H 1982) 390
4.11 A comparison of downtime and repair time (Smith 1981) 404
4.12 Example of a simple power-generating plant 411
4.13 Parameter profile matrix 418
4.14 Simulation-based design model from two different disciplines (Du et al. 1999c) 430
4.15 Flowchart for the extreme condition approach for uncertainty analysis (Du et al. 1999c) 431
4.16 Flowchart of the Monte Carlo simulation procedure (Law et al. 1991) 433

4.17 Propagation and mitigation strategy of the effect of uncertainties (Parkinson et al. 1993) 436
4.18 Translation of a flowchart to a Petri net (Peterson 1981) 438
4.19 Typical graphical representation of a Petri net (Lindemann et al. 1999) 440
4.20 Illustrative example of an MSPN for a fault-tolerant process system (Ajmone Marsan et al. 1995) 444
4.21 MSPN for a process system based an a queuing client-server paradigm (Ajmone Marson et al. 1995) 446
4.22 Extended reachability graph generated from the MSPN model (Ajmone Marsan et al. 1995) 446
4.23 Reduced reachability graph generated from the MSPN model 448
4.24 MRSPN model for availability with preventive maintenance (Bobbio et al. 1997) 453
4.25 MRSPN model results for availability with preventive maintenance . 455
4.26 Models of closed and open systems 462
4.27 Coal gas production and clarifying plant schematic block diagram . . 464
4.28 **a** Series reliability block diagram. **b** Series reliability graph 467
4.29 **a** Parallel reliability block diagram. **b** Parallel reliability graph 467
4.30 Process flow block diagram 468
4.31 Availability block diagram (ABD) 469
4.32 Simple power plant schematic process flow diagram 469
4.33 Power plant process flow diagram systems cross connections 470
4.34 Power plant process flow diagram sub-system grouping 471
4.35 Simple power plant subgroup capacities 472
4.36 Process block diagram of a turbine/generator system 479
4.37 Availability block diagram of a turbine/generator system, where A = availability, MTBF = mean time between failure (h), MTTR = mean time to repair (h) 479
4.38 Example of defined computer automated complexity (Tang et al. 2001) 483
4.39 Logistic function of complexity vs. complicatedness (Tang et al. 2001) 484
4.40 Blackboard model and the process simulation model 488
4.41 Systems selection in the blackboard model 489
4.42 Design equipment list data in the blackboard model 490
4.43 Systems hierarchy in the blackboard model context 491
4.44 User interface in the blackboard model 492
4.45 Dynamic systems simulation in the blackboard model 493
4.46 General configuration of process simulation model 495
4.47 Composition of systems of process simulation model 496
4.48 PEM library and selection for simulation modelling 497
4.49 Running the simulation model 499
4.50 Simulation model output results 500
4.51 Process flow diagram for simulation model sector 1 504

4.52 Design details for simulation model sector 1: logical flow initiation . . . 505
4.53 Design details for simulation model sector 1: logical flow storage PEMs . . . 506
4.54 Design details for simulation model sector 1: output performance results . . . 507
4.55 Simulation output for simulation model sector 1 . . . 508
4.56 Process flow diagram for simulation model sector 2 . . . 510
4.57 Design details for simulation model sector 2: holding tank process design specifications . . . 511
4.58 Design details for simulation model sector 2: output performance results . . . 512
4.59 Simulation output for simulation model sector 2 . . . 514
4.60 Process flow diagram for simulation model sector 3 . . . 517
4.61 Design details for simulation model sector 3: process design specifications . . . 518
4.62 Design details for simulation model sector 3: output performance results . . . 519
4.63 Simulation output for simulation model sector 3 . . . 520

5.1 Fault-tree analysis . . . 542
5.2 Event tree . . . 543
5.3 Cause-consequence diagram . . . 544
5.4 Logic and event symbols used in FTA . . . 546
5.5 Safety control of cooling water system . . . 548
5.6 Outage cause investigation logic tree expanded to potential root cause areas . . . 554
5.7 Root cause factors for the systems and equipment design area . . . 554
5.8 Factor tree for origin of design criteria . . . 555
5.9 Event tree for a dust explosion (IEC 60300-3-9) . . . 558
5.10 Event tree branching for reactor safety study . . . 562
5.11 Event tree with boundary conditions . . . 563
5.12 Event tree with fault-tree linking . . . 564
5.13 Function event tree for loss of coolant accident in nuclear reactor (NUREG 75/014 1975) . . . 566
5.14 Example cause-consequence diagram . . . 568
5.15 Structure of the cause-consequence diagram . . . 569
5.16 Redundant decision box . . . 570
5.17 Example fault tree indicating system failure causes . . . 571
5.18 Cause-consequence diagram for a three-component system . . . 572
5.19 Reduced cause-consequence diagram . . . 573
5.20 BDD with variable ordering $A < B < C$ . . . 573
5.21 Example of part of a cooling water system . . . 602
5.22 Fault tree of dormant failure of a high-integrity protection system (HIPS; Andrews 1994) . . . 620

5.23 Schematic of a simplified high-pressure protection system 625
5.24 Typical logic event tree for nuclear reactor safety (NUREG-751014 1975) 630
5.25 Risk curves from nuclear safety study (NUREG 1150 1989) Appendix VI WASH 1400: c.d.f. for early fatalities 631
5.26 Simple RBD construction 636
5.27 Layout of a complex RBD (NASA 1359 1994) 637
5.28 Example RBD 638
5.29 RBD to fault tree transformation 639
5.30 Fault tree to RBD transformation 640
5.31 Cut sets and path sets from a complex RBD 641
5.32 Transform of an event tree into an RBD 641
5.33 Transform of an RBD to a fault tree 642
5.34 High-integrity protection system (HIPS) 644
5.35 Cause-consequence diagram for HIPS system (Ridley et al. 1996) . . 645
5.36 Combination fault trees for cause-consequence diagram 646
5.37 Modified cause-consequence diagram for HIPS system (Ridley et al. 1996) 647
5.38 Combination fault trees for modified cause-consequence diagram . . . 648
5.39 Final cause-consequence diagram for HIPS system (Ridley et al. 1996) 649
5.40 Combination fault trees for the final cause-consequence diagram (Ridley et al. 1996) 650
5.41 **a** Kaplan–Meier survival curve for rotating equipment, **b** estimated hazard curve for rotating equipment 655
5.42 **a** Risk exposure pattern for rotating equipment, **b** risk-based maintenance patterns for rotating equipment 656
5.43 Typical cost optimisation curve 657
5.44 Probability distribution definition with @RISK (Palisade Corp., Newfield, NY) 675
5.45 Schema of a conceptual design space 679
5.46 Selecting design objects in the design knowledge base 682
5.47 Conceptual design solution of the layout of a gas cleaning plant 683
5.48 Schematic design model of the layout of a gas cleaning plant 683
5.49 Detail design model of the scrubber in the layout of a gas cleaning plant 684
5.50 Fault-tree structure for safety valve selection (Pattison et al. 1999) . . 695
5.51 Binary decision diagram (BDD) for safety valve selection 696
5.52 High-integrity protection system (HIPS): example of BDD application 697
5.53 Schematic layout of a complex artificial neural network (Valluru 1995) 705
5.54 The building blocks of artificial neural networks, where σ is the non-linearity, x_i the output of unit i, x_j the input to unit j, and w_{ij} are the weights that connect unit i to unit j 705

5.55 Detailed view of a processing element (PE) 705
5.56 A fully connected ANN, and its weight matrix 706
5.57 Multi-layer perceptron structure 706
5.58 Weight matrix structure for the multi-layer perception 707
5.59 Basic structure of an artificial neural network 707
5.60 Input connections of the artificial perceptron (a_n, b_1) 708
5.61 The binary step-function threshold logic unit (TLU) 708
5.62 The non-binary sigmoid-function threshold logic unit (TLU) 709
5.63 Boolean-function input connections of the artificial perceptron (a_n, o_0) 710
5.64 Boolean-function pattern space and TLU of the artificial perceptron (a_n, o_0) 710
5.65 The gradient descent technique 711
5.66 Basic structure of an artificial neural network: back propagation 712
5.67 Graph of membership function transformation of a fuzzy ANN 714
5.68 A fuzzy artificial perceptron (AP) 715
5.69 Three-dimensional plots generated from a neural network model illustrating the relationship between speed, load, and wear rate (Fusaro 1998) 716
5.70 Comparison of actual data to those of an ANN model approximation (Fusaro 1998) 716
5.71 Example failure data using cusum analysis (Ilott et al. 1997) 718
5.72 Topology of the example ANN (Ilott et al. 1997) 719
5.73 **a)** An example fuzzy membership functions for pump motor current (Ilott et al. 1995), **b)** example fuzzy membership functions for pump pressure (Ilott et al. 1995) 720
5.74 Convergence rate of ANN iterations 721
5.75 Standard back-propagation ANN architecture (Schocken 1994) 723
5.76 Jump connection back-propagation ANN architecture (Schocken 1994) 723
5.77 Recurrent back-propagation with dampened feedback ANN architecture (Schocken 1994) 723
5.78 Ward back propagation ANN architecture (Schocken 1994) 724
5.79 Probabilistic (PNN) ANN architecture (Schocken 1994) 724
5.80 General regression (GRNN) ANN architecture (Schocken 1994) 724
5.81 Kohonen self-organising map ANN architecture (Schocken 1994) . . 724
5.82 AIB blackboard model for engineering design integrity (ICS 2003) . 728
5.83 AIB blackboard model with systems modelling option 729
5.84 Designing for safety using systems modelling: system and assembly selection 730
5.85 Designing for safety using systems modelling 731
5.86 Treeview of systems hierarchical structure 732
5.87 Technical data sheets for modelling safety 733
5.88 Monte Carlo simulation of RBD and FTA models 734
5.89 FTA modelling in designing for safety 736

5.90 Weibull cumulative failure probability graph of HIPS 737
5.91 Profile modelling in designing for safety 738
5.92 AIB blackboard model with system simulation option 739
5.93 PFD for simulation modelling 740
5.94 PEMs for simulation modelling 741
5.95 PEM simulation model performance variables for process information 742
5.96 PEM simulation model graphical display of process information 743
5.97 Petri net-based optimisation algorithms in system simulation 744
5.98 AIB blackboard model with CAD data browser option 745
5.99 Three-dimensional CAD integrated model for process information . . 746
5.100 CAD integrated models for process information 747
5.101 ANN computation option in the AIB blackboard 748
5.102 ANN NeuralExpert problem selection 749
5.103 ANN NeuralExpert example input data attributes 750
5.104 ANN NeuralExpert sampling and prediction 751
5.105 ANN NeuralExpert sampling and testing 752
5.106 ANN NeuralExpert genetic optimisation 753
5.107 ANN NeuralExpert network complexity 754
5.108 Expert systems functional overview in the AIB blackboard knowledge base 755
5.109 Determining the conditions of a process 756
5.110 Determining the failure effect on a process 757
5.111 Determining the risk of failure on a process 758
5.112 Determining the criticality of consequences of failure 759
5.113 Assessment of design problem decision logic 760
5.114 AIB blackboard knowledge-based expert systems 761
5.115 Knowledge base facts frame in the AIB blackboard 762
5.116 Knowledge base conditions frame slot 763
5.117 Knowledge base hierarchical data frame 764
5.118 The Expert System blackboard and goals 765
5.119 Expert System questions factor—temperature 766
5.120 Expert System multiple-choice question editor 767
5.121 Expert System branched decision tree 768
5.122 Expert System branched decision tree: nodes 769
5.123 Expert System rules of the knowledge base 770
5.124 Expert System rule editor 771
5.125 Testing and validating Expert System rules 772
5.126 Fuzzy logic for managing uncertain data 774
5.127 AIB blackboard model with plant analysis overview option 775
5.128 Automated continual design review: component SBS 776
5.129 Automated continual design review: component criticality 777

List of Tables

3.1 Reliability of a high-speed self-lubricated reducer 49
3.2 Power train system reliability of a haul truck 54
3.3 Component and assembly reliabilities and system reliability of slurry mill engineered installation 58
3.4 Failure detection ranking 81
3.5 Failure mode occurrence probability 81
3.6 Severity of the failure mode effect 82
3.7 Failure mode effect severity classifications 83
3.8 Qualitative failure probability levels 83
3.9 Failure effect probability guideline values 84
3.10 Labelled intervals for specific performance parameters 131
3.11 Parameter interval matrix 131
3.12 Fuzzy term *young* 151
3.13 Modifiers (hedges) and linguistic expressions 152
3.14 Truth table applied to propositions 163
3.15 Extract from FMECA worksheet of quantitative RAM analysis field study: RJS pump no. 1 assembly 181
3.16 Extract from FMECA worksheet of quantitative RAM analysis field study: motor RJS pump no. 1 component 183
3.17 Extract from FMECA worksheet of quantitative RAM analysis field study: MCC RJS pump no. 1 component 185
3.18 Extract from FMECA worksheet of quantitative RAM analysis field study: RJS pump no. 1 control valve component 186
3.19 Extract from FMECA worksheet of quantitative RAM analysis field study: RJS pump no. 1 instrument loop (pressure) assembly 187
3.20 Uncertainty in the FMECA of a critical control valve 188
3.21 Uncertainty in the FMECA of critical pressure instruments 189
3.22 Median rank table for failure test results 200
3.23 Median rank table for Bernard's approximation 202
3.24 Acid plant failure modes and effects analysis (ranking on criticality) . 276
3.25 Acid plant failure modes and effects criticality analysis 279

3.26 Acid plant failure data (repair time RT and time before failure TBF) . . 284
3.27 Total downtime of the environmental plant critical systems 286
3.28 Values of distribution models for time between failure 286
3.29 Values of distribution models for repair time 287

4.1 Double turbine/boiler generating plant state matrix 412
4.2 Double turbine/boiler generating plant partial state matrix 413
4.3 Distribution of the tokens in the reachable markings 447
4.4 Power plant partitioning into sub-system grouping 471
4.5 Process capacities per subgroup 473
4.6 Remaining capacity versus unavailable subgroups 474
4.7 Flow capacities and state definitions of unavailable subgroups 474
4.8 Flow capacities of unavailable sub-systems per sub-system group . . . 475
4.9 Unavailable sub-systems and flow capacities per sub-system group . . 475
4.10 Unavailable sub-systems and flow capacities per sub-system group: final summary 475
4.11 Unavailable subgroups and flow capacities incidence matrix 477
4.12 Probability of incidence of unavailable systems and flow capacities . . 477
4.13 Sub-system/assembly integrity values of a turbine/generator system . 480
4.14 Preliminary design data for simulation model sector 1 503
4.15 Comparative analysis of preliminary design data and simulation output data for simulation model sector 1 507
4.16 Acceptance criteria of simulation output data, with preliminary design data for simulation model sector 1 508
4.17 Preliminary design data for simulation model sector 2 509
4.18 Comparative analysis of preliminary design data and simulation output data for simulation model sector 2 513
4.19 Acceptance criteria of simulation output data, with preliminary design data for simulation model sector 2 515
4.20 Preliminary design data for simulation model sector 3 516
4.21 Comparative analysis of preliminary design data and simulation output data for simulation model sector 3 516
4.22 Acceptance criteria of simulation output data, with preliminary design data for simulation model sector 3 521

5.1 Hazard severity ranking (MIL-STD-882C 1993) 539
5.2 Sample HAZID worksheet 540
5.3 Categories of hazards relative to various classifications of failure 540
5.4 Cause-consequence diagram symbols and functions 569
5.5 Standard interpretations for process/chemical industry guidewords . . . 578
5.6 Matrix of attributes and guideword interpretations for mechanical systems 579
5.7 Risk assessment scale 585
5.8 Initial failure rate estimates 586
5.9 Operational primary keywords 600

5.10 Operational secondary keywords: standard HazOp guidewords 601
5.11 Values of the Q-matrix 612
5.12 Upper levels of systems unreliability due to CCF 623
5.13 Analysis of valve data to determine CCF beta factor 626
5.14 Sub-system component reliability bands 638
5.15 Component functions for HIPS system 644
5.16 Typical FMECA for process criticality 658
5.17 FMECA with preventive maintenance activities 659
5.18 FMECA for cost criticality 663
5.19 FMECA for process and cost criticality 665
5.20 Risk assessment scale 667
5.21 Qualitative risk-based FMSE for process criticality, where (1)=likelihood of occurrence (%), (2)=severity of the consequence (rating), (3)=risk (probability×severity), (4)=failure rate (1/MTBF), (5)=criticality (risk×failure rate) 668
5.22 FMSE for process criticality using residual life 674
5.23 Fuzzy and induced preference predicates 680
5.24 Required design criteria and variables 697
5.25 GA design criteria and variables results 701
5.26 Boolean-function input values of the artificial perceptron (a_n, o_0) 710
5.27 Simple 2-out-of-4 vote arrangement truth table 735
5.28 The AIB blackboard data object construct 785
5.29 Computation of $\Gamma_{j,k}$ and $\theta_{j,k}$ for blackboard B1 787
5.30 Computation of non-zero $\Omega_{j,k}, \Sigma_{j,k}$ and $\Pi_{j,k}$ for blackboard B1 787
5.31 Computation of $\Gamma_{j,k}$ and $\theta_{j,k}$ for blackboard B2 789
5.32 Computation of non-zero $\Omega_{j,k}, \Sigma_{j,k}$ and $\Pi_{j,k}$ for blackboard B2 789

Part I
Engineering Design Integrity Overview

Chapter 1
Design Integrity Methodology

Abstract In the design of critical combinations and complex integrations of large engineering systems, their *engineering integrity* needs to be determined. Engineering integrity includes *reliability*, *availability*, *maintainability* and *safety* of inherent systems functions and their related equipment. The *integrity of engineering design* therefore includes the *design criteria* of reliability, availability, maintainability and safety of systems and equipment. The overall combination of these four topics constitutes a methodology that ensures good engineering design with the desired engineering integrity. This methodology provides the means by which complex engineering designs can be properly analysed and reviewed, and is termed a RAMS analysis. The concept of RAMS analysis is not new and has been progressively developed, predominantly in the field of product assurance. Much consideration is being given to engineering design based on the theoretical expertise and practical experiences of chemical, civil, electrical, electronic, industrial, mechanical and process engineers, particularly from the point of view of '*what should be achieved*' to meet design criteria. Unfortunately, not enough consideration is being given to '*what should be assured*' in the event design criteria are not met. Most of the problems encountered in engineered installations stem from the lack of a proper evaluation of their *design integrity*. This chapter gives an overview of methodology for determining the integrity of engineering design to ensure that consideration is given to '*what should be assured*' through appropriate design review techniques. Such design review techniques have been developed into automated continual design reviews through intelligent computer automated methodology for determining the integrity of engineering design. This chapter thus also introduces the application of artificial intelligence (AI) in engineering design and gives an overview of artificial intelligence-based (AIB) modelling in designing for reliability, availability, maintainability and safety to provide a means for continual design reviews throughout the engineering design process. These models include a RAM analysis model, a dynamic systems simulation blackboard model, and an artificial intelligence-based (AIB) blackboard model.

R.F. Stapelberg, *Handbook of Reliability, Availability, Maintainability and Safety in Engineering Design*, © Springer 2009

1.1 Designing for Integrity

In the past two decades, industry, and particularly the process industry, has witnessed the development of large super-projects, most in excess of a billion dollars. Although these super-projects create many thousands of jobs resulting in significant decreases in unemployment, especially during construction, as well as projected increases in the wealth and growth of the economy, they bear a high risk in achieving their forecast profitability through maintaining budgeted costs. Because of the *complexity of design* of these projects, and the fact that most of the problems encountered in the projects stem from a lack of proper evaluation of their *integrity of design*, it is expected that research in this field should arouse significant interest within most engineering-based industries in general. Most of the super-projects researched by the author have either exceeded their budgeted establishment costs or have experienced operational costs far in excess of what was originally estimated in their feasibility prospectus scope. The poor performances of these projects are given in the following points that summarise the findings of this research:

- In all of the projects studied, additional funding had to be obtained for cost overruns and to cover shortfalls in working capital due to extended construction and commissioning periods. Final capital costs far exceeded initial feasibility estimates. Additional costs were incurred mainly for rectification of insufficiently designed system circuits and equipment, and increased engineering and maintenance costs. Actual construction completion schedule overruns averaged 6 months, and commissioning completion schedule overruns averaged 11 months. Actual start-up commenced $+1$ year after forecast with all the projects.
- Estimated cash operating costs were over-optimistic and, in some cases, no further cash operating costs were estimated due to project schedule overruns as well as over-extended ramp-up periods in attempts to obtain design forecast output.
- Technology and engineering problems were numerous in all the projects studied, especially in the various process areas, which indicated insufficient design and/or specifications to meet the inherent process problems of corrosion, scaling and erosion.
- Procurement and construction problems were experienced by all the projects studied, especially relating to the lack of design data sheets, incomplete equipment lists, inadequate process control and instrumentation, incorrect spare parts lists, lack of proper identification of spares and facilities equipment such as manual valves and piping both on design drawings and on site, and basic quality 'corner cutting' resulting from cost and project overruns. Actual project schedule overruns averaged $+1$ year after forecast.
- Pre-commissioning as well as commissioning schedules were over-optimistic in most cases where actual commissioning completion schedule overruns averaged 11 months. Inadequate references to equipment data sheets and design specifications resulted in it later becoming an exercise of identifying as-built equipment, rather than of confirming equipment installation with design specifications.

- The need to rectify processes and controls occurred in all the projects because of detrimental erosion and corrosion effects on all the equipment with design and specification inadequacies, resulting in cost and time overruns. Difficulties with start-ups after resulting forced stoppages, and poor systems performance with regard to availability and utilisation resulted in longer ramp-up periods and shortfalls of operating capital to ensure proper project handover.
- In all the projects studied, schedules were over-optimistic with less than optimum performance being able to be reached only much later than forecast. Production was much lower than envisaged, ranging from 10 to 60% of design capacity 12 months after the forecast date that design capacity would be reached. Problems with regard to achieving design throughput occurred in all the projects. This was due mainly to low plant utilisation because of poor process and equipment design reliability, and short operating periods.
- Project management and control problems relating to construction, commissioning, start-up and ramp-up were proliferate as a result of an inadequate assessment of design complexity and project volume with regard to the many integrated systems and equipment.

It is obvious from the previous points, made available in the public domain through published annual reports of real-world examples of recently constructed engineering projects, that most of the problems stem from a lack of proper evaluation of their *engineering integrity*. The important question to be considered therefore is:

What does integrity of engineering design actually imply?

Engineering Integrity

In determining the complexity and consequent frequent failure of the critical combination and complex integration of large engineering processes, both in technology as well as in the integration of systems, their *engineering integrity* needs to be determined. This engineering integrity includes *reliability*, *availability*, *maintainability* and *safety* of the inherent process systems functions and their related equipment. Integrity of *engineering design* therefore includes the *design criteria* of *reliability*, *availability*, *maintainability* and *safety* of these systems and equipment.

Reliability can be regarded as the probability of successful operation or *performance* of systems and their related equipment, with minimum risk of loss or disaster or of *system failure*. Designing for reliability requires an evaluation of the *effects of failure* of the inherent systems and equipment.

Availability is that aspect of system reliability that takes equipment *maintainability* into account. Designing for availability requires an evaluation of the *consequences of unsuccessful operation or performance* of the integrated systems, and the critical requirements necessary to restore operation or performance to design expectations.

Maintainability is that aspect of maintenance that takes *downtime* of the systems into account. Designing for maintainability requires an evaluation of the *accessi-*

bility and '*repairability*' of the inherent systems and their related equipment in the event of failure, as well as of integrated systems shutdown during planned maintenance.

Safety can be classified into three categories, one relating to *personal protection*, another relating to *equipment protection*, and yet another relating to *environmental protection*. Safety in this context may be defined as "not involving risk", where risk is defined as "the chance of loss or disaster". Designing for safety is inherent in the development of designing for reliability and maintainability of systems and their related equipment. *Environmental protection* in engineering design, particularly in industrial process design, relates to the prevention of failure of the inherent process systems resulting in environmental problems associated predominantly with the treatment of wastes and emissions from chemical processing operations, high-temperature processes, hydrometallurgical and mineral processes, and processing operations from which by-products are treated.

The overall combination of these four topics constitutes a methodology that ensures good engineering design with the desired engineering integrity. This methodology provides the means by which complex engineering designs can be properly analysed and reviewed. Such an analysis and review is conducted not only with a focus upon individual inherent systems but also with a perspective of the critical combination and complex integration of all the systems and related equipment, in order to achieve the required reliability, availability, maintainability and safety (i.e. integrity).

This analysis is often termed a *RAMS analysis*. The concept of RAMS analysis is not new and has been progressively developed over the past two decades, predominantly in the field of *product assurance*. Those industries applying product assurance methods have unquestionably witnessed astounding revolutions of knowledge and techniques to match the equally astounding progress in technology, particularly in the electronic, micro-electronic and computer industries. Many technologies have already originated, attained peak development, and even become obsolete within the past two decades. In fact, most systems of products built today will be long since obsolete by the time they wear out. So, too, must the development of ideas, knowledge and techniques to adequately manage the application and maintenance of newly developed systems be compatible *and* adaptable, or similarly become obsolete and fall into disuse. This applies to the concept of engineering integrity, particularly to the integrity of engineering design.

Engineering knowledge and techniques in the design and development of complex systems either must become part of a new information revolution in which compatible and, in many cases, more stringent methods of design reviews and evaluations are adopted, especially in the application of *intelligent computer automated methodology*, or must be relegated to the archives of obsolete practices.

However, the phenomenal progress in technology over the past few decades has also confused the language of the engineering profession and, between engineering disciplines, engineers still have trouble *speaking the same language*, especially with regard to understanding the intricacies of concepts such as *integrity*, *reliability*,

availability, *maintainability* and *safety* not only of components, assemblies, subsystems or systems but also of their integration into larger complex installations.

Some of the more significant contributors to cost 'blow-outs' experienced by most engineering projects can be attributed to the complexity of their engineering design, both in technology and in the complex integration of their systems, as well as a lack of meticulous engineering design project management. The individual process systems on their own are adequately designed and constructed, often on the basis of previous similar, although smaller designs.

> *It is the critical combination and complex integration of many such process systems that gives rise to design complexity and consequent frequent failure, where high risks of the integrity of engineering design are encountered.*

Research by the author into this problem has indicated that large, expensive engineering projects may often have superficial *design reviews*. As an essential control activity of engineering design, design review practices can take many forms. At the lowest level, they consist of an examination of engineering drawings and specifications before construction begins. At the highest level, they consist of comprehensive *due diligence* evaluations. Comprehensive design reviews are included at different phases of the engineering design process, such as conceptual design, preliminary or schematic design, and final detail design.

> *In most cases, a predefined and structured basis of measure is rarely used against which the design, or design alternatives, should be reviewed.*

This situation inevitably prompts the question *how can the integrity of design be determined prior to any data being accumulated on the results of the operation and performance of the design*? In fact, how can the reliability of engineering plant and equipment be determined prior to the accumulation of any statistically meaningful failure data of the plant and its equipment? To further complicate matters, *how will plant and equipment perform in large integrated systems, even if nominal reliability values of individual items of equipment are known*? This is the dilemma that most design engineers are confronted with. The tools that most design engineers resort to in determining integrity of design are techniques such as hazardous operations (HazOp) studies, and simulation. Less frequently used techniques include hazards analysis (HazAn), fault-tree analysis, failure modes and effects analysis (FMEA), and failure modes effects and criticality analysis (FMECA).

This is evident by scrutiny of a typical Design Engineer's Definitive Scope of Work given in Appendix A. Despite the vast amount of research already conducted in the field of reliability analysis, many of these techniques seem to be either misunderstood or conducted incorrectly, or not even conducted at all, with the result that many high-cost super-projects eventually reach the construction phase without having been subjected to a rigorous and correct evaluation of the integrity of their designs. Verification of this statement is given in the extract below in which comment is delivered in part on an evaluation of the intended application of *HazOp* studies in conducting a preliminary design review for a recent laterite–nickel process design.

The engineer's definitive scope of work for a project includes the need for conducting preliminary design HazOp reviews as part of design verification. Reference to determining equipment criticality for mechanical engineering as well as for electrical engineering input can be achieved only through the establishment of failure modes and effects analysis (FMEA). There are, however, some concerns with the approach, as indicated in the following points.

Comment on intended HazOp studies for use in preliminary design reviews of a new engineering project:

- In HazOp studies, the differentiation between analyses at higher and at lower systems levels in assessing either hazardous operational failure consequences or system failure effects is extremely important from the point of view of determining *process criticality*, or of determining *equipment criticality*.
- The determination of *process criticality* can be seen as a preliminary HazOp, or a higher systems-level determination of process *failure consequences*, based upon *process function definition* in relation to the classical HazOp 'guide words', and obtained off the *schematic design* process flow diagrams (PFDs).
- The determination of *equipment criticality* can be seen as a detailed HazOp (or HazAn), or determination of system *failure effects*, which is based upon *equipment function definition*.
- The extent of analysis is very different between a preliminary HazOp and a detailed HazOp (or HazAn). Both are, however, essential for the determination of integrity of design, the one at a higher process level, and the other at a lower equipment level.
- A preliminary HazOp study is essential for the determination of integrity of design at process level, and should include *process reliability* that can be quantified from *process design criteria*.
- *The engineer's definitive scope of work for the project does not include a determination of process reliability, although process reliability can be quantified from process design criteria.*
- A detailed HazOp (or HazAn) is essential for the determination of integrity of design at a lower equipment level, and should include estimations of critical *equipment reliability* that can be quantified from *equipment design criteria*.
- *The engineer's definitive scope of work does not include a determination of equipment reliability, although equipment reliability is quantified from detail equipment design criteria.*
- Failure modes and effects analysis (FMEA) is dependent upon equipment function definition at assembly and component level in the systems breakdown structure (SBS), which is considered in equipment specification development during *schematic* and *detail design*. Furthermore, FMEA is strictly dependent upon a correctly structured SBS at the lower systems levels, usually obtained off the *detail design* pipe and instrument drawings (P&IDs).

It is obvious from the above comments that a severe lack of insight exists in the essential activities required to establish a proper evaluation of the *integrity* of engineering design, with the consequence that many 'good intentions' inevitably result

in superficial design reviews, especially with large, complex and expensive process designs.

Based on hands-on experience, as well as in-depth analysis of the potential causes of the cost 'blow-outs' of several super-projects, an inevitable conclusion can be derived that insufficient research has been conducted in determining the integrity of process engineering design, as well as in design review techniques. Much consideration is being given to engineering design based on the theoretical expertise and practical experience of process, chemical, civil, mechanical, electrical, electronic and industrial engineers, particularly from the point of view of '*what should be achieved*' to meet the design criteria. Unfortunately, it is apparent that not enough consideration is being given to '*what should be assured*' in the event the design criteria are not met. Thus, many high-cost super-projects eventually reach the construction phase without having been subjected to a rigorous evaluation of the integrity of their designs.

The contention that not enough consideration is being given in engineering design, as well as in design review techniques, to '*what should be assured*' in the event of design criteria not being met has therefore initiated the research presented in this handbook into a methodology for determining the integrity of engineering design. This is especially of concern with respect to the critical combinations and complex integrations of large engineering systems and their related equipment. Furthermore, an essential need has been identified in most engineering-based industries for a practical intelligent computer automated methodology to be applied in engineering design reviews as a structured basis of measure in determining the integrity of engineering design to achieve the required reliability, availability, maintainability and safety.

The objectives of this handbook are thus to:

1. Present concise theoretical formulation of conceptual and mathematical models of engineering design integrity in design synthesis, which includes design for reliability, availability, maintainability and safety during the conceptual, schematic or preliminary, and detail design phases.
2. Consider critical development criteria for intelligent computer automated methodology whereby the conceptual and mathematical models can be used practically in the mining, process and construction industries, as well as in most other engineering-based industries, to establish a structured basis of measure in determining the integrity of engineering design.

Several target platforms for evaluating and optimising the practical contribution of research in the field of engineering design integrity that is addressed in this handbook are focused on the design of large industrial processes that consist of many systems that give rise to design complexity and consequent high risk of design integrity. These industrial process engineering design 'super-projects' are insightful in that they incorporate almost all the different basic engineering disciplines, from chemical, civil, electrical, industrial, instrumentation and mechanical to process engineering. Furthermore, the increasing worldwide activity in the mining, process and construction industries makes such research and development very timely. The

following models have been developed, each for a specific purpose and with specific expected results, either to validate the developed theory on engineering design integrity or to evaluate and verify the design integrity of critical combinations and complex integrations of systems and equipment.

RAMS analysis modelling This was applied to validate the developed theory on the determination of the integrity of engineering design. This computer model was applied to a recently constructed engineering design of an environmental plant for the recovery of sulphur dioxide emissions from a nickel smelter to produce sulphuric acid.

Eighteen months after the plant was commissioned and placed into operation, failure data were obtained from the plant's distributed control system (DCS), and analysed with a view to matching the developed theory with real operational data after plant start-up. The comparative analysis included determination of systems and equipment criticality and reliability.

Dynamic systems simulation modelling This was applied with individually developed process equipment models (PEMs) based on *Petri net* constructs, to initially determine mass-flow balances for preliminary engineering designs of large integrated process systems. The models were used to evaluate and verify the process design integrity of critical combinations and complex integrations of systems and related equipment, for schematic and detail engineering designs. The process equipment models have been verified for correctness, and the relevant results validated, by applying the PEMs in a large dynamic simulation of a complex integration of systems.

Simulation modelling for design verification is common to most engineering designs, particularly in the application of simulating outcomes during the preliminary design phase. Dynamic simulation models are also used for design verification during the detail design phase but not to the extent of determining outcomes, as the level of complexity of the simulation models (and, therefore, the extent of data analysis of the simulation results) varies in accordance with the level of detail of the design.

At the higher systems level, typical of preliminary designs, dynamic simulation of the behaviour of exogenous, endogenous and status variables is both feasible and applicable. However, at the lower, more detailed equipment level, typical of detail designs, dynamic continuous and/or discrete event simulation is applicable, together with the appropriate verification and validation analysis of results, their sensitivity to changes in primary or base variables, and the essential need for adequate simulation run periods determined from statistical experimental design. Simulation analysis should not be based on model development time.

Mathematical modelling Modelling in the form of developed optimisation algorithms (OAs) of process design integrity was applied in predicting, assessing and evaluating reliability, availability, maintainability and safety requirements for the complex integration of process systems. These models were programmed into the PEM's script so that each individual process equipment model inherently has the facility for simplified data input, and the ability to determine its design integrity with

relevant output validation that includes the ability to determine the accumulative effect of all the PEMs' reliabilities in a PFD configuration.

Artificial intelligence-based (AIB) modelling This includes new artificial intelligence (AI) modelling techniques, such as *knowledge-based expert systems* within a *blackboard model*, which have been applied in the development of intelligent computer automated methodology for determining the integrity of engineering design. The *AIB* model provides a novel concept of *automated continual design reviews* throughout the engineering design process on the basis of *concurrent design* in an integrated *collaborative engineering design* environment. This is implemented through remotely located multidisciplinary groups of design engineers communicating via the Internet, who input specific design data and schematics into relevant knowledge-based expert systems, whereby each designed system or related equipment is automatically evaluated for integrity by the design group's expert system. The measures of integrity are based on the developed theory for predicting, assessing and evaluating reliability, availability, maintainability and safety requirements for complex integrations of engineering process systems. The relevant design criteria pertaining to each level of a systems hierarchy of the engineering designs are incorporated in an all-encompassing blackboard model. The blackboard model incorporates multiple, diverse program modules, called knowledge sources (in knowledge-based expert systems), which cooperate in solving design problems such as determining the integrity of the designs. The blackboard is an OOP application containing several databases that hold shared information among knowledge sources. Such information includes the RAMS analysis data, results from the optimisation algorithms, and compliance to specific design criteria, relevant to each level of systems hierarchy of the designs. In this manner, integrated systems and related equipment are continually evaluated for design compatibility and integrity throughout the engineering design process, particularly where designs of large systems give rise to design complexity and consequent high risk of design integrity.

Contribution of research in integrity of engineering design Many of the methods covered in this handbook have already been thoroughly explored by other researchers in the various fields of reliability, availability, maintainability and safety, though more in the field of engineering processes than of engineering design. What makes this handbook unique is the combination of practical methods with techniques in probability and possibility modelling, mathematical algorithmic modelling, evolutionary algorithmic modelling, symbolic logic modelling, artificial intelligence modelling, and object oriented computer modelling, in a structured approach to determining the integrity of engineering design. This endeavour has encompassed not only a depth of research into these various methods and techniques but also a breadth of research into the concept of integrity in engineering design. Such breadth is represented by the combined topics of reliability and performance, availability and maintainability, and safety and risk, in an overall concept of the integrity of engineering design—which has been practically segmented into three progressive phases, i.e. a conceptual design phase, a preliminary or schematic design phase, and a detail design phase.

Thus, a matrix combination of the topics has been considered in each of the three phases—a total of 18 design methodology aspects for consideration—hence, the voluminous content of this handbook. Such a comprehensive combination of depth and breadth of research resulted in the conclusion that certain methods and techniques are more applicable to specific phases of the engineering design process, as indicated in the theoretical overview and analytic development of each of the topics. The research has not remained on a theoretical basis, however, but includes the application of various computer models in specific target industry projects, resulting in a wide range of design deliverables related to the theoretical topics. Taking all these design methodology aspects into consideration, the research presented in this handbook can rightfully claim uniqueness in both integrative modelling and practical application in determining the integrity of process engineering design. A practical industry-based outcome is given in the establishment of an intelligent computer automated methodology for determining integrity of engineering design, particularly for design reviews at the various progressive phases of the design process, namely conceptual, preliminary and detail engineering design. The overall value of such methodology is in the enhancement of design review methods for future engineering projects.

1.1.1 Development and Scope of Design Integrity Theory

The scope of research for this handbook necessitated an in-depth coverage of the relevant theory underlying the approach to determining the *integrity* of engineering design, as well as an overall combination of the topics that would constitute such a methodology. The scope of theory covered in a comprehensive selection of available literature included the following subjects:

- *Failure analysis*: the basics of failure, failure criticality, failure models, risk and safety.
- *Reliability analysis*: reliability theory, methods and models, reliability and systems engineering, control and prediction.
- *Availability analysis*: availability theory, methods and models, availability engineering, control and prediction.
- *Maintainability analysis*: maintainability theory, methods and models, maintainability engineering, control and testing.
- *Quantitative analysis*: programming, statistical distributions, quantitative uncertainty, Markov analysis and probability theory.
- *Qualitative analysis*: descriptive statistics, complexity, qualitative uncertainty, fuzzy logic and possibility theory.
- *Systems analysis*: large systems integration, optimisation, dynamic optimisation, systems modelling, decomposition and control.
- *Simulation analysis*: planning, formulation, specification, evaluation, verification, validation, computation, modelling and programming.

- *Process analysis*: general process reactions, mass transfer, and material and energy balance, and process engineering.
- *Artificial intelligence modelling*: knowledge-based expert systems and blackboard models ranging from domain expert systems (DES), artificial neural systems (ANS) and procedural diagnostic systems (PDS) to blackboard management systems (BBMS), and the application of expert system shells such as CLIPS, fuzzy CLIPS, EXSYS and CORVID.

Essential preliminaries The very many methods and techniques presented in this handbook, and developed by as many authors, are referenced at the end of each following chapter. Additionally, a listing of books on the scope of the theory covered is given in Appendix B. However, besides these methods and techniques and theory, certain essential preliminaries used by design engineers in determining the integrity of engineering design include activities such as:

- Systems breakdown structures (SBSs) development
- Process function definition
- Quantification of engineering design criteria
- Determination of failure consequences
- Determination of preliminary design reliability
- Determination of systems interdependencies
- Determination of process criticality
- Equipment function definition
- Quantification of detail design criteria
- Determination of failure effects
- Failure modes and effects analysis (FMEA)
- Determination of detail design reliability
- Failure modes effects and criticality analysis (FMECA)
- Determination of equipment criticality.

However, very few engineering designs actually incorporate all of these activities (except for the typical quantification of process design criteria and detail equipment design criteria) and, unfortunately, very few design engineers apply or even understand the theoretical implications and practical application of such activities. The methodology researched in this handbook, in which engineering design problems are formulated to achieve optimal integrity, has been extended to accommodate its use in conceptual and preliminary or schematic design in which most of the design's components have not yet been precisely defined in terms of their final configuration and functional performance.

> *The approach, then, is to determine methodology, particularly intelligent computer automated methodology, in which design for reliability, availability, maintainability and safety is applied to systems the components of which have not been precisely defined.*

1.1.2 Designing for Reliability, Availability, Maintainability and Safety

The fundamental understanding of the concepts of reliability, availability and maintainability (and, to a large extent, an empirical understanding of safety) has in the main dealt with statistical techniques for the measure and/or estimation of various parameters related to each of these concepts, *based on obtained data*. Such data may be obtained from current observations or past experience, and may be complete, incomplete or censored. Censored data arise from the cessation of experimental observations prior to a final conclusion of the results. These statistical techniques are predominantly couched in probability theory.

The usual meaning of the term *reliability* is understood to be '*the probability of performing successfully*'. In order to assess reliability, the approach is based upon available test data of successes or failures, or on field observations relative to performance under either actual or simulated conditions. Since such results can vary, the estimated reliability can be different from one set of data to another, even if there are no substantial changes in the physical characteristics of the item being assessed. Thus, associated with the reliability estimate, there is also a measure of the significance or accuracy of the estimate, termed the 'confidence level'. This measure depends upon the amount of data available and/or the results observed. The data are normally governed by some parametric probability distribution. This means that the data can be interpreted by one or other mathematical formula representing a specific statistical probability distribution that belongs to a family of distributions differing from one another only in the values of their parameters.

Such a family of distributions may be grouped accordingly:

- Beta distribution
- Binomial distribution
- Lognormal distribution
- Exponential (Poisson) distribution
- Weibull distribution.

Estimation techniques for determining the level of confidence related to an assessment of reliability based on these probability distributions are the methods of *maximum likelihood*, and *Bayesian estimation*.

In contrast to reliability, which is typically assessed for *non-repairable systems*, i.e. without regard to whether or not a system is repaired and restored to service after a failure, *availability* and *maintainability* are principally assessed for *repairable systems*. Both availability and maintainability have the dimensions of a probability distribution in the range zero to one, and are based upon time-dependent phenomena. The difference between the two is that availability is a measure of total performance effectiveness, usually of systems, whereas maintainability is a measure of effectiveness of performance during the period of restoration to service, usually of equipment.

Reliability assessment based upon the family of statistical probability distributions considered previously is, however, subject to a somewhat narrow point of view—success or failure in the function of an item. They do not consider situations in which there are some means of backup for a failed item, either in the form of *replacement*, or in the form of *restoration*, or which include multiple failures with standby reliability, i.e. the concept of *redundancy*, where a redundant item is placed into service after a failure. Such situations are represented by additional probability distributions, namely:

- Gamma distribution
- Chi-square distribution.

Availability, on the other hand, has to do with two separate events—failure and repair. Therefore, assigning confidence levels to values of availability cannot be done parametrically, and a technique such as Monte Carlo simulation is employed, based upon the estimated values of the parameters of time-to-failure and time-to-repair distributions. When such distributions are exponential, they can be reviewed in a Bayesian framework so that not only the time period to specific events is simulated but also the *values* of the parameters. Availability is usually assessed with Poisson or Weibull time-to-failure and exponential or lognormal time-to-repair.

Maintainability is concerned with only one random variable—the repair time for a failed system. Thus, assessing maintainability implies the same level of difficulty as does assessing reliability that is concerned with only one event, namely the failure of a system in its operating condition. In both cases, if the time to an event of failure is governed by either a parametric, Poisson or Weibull distribution, then the confidence levels of the estimates can also be assigned parametrically.

However, in *designing for reliability, availability and maintainability*, it is more often the case that the measure and/or estimation of various parameters related to each of these concepts *is not based on obtained data*. This is simply due to the fact that available data do not exist. This poses a severe problem for engineering design analysis in determining the integrity of the design, in that the analysis cannot be *quantitative*. Furthermore, the complexity arising from an integration of engineering systems and their interactions makes it somewhat impossible to gather meaningful statistical data that could allow for the use of objective probabilities in the analysis. Other acceptable methods must be sought to determine the integrity of engineering design in the situation where data are not available or not meaningful. These methods are to be found in a *qualitative* approach to engineering design analysis. A qualitative analysis of the integrity of engineering design would need to incorporate qualitative concepts such as *uncertainty* and *incompleteness*. Uncertainty and incompleteness are inherent to engineering design analysis, whereby uncertainty, arising from a complex integration of systems, can best be expressed in qualitative terms, necessitating the results to be presented in the same qualitative measures. Incompleteness considers results that are more or less sure, in contrast to those that are only possible. The methodology for determining the integrity of engineering design is thus not solely a consideration of the fundamental *quantitative measures* of engineering design analysis based on *probability theory* but also consideration of

a *qualitative analysis* approach to selected conventional techniques. Such a qualitative analysis approach is based upon conceptual methodologies ranging from intervals and labelled intervals; uncertainty and incompleteness; fuzzy logic and fuzzy reasoning; through to approximate reasoning and *possibility theory*.

a) Designing for Reliability

In an elementary process, *performance* may be measured in terms of input, throughput and output quantities, whereas *reliability* is generally described in terms of the probability of failure or a mean time to failure of equipment (i.e. assemblies and components). This distinction is, however, not very useful in engineering design because it omits the assessment of *system reliability* from preliminary design considerations, leaving the task of evaluating *equipment reliability* during detail design, when most equipment items have already been specified. A closer scrutiny of reliability is thus required, particularly the broader concept of *system reliability*.

> *System reliability* can be defined as "*the probability that a system will perform a specified function within prescribed limits, under given environmental conditions, for a specified time*".

An important part of the definition of system reliability is the ability to perform within prescribed limits. The boundaries of these limits can be quantified by defining constraints on acceptable performance. The constraints are identified by considering the *effects of failure* of each identified performance variable. If a particular performance variable (designating a specific required duty) lies within the space bounded by these constraints, then it is a feasible design solution, i.e. the design solution for a chosen performance variable does not violate its constraints and result in unacceptable performance. The best performance variable would have the greatest variance or *safety margin* from its relative constraints. Thus, a design that has the highest safety margin with respect to all constraints will inevitably be the most reliable design.

Designing for reliability at the systems level includes all aspects of the ability of a system to perform. When assemblies are configured together in a system, the system gains a collective identity with multiple functions, each function identified by the collective result of the duties of each assembly. Preliminary design considerations describe these functions at the system level and, as the design process progresses, the required duties at the assembly level are identified, in effect constituting the collective performance of components that are defined at the detail design stage. In process systems, no difference is made between *performance* and *reliability* at the component level. When components are configured together in an assembly, the assembly gains a collective identity with designated duties.

Performance is the ability of such an assembly of components to carry out its duties, while reliability at the component level is determined by the ability of each of the components to resist failure. Unacceptable performance is considered from the point of view of the assembly not being able to meet a specific performance variable or designated duty, by an evaluation of the *effects of failure* of the inherent

components on the duties of the assembly. Designing for reliability at the *preliminary* design stage would be to maximise the reliability of a *system* by ensuring that there are no 'weak links' (i.e. assemblies) resulting in failure of the system to perform its required functions.

Similarly, designing for reliability at the *detail* design stage would be to maximise the reliability of an *assembly* by ensuring that there are no 'weak links' (i.e. components) resulting in failure of the assembly to perform its required duties.

For example, in a mechanical system, a pump is an assembly of components that performs specific duties that can be measured in terms of performance variables such as pressure, flow rate, efficiency and power consumption. However, if a pump continues to operate but does not deliver the correct flow rate at the right pressure, then it should be regarded as having failed because it does not fulfil its prescribed duty. It is incorrect to describe a pump as 'reliable' if the rates of failure of its components are low, yet it does not perform a specific duty required of it.

Similarly, in a hydraulic system, a particular assembly may appear to be 'reliable' if the rates of failure of its components are low, yet it may fail to perform a specific duty required of it. Numerous examples can be listed in systems pertaining to the various engineering disciplines (i.e. chemical, civil, electrical, electronic, industrial, mechanical, process, etc.), many of which become critical when multiple assemblies are configured together in single systems and, in turn, multiple systems are integrated into large, complex engineering installations.

The intention of designing for reliability is thus to design integrated systems with assemblies that effectively fulfil all their required duties.

The design for reliability method thus integrates functional failure as well as functional performance criteria so that a *maximum safety margin* is achieved with respect to acceptable limits of performance. The objective is to produce a design that has the highest possible safety margin with respect to all constraints. However, because many different constraints defined in different units may apply to the overall performance of the system, a method of data point generation based on the limits of non-dimensional performance measures allows design for reliability to be quantified.

The choice of limits of performance for such an approach is generally made with respect to the consequences of failure and reliability expectations. If the consequences of failure are high, then limits of acceptable performance with high safety margins that are well clear of failure criteria are chosen. Similarly, if failure criteria are imprecise, then high safety margins are adopted.

This approach has been further expanded, applying the method of *labelled interval calculus* to represent sets of systems functioning under sets of failures and performance intervals. The most significant advantage of this method is that, besides not having to rely on the propagation of single estimated values of failure data, it does not have to rely on the determination of single values of maximum and minimum acceptable limits of performance for each criterion. Instead, *constraint propagation* of intervals about sets of performance values is applied. As these intervals are defined, a multi-objective optimisation of *availability* and *maintainability*

performance values is computed, and optimal solution sets to different sets of performance intervals are determined.

In addition, the concept of *uncertainty* in design integrity, both in technology as well as in the complex integration of multiple systems of large engineering processes, is considered through the application of *uncertainty calculus* utilising *fuzzy sets* and *possibility theory*. Furthermore, the application of *uncertainty* in failure mode effects and criticality analyses (FMECAs) describes the impact of possible faults that could arise from the complexity of process engineering systems, and forms an essential portion of knowledge gathered during the schematic design phase of the engineering design process.

The knowledge gathered during the schematic design phase is incorporated in a knowledge base that is utilised in an artificial intelligence-based blackboard system for detail design. In the case where *data are sparse or non-existent* for evaluating the performance and reliability of engineering designs, *information integration technology (IIT)* is applied. This multidisciplinary methodology is particularly considered where complex integrations of engineering systems and their interactions make it difficult and even impossible to gather meaningful statistical data.

b) Designing for Availability

Designing for availability, as it is applied to an item of equipment, includes the aspects of *utility* and *time*. Designing for availability is concerned with equipment *usage* or *application* over a period of *time*. This relates directly to the equipment (i.e. assembly or component) being able to perform a specific function or duty within a given time frame, as indicated by the following definition:

Availability can be simply defined as "*the item's capability of being used over a period of time*", and the measure of an item's availability can be defined as "*that period in which the item is in a usable state*". Performance variables relating availability to reliability and maintainability are concerned with the measures of time that are subject to equipment failure. These measures are mean time between failures (MTBF), and mean downtime (MDT) or mean time to repair (MTTR). As with designing for reliability, which includes all aspects of the ability of a system to perform, designing for availability includes reliability and maintainability considerations that are integrated with the performance variables related to the measures of time that are subject to equipment failure. Designing for availability thus incorporates an assessment of *expected performance* with respect to the performance measures of MTBF, MDT or MTTR, in relation to the performance capabilities of the equipment. In the case of MTBF and MTTR, there are no limits of capability. Instead, prediction of the performance of equipment considers the *effects of failure* for each of the measures of MTBF and MTTR.

System availability implies the ability to perform within prescribed limits quantified by defining constraints on acceptable performance that is identified by considering the *consequences of failure* of each identified performance variable. Designing for availability during the *preliminary* or *schematic design* phase of the engineering

design process includes intelligent computer automated methodology based on *Petri nets (PN)*. Petri nets are useful for modelling complex systems in the context of systems performance, in designing for availability subject to preventive maintenance strategies that include complex interactions such as component renewal. Such interactions are time related and dependent upon component age and estimated residual life of the components.

c) Designing for Maintainability

Maintainability is that aspect of maintenance that takes *downtime* into account, and can be defined as "*the probability that a failed item can be restored to an operational effective condition within a given period of time*". This restoration of a failed item to an operational effective condition is usually when *repair action*, or *corrective maintenance action*, is performed in accordance with prescribed standard procedures. The item's operational effective condition in this context is also considered to be the item's *repairable condition*.

Corrective maintenance action is the *action* to rectify or set right defects in the item's *operational* and *physical conditions*, on which its functions depend, in accordance with a standard. Maintainability is thus the *probability* that an item can be restored to a *repairable condition* through *corrective action*, in accordance with prescribed standard procedures within a given period of time. It is significant to note that maintainability is achieved not only through restorative corrective maintenance action, or repair action, in accordance with prescribed standard procedures, but also within a given period of *time*. This *repair action* is in fact determined by the mean time to repair (MTTR), which is a measure of the performance of maintainability. A fundamental principle is thus identified:

> *Maintainability is a measure of the repairable condition of an item that is determined by the mean time to repair (MTTR), established through corrective maintenance action.*

Designing for maintainability fundamentally makes use of maintainability prediction techniques as well as specific quantitative maintainability analysis models relating to the operational requirements of the design. Maintainability predictions of the operational requirements of a design during the conceptual design phase can aid in design decisions where several design options need to be considered. Quantitative maintainability analysis during the schematic and detail design phases considers the assessment and evaluation of maintainability from the point of view of *maintenance* and *logistics support* concepts. Designing for maintainability basically entails a consideration of design criteria such as *visibility*, *accessibility*, *testability*, *repairability* and *inter-changeability*. These criteria need to be verified through *maintainability design reviews*, conducted during the various design phases.

Designing for maintainability at the *systems* level requires an evaluation of the *visibility*, *accessibility* and *repairability* of the system's equipment in the event of failure. This includes integrated systems shutdown during planned maintenance.

Designing for maintainability, as it is applied to an item of *equipment*, includes the aspects of *testability*, *repairability* and *inter-changeability* of an assembly's inherent components. In general, the concept of designing for maintainability is concerned with the restoration of equipment that has failed to perform over a period of time. The performance variable used in the determination of maintainability that is concerned with the measure of time subject to equipment failure is the mean time to repair (MTTR).

Thus, besides providing for *visibility*, *accessibility*, *testability*, *repairability* and *inter-changeability*, designing for maintainability also incorporates an assessment of *expected performance* in terms of the measure of MTTR in relation to the performance capabilities of the equipment. Designing for maintainability during the preliminary design phase would be to minimise the MTTR of a *system* by ensuring that failure of an inherent assembly to perform a specific duty can be restored to its *expected performance* over a period of *time*. Similarly, designing for maintainability during the detail design phase would be to minimise the MTTR of an *assembly* by ensuring that failure of an inherent component to perform a specific function can be restored to its *expected initial state* over a period of *time*.

d) Designing for Safety

Traditionally, assessments of the *risk of failure* are made on the basis of allowable *factors of safety* obtained from previous failure experiences, or from empirical knowledge of similar systems operating in similar anticipated environments. Conventionally, the factor of safety has been calculated as the ratio of what are assumed to be nominal values of *demand* and *capacity*. In this context, *demand* is the resultant of many uncertain variables of the system under consideration, such as loading stress, pressures and temperatures. Similarly, *capacity* depends on the properties of materials strength, physical dimensions, constructability, etc. The nominal values of both *demand* and *capacity* cannot be determined with certainty and, hence, their ratio, giving the conventional factor of safety, is a random variable. Representation of the values of *demand* and *capacity* would thus be in the form of probability distributions whereby, if maximum *demand* exceeded minimum *capacity*, the distributions would overlap with a non-zero *probability of failure*.

A convenient way of assessing this probability of failure is to consider the difference between the demand and capacity functions, termed the *safety margin*, a random variable with its own probability distribution. *Designing for safety*, or the measure of *adequacy of a design*, where *inadequacy* is indicated by the measure of the probability of failure, is associated with the determination of a *reliability index* for items at the equipment and component levels. The reliability index is defined as the number of standard deviations between the mean value of the probability distribution of the safety margin, where the safety margin is zero. It is the reciprocal of the coefficient of variation of the safety margin.

Designing for safety furthermore includes analytic techniques such as *genetic algorithms* and/or *artificial neural networks (ANN)* to perform multi-objective optimi-

sations of engineering design problems. The use of genetic algorithms in designing for safety is a new approach in determining solutions to the redundancy allocation problem for series-parallel systems design comprising multiple components. Artificial neural networks in designing for safety offer feasible solutions to many design problems because of their capability to simultaneously relate multiple quantitative *and* qualitative variables, as well as to form models based solely on minimal data.

1.2 Artificial Intelligence in Design

Analysis of Target Engineering Design Projects

A stringent approach of objectivity is essential in implementing the theory of design integrity in any target engineering design project, particularly with regard to the numerous applications of mathematical models in intelligent computer automated methodology. Selection of target engineering projects was therefore based upon illustrating the development of mathematical and simulation models of process and equipment functionality, and development of an artificial intelligence-based (AIB) blackboard model to determine the integrity of process engineering design.

As a result, three different target engineering design projects are selected that relate directly to the progressive stages in the development of the theory, and to the levels of modelling sophistication in the practical application of the theory:

- *RAMS analysis model (product assurance)* for an engineering design project of an environmental plant for the recovery of sulphur dioxide emissions from a metal smelter to produce sulphuric acid as a by-product. The purpose of implementing the RAMS analysis model in this target engineering design project is to *validate the developed theory* of design integrity in designing for reliability, availability, maintainability and safety, for eventual inclusion in intelligent computer automated methodology using artificial intelligence-based (AIB) modelling.
- *OOP simulation model (process analysis)* for an engineering design super-project of an alumina plant with establishment costs in excess of a billion dollars. The purpose of implementing the object oriented programming (OOP) simulation model in this target engineering design project was to evaluate the mathematical algorithms developed for assessing the reliability, availability, maintainability and safety requirements of complex process systems, as well as for the complex integration of process systems, for eventual inclusion in intelligent computer automated methodology using AIB modelling.
- *AIB blackboard model (design review)* for an engineering design super-project of a nickel-from-laterite processing plant with establishment costs in excess of two billion dollars. The AIB blackboard model includes intelligent computer automated methodology for application of the developed theory and the mathematical algorithms.

1.2.1 Development of Models and AIB Methodology

Applied computer modelling includes up-to-date object oriented software programming applications incorporating integrated systems simulation modelling, and *AIB* modelling including knowledge-based expert systems as well as blackboard modelling. The *AIB* modelling provides for *automated continual design reviews* throughout the engineering design process on the basis of *concurrent design* in an integrated *collaborative engineering design* environment. Engineering designs are composed of highly integrated, tightly coupled components where interactions are essential to the economic execution of the design.

Thus, *concurrent*, rather than *sequential* consideration of requirements such as structural, thermal, hydraulic, manufacture, construction, operational and maintenance constraints will inevitably result in superior designs. Creating *concurrent design* systems for engineering designers requires knowledge of downstream activities to be infused into the design process so that designs can be generated rapidly and correctly. The *design space* can be viewed as a multi-dimensional space, in which each dimension has a different life-cycle objective such as serviceability or integrity.

An intelligent design system should aid the designer in understanding the interactions and trade-offs among different and even conflicting requirements. The intention of the AIB blackboard is to surround the designer with expert systems that provide feedback on continual design reviews of the design as it evolves throughout the engineering design process. These experts systems, termed *perspectives*, must be able to generate information that becomes part of the design (e.g. mass-flow balances and flow stresses), and portions of the geometry (e.g. the shapes and dimensions). The perspectives are not just a sophisticated toolbox for the designer; rather, they are a group of *advisors* that interact with one another and with the designer, as well as identify conflicting inputs in a *collaborative design* environment. Implementation by multidisciplinary remotely located groups of designers inputs design data and schematics into the relevant perspectives or knowledge-based expert systems, whereby each design solution is collaboratively evaluated for integrity. Engineering design includes important characteristics that have to be considered when developing design models, such as:

- Design is an optimised search of a number of design alternatives.
- Previous designs are frequently used during the design process.
- Design is an increasingly distributed and collaborative activity.

Engineering design is a complex process that is often characterised as a top-down search of the space of possible solutions, considered to be the general norm of how the design process should proceed. This process ensures an optimal solution and is usually the construct of the initial design specification. It therefore involves maintaining numerous candidate solutions to specific design problems in parallel, whereby designers need to be adept at generating and evaluating a range of candidate solutions.

The term *satisficing* is used to describe how designers sometimes limit their search of the design solution space, possibly in response to technology limitations, or to reduce the time taken to reach a solution because of schedule or cost constraints. Designers may opportunistically deviate from an optimal strategy, especially in engineering design where, in many cases, the design may involve early commitment to and refining of a sub-optimal solution. In such cases, it is clear that satisficing is often advantageous due to potentially reduced costs or where a satisfactory, rather than an optimal design is required. However, solving complex design problems relies heavily on the designer's knowledge, gained through experience, or making use of previous design solutions.

The concept of *reuse* in design was traditionally limited to utilising personal experience, with reluctance to copy solutions of other designers. The modern trend in engineering design is, however, towards more extensive design reuse in a collaborative environment. New computing technology provides greater opportunities for design reuse and satisficing to be applied, at least in part, as a collaborative, distributed activity. A large amount of current research is concerned with developing tools and methodologies to support design teams separated by space and time to work effectively in a *collaborative design* environment.

a) The RAMS Analysis Model

The *RAMS analysis model* incorporates all the essential preliminaries of *systems analysis* to validate the developed theory for the determination of the integrity of engineering design. A layout of part of the RAMS analysis model of an environmental plant is given in Fig. 1.1.

The *RAMS analysis model* includes systems breakdown structures, process function definition, determination of failure consequences on system performance, determination of process criticality, equipment functions definition, determination of failure effects on equipment functionality, failure modes effects and criticality analysis (FMECA), and determination of equipment criticality.

b) The OOP Simulation Model

The *OOP simulation model* incorporates all the essential preliminaries of *process analysis* to initially determine process characteristics such as process throughput, output, input and capacity. The application of the model is primarily to determine its capability of accurately assessing the effect of complex integrations of systems, and process output mass-flow balancing in preliminary engineering design of large integrated processes. A layout of part of the OOP simulation model is given in Fig. 1.2.

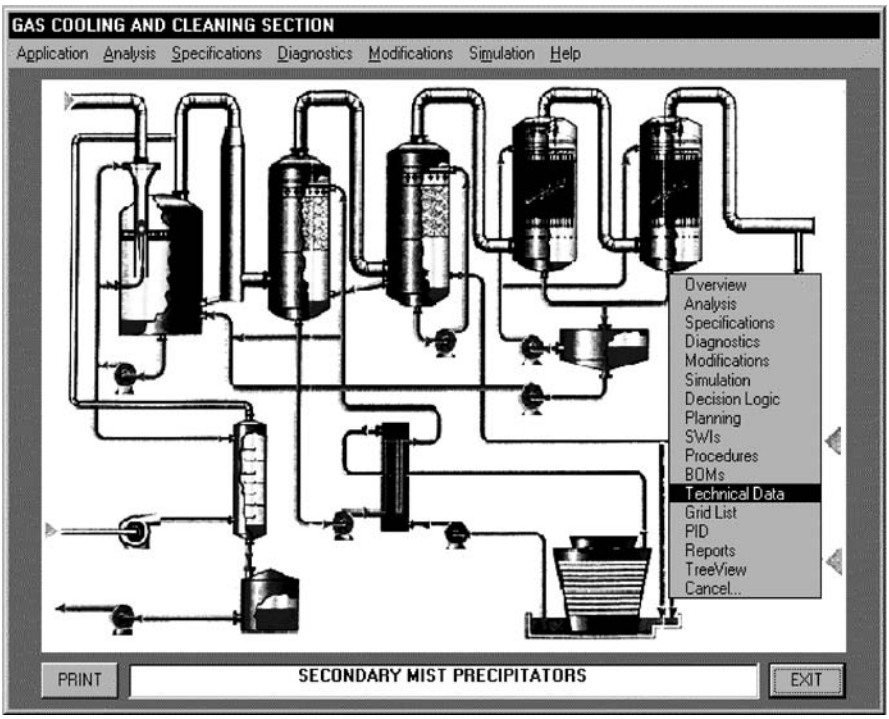


Fig. 1.1 Layout of the RAM analysis model

c) The AIB Blackboard Model

The *AIB blackboard model* consists of three fundamental stages of analysis for determining the integrity of engineering design, specifically preliminary design process analysis, detail design plant analysis and commissioning operations analysis. The preliminary design process analysis incorporates the essential preliminaries of *design review*, such as process definition, performance assessment, process design evaluation, systems definition, functions analysis, risk assessment and criticality analysis, linked to an inter-disciplinary collaborative *knowledge-based expert system*. Similarly, the detail design plant analysis incorporates the essential preliminaries of *design integrity* such as FMEA and plant criticality analysis. The application of the model is fundamentally to establish *automated continual design reviews* whereby the integrity of engineering design is determined concurrently throughout the engineering design process. Figure 1.3 shows the selection screen of a multi-user interface 'blackboard' in collaborative engineering design.

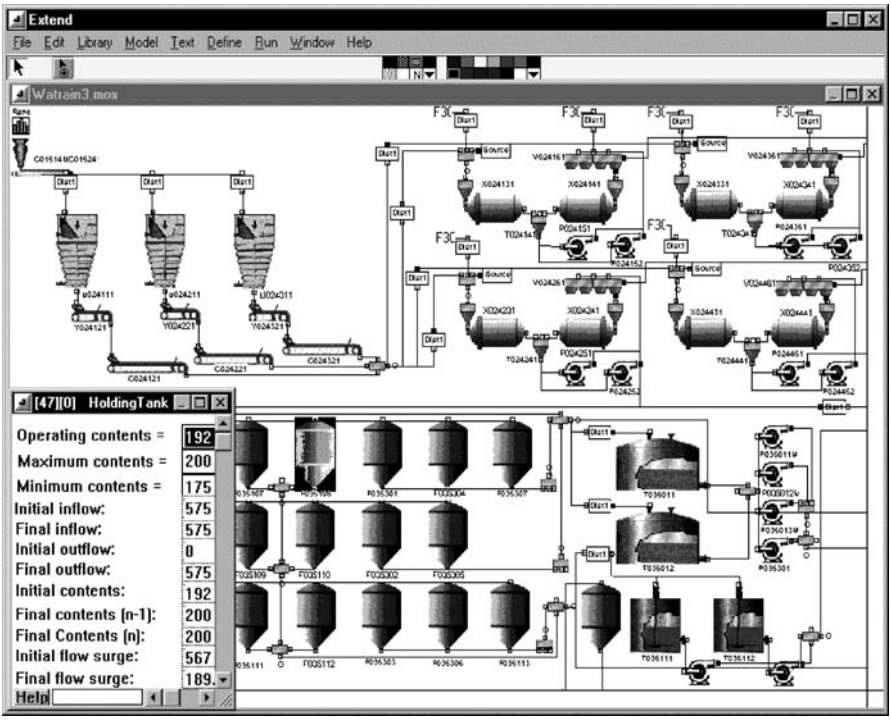


Fig. 1.2 Layout of part of the OOP simulation model

1.2.2 Artificial Intelligence in Engineering Design

Implementation of the various models covered in this handbook predominantly focuses on determining the applicability and benefit of *automated continual design reviews* throughout the engineering design process. This hinges, however, upon a broader understanding of the principles and philosophy of the use of *artificial intelligence (AI)* in engineering design, particularly in which new *AI* modelling techniques are applied, such as the inclusion of *knowledge-based expert systems* in *blackboard models*. Although these modelling techniques are described in detail later in the handbook, it is essential at this stage to give a brief account of *artificial intelligence* in engineering design.

The application of *artificial intelligence (AI)* in engineering design, through *artificial intelligence-based (AIB) computer modelling*, enables decisions to be made about acceptable design performance by considering the essential systems design criteria, the functionality of each particular system, the effects and consequences of potential and functional failure, as well as the complex integration of the systems as a whole. It is unfortunate that the growing number of unfulfilled promises and expectations about the capabilities of artificial intelligence seems to have damaged the credibility of *AI* and eroded its true contributions and benefits. The early advances

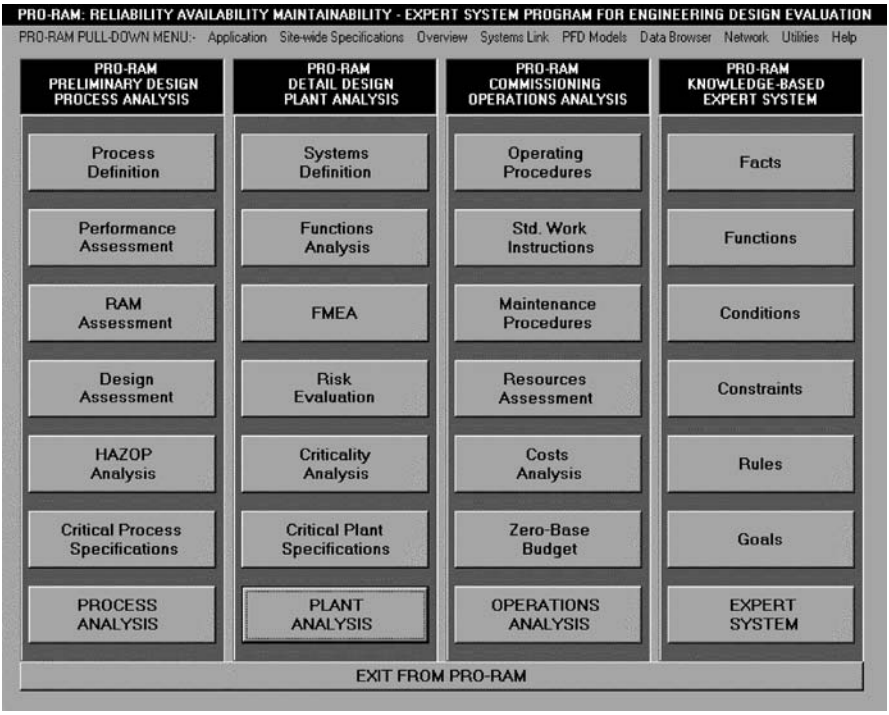


Fig. 1.3 Layout of the AIB blackboard model

of *expert systems*, which were based on more than 20 years of research, were over-extrapolated by many researchers looking for a feasible solution to the complexity of integrated systems design. Notwithstanding the problems of *AI*, recent artificial intelligence research has produced a set of new techniques that can usefully be employed in determining the integrity of engineering design. This does not mean that *AI* in itself is sufficient, or that *AI* is mutually exclusive of traditional engineering design. In order to develop a proper perspective on the relationship between *AI* technology and engineering design, it is necessary to establish a framework that provides the means by which *AI* techniques can be applied with conventional engineering design. *Knowledge-based systems* provide such a framework.

a) Knowledge-Based Systems

Knowledge engineering is a problem-solving strategy and an approach to programming that characterises a problem principally by the type of knowledge involved.

At one end of the spectrum lies conventional engineering design technology based on well-defined, algorithmic knowledge. At the other end of the spectrum lies *AI*-related engineering design technology based on ill-defined heuristic knowledge.

Among the problems that are well suited for knowledge-based systems are design problems, in particular engineering design. As engineering knowledge is heterogeneous in terms of the kinds of problems that it encompasses and the methods used to solve these, the use of heterogeneous representations is necessary. Attempts to characterise engineering knowledge have resulted in the following classification of the properties that are essential in constructing a knowledge-based expert system:

- Knowledge representation,
- Problem-solving strategy, and
- Knowledge abstractions.

b) Engineering Design Expert Systems

The term '*expert system*' refers to a computer program that is largely a collection of heuristic rules (rules of thumb) and detailed domain facts that have proven useful in solving the special problems of some or other technical field. *Expert systems* to date are basically an outgrowth of *artificial intelligence*, a field that has for many years been devoted to the study of problem-solving using heuristics, to the construction of symbolic representations of knowledge, to the process of communicating in natural language and to learning from experience.

Expertise is often defined to be that body of knowledge that is acquired over many years of experience with a certain class of problem. One of the hallmarks of an expert system is that it is constructed from the interaction of two types of disciplines: *domain experts*, or practicing experts in some technical domain, and *knowledge engineers*, or *AI* specialists skilled in analysing processes and problem-solving approaches, and encoding these in a computer system.

The best domain expert is one with years, even decades, of practical experience, and the best expert system is one that has been created through a close scrutiny of the expert's domain by a '*knowledgeable*' knowledge engineer. However, the question often asked is *which kinds of problems are most amenable to this type of approach?*

Inevitably, problems requiring knowledge-intensive problem solving, where years of accumulated experience produce good performance results, must be the most suited to such an approach. Such domains have complex fact structures, with large volumes of specific items of information, organised in particular ways. The domain of engineering design is an excellent example of knowledge-intensive problem solving for which the application of expert systems in the design process is ideally suited, even more so for determining the integrity of engineering design. Often, though, there are no known algorithms for approaching these problems, and the domain may be poorly formalised. Strategies for approaching design problems may be diverse and depend on particular details of a problem situation. Many aspects of the situation need to be determined during problem solving, usually selected from a much larger set of possible needs of which some may be expensive to determine—thus, the *significance* of a particular need must also be considered.

c) Expert Systems in Engineering Design Project Management

The advantages of an expert system are significant enough to justify a major effort to develop these. Decisions can be obtained more reliably and consistently, where an explanation of the final answers becomes an important benefit. An expert system is thus especially useful in a consultation mode of complex engineering designs where obscure factors may be overlooked, and is therefore an ideal tool in *engineering design project management* in which the following important areas of engineering design may be impacted:

- Rapid checking of preliminary design concepts, allowing more alternatives to be considered;
- Iteration over the design process to improve on previous attempts;
- Assistance with and automation of complex tasks and activities of the design process where expertise is specialised and technical;
- Strategies for searching in the space of alternative designs, and monitoring of progress towards the targets of the design process;
- Integration of a diverse set of tools, with expertise applied to the problem of engineering design project planning and control;
- Integration of the various stages of an engineering design project, inclusive of procurement/installation, construction/fabrication, and commissioning/warranty by having knowledge bases that can be distributed for wide access in a collaborative design environment.

d) Research in Expert Systems for Engineering Design

Within the past several years, a number of tools have been developed that allow a higher-level approach to building expert systems in general, although most still require some programming skill. A few provide an integrated knowledge engineering environment combining features of all of the available *AI* languages.

These languages (CLIPS, JESS, etc.) are suitable and efficient for use by *AI* professionals. A number of others are very specialised to specific problem types, and can be used without programming to build up a knowledge base, including a number of small tools that run on personal computers (EXSYS, CORVID, etc.). A common term for the more powerful tools is *shell*, referring to their origins as specialised expert systems of which the *knowledge base* has been removed, leaving only a *shell* that can perform the essential functions of an expert system, such as

- an inference engine,
- a user interface, and
- a knowledge storage medium.

For engineering design applications, however, good expert system development tools are still being conceptualised and experimented with. Some of the most recent techniques in *AI* may become the basis for powerful design tools. Also, a number of the elements of the design process fall into the diagnostic–selection category, and

these can be tackled with existing expert system shells. Many expert systems are now being developed along these limited lines. The development of a shell that has the basic ingredients for assisting or actually doing design is still an open research topic.

e) Blackboard Models

Early expert systems used *rules* as the basic data structure to address heuristic knowledge. From the rule-based expert system, there has been a shift to a more powerful architecture based on the notion of cooperating experts (termed *blackboard models*) that allows for the integration of algorithmic design approaches with *AI* techniques. Blackboard models provide the means by which *AI* techniques can be applied in determining the integrity of engineering designs.

Currently, one of the main areas of development is to provide integrative means to allow various design systems to communicate with each other both dynamically and cooperatively while working on the same design problem from different viewpoints (i.e. concurrent design). What this amounts to is having a diverse team of experts or multidisciplinary groups of design engineers, available at all stages of a design, represented by their expert systems. This leads to a design process in which technical expertise can be shared freely in the form of each group's expert system (i.e. collaborative design). Such a design process allows various groups of design engineers to work on parts of a design problem independently, using their own expert systems, and accessing the expert systems of other disciplinary groups at those stages when group cooperation is required. This would allow one disciplinary group (i.e. process/chemical engineering) to produce a design and obtain an evaluation of the design from other disciplinary groups (i.e. mechanical/electrical engineering), without involving the people concerned. Such a design process results in a much more rapid consideration of major design alternatives, and thus improves the quality of the result, the effectiveness of the design review process, and the integrity of the final design.

A class of *AI* tools constructed along these lines is the *blackboard model*, which provides for integrated design data management, and for allowing various knowledge sources to cooperate in data development, verification and validation, as well as in information sharing (i.e. concurrent *and* collaborative design). The blackboard model is a paradigm that allows for the flexible integration of modular portions of design code into a single problem-solving environment. It is a general and simple model that enables the representation of a variety of design disciplines. Given its nature, it is prescribed for problem solving in knowledge-intensive domains that use large amounts of diverse, error-full and incomplete knowledge, therefore requiring multiple cooperation between knowledge sources in searching a large problem space—which is typical of engineering designs. In terms of the type of problems that it can solve, there is only one major assumption—that the problem-solving activity generates a set of intermediate results that contribute to the final solution.

The *blackboard model* consists of a data structure (the *blackboard*) containing information that permits a set of modules or knowledge sources to interact. The blackboard can be seen as a global database, or working memory in which distinct representations of knowledge and intermediate results are integrated uniformly.

The *blackboard model* can also be seen as a means of communication among knowledge sources, mediating all of their interactions. Finally, it can be seen as a common display, review, and performance evaluation area. It may be structured so as to represent different levels of abstraction and also distinct and/or overlapping phases in the design process. The division of the blackboard into levels parallels the process of hierarchical structuring and of abstraction of knowledge, allowing elements at each level to be described approximately as abstractions of elements at the next lower level. The partition of knowledge into hierarchical levels is useful, in that a partial solution (i.e. group of hypotheses) at one hierarchical level can be used to constrain the search at lower levels—typical of systems hierarchical structuring in engineering design. The blackboard thus provides a shared representation of a design and is composed of a hierarchy of three *panels*:

- A geometry panel, which is the lowest-level representation of the design in the form of geometric models.
- A feature panel, which is a symbolic-level representation of the design. It provides symbolic representations of features, constraints, specifications, and the design record.
- The control panel, which contains the information necessary to manage the operation of the blackboard model.

f) Implementation and Analysis

When dealing with the automated generation of solutions to design problems in a target engineering design project, it is necessary to distinguish between *design* and *performance*. The former denotes the geometric and physical properties of a solution that design engineers determine directly through their decisions to meet specific design criteria. The latter denotes those properties that are derived from combinations of design variables. In general, the relationships between design and performance variables are complex. A single design variable is likely to influence several performance variables and, conversely, a single performance variable normally depends on several design variables. For example, a system's *load and strength distributions* are indicative of the level of *stress* that the system's primary function may be subject to, as performed by the system's equipment (i.e. assemblies or components). This stress design variable is likely to influence several performance variables, such as expected failure rate or the mean time between failures.

Conversely, a single performance variable such as *system availability*, which relates to the performance variables of *reliability* and *maintainability*, all of which are concerned with the period of *time* that the system's equipment may be subject to *failure*, as measured by the variables of the mean time between failures and the mean time to repair, depends upon several design variables.

These design variables are concerned with equipment *usage* or *application* over a period of *time*, the *accessibility* and *repairability* of the system's related equipment in the event of *failure*, and the system's *load* and *strength* distributions. As a consequence, neither design nor performance variables should be considered in isolation. Whenever a design is evaluated, it should be reasonably complete (relative to the particular level of abstraction—i.e. design stage—at which it is conceived), and it should be evaluated over the entire spectrum of performance variables that are relevant for that level. Thus, for conventional engineering designs, the tendency is to separate the generation of a design from its subsequent evaluation (as opposed to optimisation, where the two processes are linked), whereas the use of an AIB blackboard model looks at preliminary design analysis and process definition concurrently with design constraints and process performance assessment.

On this basis, particularly with respect to the *design constraints* and *performance assessment*, the results of trial tests of the implementation of the AIB blackboard model in a target engineering design project are analysed to determine the applicability of automated continual design reviews throughout the engineering design process. This is achieved by defining a set of *performance measures* for each system, such as temperature range, pressure rating, output, and flow rate, according to the required design specifications identified in the *process definition*.

It is not particularly meaningful, however, to use an actual performance measure; rather, it is the proximity of the actual performance to the limits of capability (design constraints) of the system (i.e. the *safety margin*) that is more useful. In preliminary design reviews, the proximity of performance to a limit closely relates to a measure of its *safety margin*. This is determined by formulating a set of *performance constraints* for which a design solution is found that maximises the *safety margins* with respect to these performance constraints, so that a maximum safety margin is achieved with respect to all performance criteria.

Chapter 2
Design Integrity and Automation

Abstract The overall combination of the topics of reliability and performance, availability and maintainability, and safety and risk in engineering design constitutes a methodology that provides the means by which complex engineering designs can be properly analysed and reviewed. Such an analysis and review is conducted not only with a focus on individual inherent systems but also with a perspective of the critical combination and complex integration of all of the design's systems and related equipment, in order to achieve the required *design integrity*. A basic and fundamental understanding of the concepts of reliability, availability and maintainability and, to a large extent, an empirical understanding of safety have in the main dealt with statistical techniques for the measure and/or estimation of various parameters related to each of these concepts that are *based on obtained data*. However, in *designing* for reliability, availability, maintainability and safety, it is more often the case that the measures and/or estimations of various parameters related to each of these concepts are *not based on obtained data*. Furthermore, the complexity arising from an integration of engineering systems and their interactions makes it somewhat impossible to gather meaningful statistical data that could allow for the use of objective probabilities in the analysis of the integrity of engineering design. Other acceptable methods must therefore be sought to determine the integrity of engineering design in the situation where data are not available or not meaningful. Methodology in which the technical uncertainty of inadequately defined design problems may be formulated in order to achieve maximum design integrity has thus been developed to accommodate its use in conceptual and preliminary engineering design in which most of the design's systems and components have not yet been precisely defined. This chapter gives an overview of *design automation* methodology in which the technical uncertainty of inadequately defined design problems may be formulated through the application of intelligent design systems that can be used in creating or altering conceptual and preliminary engineering designs in which most of the design's systems and components still need to be defined, as well as *evaluate a design* through the use of evaluation design automation (EDA) tools.

R.F. Stapelberg, *Handbook of Reliability, Availability, Maintainability and Safety in Engineering Design*, © Springer 2009

2.1 Industry Perception and Related Research

It is obvious that most of the problems of recently constructed super-projects stem from the lack of a proper evaluation of the *integrity* of their design. Furthermore, it is obvious that a severe lack of insight exists in the essential activities required to establish a proper evaluation of the integrity of engineering design—with the consequence that many engineering design projects are subject to relatively superficial design reviews, especially with large, complex and expensive process plants.

Based on the results of cost 'blow-outs' of these super-projects, the conclusion reached is that insufficient research has been conducted in the determination of the integrity of engineering design, its application in design procedure, as well as in the severe shortcomings of current design review techniques.

2.1.1 Industry Perception

It remains a fact that, in most engineering design organisations, the designs of large engineering projects are based upon the theoretical expertise and practical experiences pertaining to chemical, civil, electrical, industrial, mechanical and process engineering, from the point of view of '*what should be achieved*' to meet the demands of various design criteria. It is apparent, though, that not enough consideration is being given to the point of view of '*what should be assured*' in the event that the demands of design criteria are not met.

As previously indicated, the tools that most design engineers resort to in determining integrity of design are techniques such as *hazardous operations (HazOp)* and *simulation*, whereas less frequently used techniques include *hazards analysis (HazAn)*, *fault-tree analysis (FTA)*, *failure modes and effects analysis (FMEA)* and *failure modes effects and criticality analysis (FMECA)*.

It unfortunately also remains a fact that most of these techniques are either misunderstood or conducted incorrectly, or not even conducted at all, with the result that many high-cost engineering 'super-projects' eventually reach the construction phase without having been subjected to a rigorous evaluation of the integrity of their designs. One of the outcomes of the research presented in this handbook has been the development of an *artificial intelligence-based (AIB)* model in which *AI* modelling techniques, such as the inclusion of *knowledge-based expert systems* within a *blackboard model*, have been applied in the development of intelligent computer automated methodology for determining the integrity of engineering design. The model fundamentally provides a capability for *automated continual design reviews* throughout the engineering design process, whereby groups of design engineers collaboratively input specific design data and schematics into their relevant knowledge-based expert systems, which are then concurrently evaluated for integrity of the design. The overall perception in industry of the benefits of such a methodology is still in its infant stages, particularly the concept of having a diverse team of experts or multidisciplinary groups of design engineers available at all stages of a design,

as represented by their knowledge-based expert systems. The potential savings in avoiding cost 'blow-outs' during engineering project construction are still not properly appreciated, and the practical implementation of a collaborative *AIB blackboard model* from conceptual design through to construction still needs further evaluation.

2.1.2 Related Research

As indicated previously, many of the methods and techniques applied in the fields of reliability, availability, maintainability and safety have been thoroughly explored by many other researchers. Some of the more significant findings of these researchers are grouped into the various topics of 'reliability and performance', 'availability and maintainability', and 'safety and risk' that are included in the theoretical overview and analytic development chapters in this handbook. Further research in the application of artificial intelligence in engineering design can be found in the comprehensive three-volume set of multidisciplinary research papers on 'Design representation and models of routine design'; 'Models of innovative design, reasoning about physical systems, and reasoning about geometry'; and 'Knowledge acquisition, commercial systems, and integrated environments' (Tong and Sriram 1992).

Research in the application of artificial intelligence in engineering design has also been conducted by authorities such as the US Department of Defence (DoD), the US National Aeronautics and Space Administration (NASA) and the US Nuclear Regulatory Commission (NUREG).

Under the topics of *reliability and performance*, some of the more recent researchers whose works are closely related to the integrity of engineering design, particularly *designing for reliability*, covered in this handbook are S.M. Batill, J.E. Renaud and Xiaoyu Gu in their simulation modelling of uncertainty in multidisciplinary design optimisation (Batill et al. 2000); B.S. Dhillon in his fundamental research into reliability engineering in systems design and design reliability (Dhillon 1999a); G. Thompson, J.S. Liu et al. in their practical methodology to designing for reliability (Thompson et al. 1999); W. Kerscher, J. Booker et al. in their use of fuzzy control methods in information integration technology (IIT) for process design (Kerscher et al. 1998); J.S. Liu and G. Thompson again, in their approach to multi-factor design evaluation through parameter profile analysis (Liu and Thompson 1996); D.D. Boettner and A.C. Ward in their use of artificial intelligence (AI) in engineering design and the application of labelled interval calculus in multi-factor design evaluation (Boettner and Ward 1992); and N.R. Ortiz, T.A. Wheeler et al. in their use of expert judgment in nuclear engineering process design (Ortiz et al. 1991). Note that all these data sources are included in the References list of Chapter 3.

Under the topics of *availability and maintainability*, some of the researchers whose works are related to the integrity of engineering design, particularly *designing for availability* and *designing for maintainability*, covered in this handbook are V. Tang and V. Salminen in their unique theory of complicatedness as a framework

for complex systems analysis and engineering design (Tang and Salminen 2001); X. Du and W. Chen in their extensive modelling of robustness in engineering design (Du and Chen 1999a); X. Du and W. Chen also consider a methodology for managing the effect of uncertainty in simulation-based design and simulation-based collaborative systems design (Du and Chen 1999b,c); N.P. Suh in his research into the theory of complexity and periodicity in design (Suh 1999); G. Thompson, J. Geominne and J.R. Williams in their method of plant design evaluation featuring maintainability and reliability (Thompson et al. 1998); A. Parkinson, C. Sorensen and N. Pourhassan in their approach to determining robust optimal engineering design (Parkinson et al. 1993); and J.L. Peterson in his research into Petri net (PN) theory and its specific application in the design of engineering systems (Peterson 1981). Note that all these data sources are included in the References list of Chapter 4.

Similarly, under the topics of *safety and risk*, some of the researchers whose works are also related to the integrity of engineering design and covered in this handbook are A. Blandford, B. Butterworth et al. in their modelling applications incorporating human safety factors into the design of complex engineering systems (Blandford et al. 1999); R.L. Pattison and J.D. Andrews in their use of genetic algorithms in safety systems design (Pattison and Andrews 1999); D. Cvetkovic and I.C. Parmee in their multi-objective optimisation of preliminary and evolutionary design (Cvetkovic and Parmee 1998); M. Tang in his knowledge-based architecture for intelligent design support (Tang 1997); J.D. Andrews in his determination of optimal safety system design using fault-tree analysis (Andrews 1994); D.W. Coit and A.E. Smith for their research into the use of genetic algorithms for optimising combinatorial design problems (Coit and Smith 1994); H. Zarefar and J.R. Goulding in their research into neural networks for intelligent design (Zarefar and Goulding 1992); S. Ben Brahim and A. Smith in their estimation of engineering design performance using neural networks (Ben Brahim and Smith 1992), as well as G. Chryssolouris and M. Lee in their use of neural networks for systems design (Chryssolouris and Lee 1989), and J.W. McManus of NASA Langley Research Center in his pioneering work on the analysis of concurrent blackboard systems (McManus 1991). Note that all these data sources are included in the References list of Chapter 5.

Recently published material incorporating integrity in engineering design are few and either focus on a single topic, predominantly reliability, safety and risk, or are intended for specific engineering disciplines, especially electrical and/or electronic engineering. Some of the more recent publications on the application of reliability, maintainability, safety and risk in industry, rather than in engineering design include N.W. Sachs' 'Practical plant failure analysis: a guide to understanding machinery deterioration and improving equipment reliability' (Sachs 2006), which explains how and why machinery fails and how basic failure mechanisms occur; D.J. Smith's 'Reliability, maintainability and risk: practical methods for engineers' (Smith 2005), which considers the integrity of safety-related systems as well as the latest approaches to reliability modelling; and P.D.T. O'Connor's 'Practical reliability engineering' (O'Connor 2002), which gives a comprehensive, up-to-date description of all the important methods for the design, development, manufacture

and maintenance of engineering products and systems. Recent publications relating specifically to design integrity include E. Nikolaidis' 'Engineering design reliability handbook' (Nikolaidis et al. 2005), which considers reliability-based design and modelling of uncertainty when data are limited.

2.2 Intelligent Design Systems

Methodology in which the technical uncertainty of inadequately defined design problems may be formulated in order to achieve maximum design integrity has been developed in this research to accommodate its use in conceptual and preliminary engineering design in which most of the design's systems and components have not yet been precisely defined. Furthermore, intelligent computer automated methodology has been developed through artificial intelligence-based (AIB) modelling to provide a means for continual design reviews throughout the engineering design process. This is progressively becoming acknowledged as a necessity, not only for use in future large process super-projects but for engineering design projects in general, particularly construction projects that incorporate various engineering disciplines dealing with, e.g. high-rise buildings and complex infrastructure projects.

2.2.1 The Future of Intelligent Design Systems

Starting from current methods in the engineering design process, and projecting our vision further to new methodologies such as AIB modelling to provide a means for continual design reviews throughout the engineering design process, it becomes apparent that there can and should be a rapid evolution of the application of intelligent computer automated methodology to future engineering designs. Currently, three generations of design tools and approaches can be enumerated: The first generation is what we currently have—a variety of tools for representing designs and design information, in many cases not integrated nor well catalogued, with the following features:

- Information flows consume much time of personnel involved.
- Engineers spend much of their time on managerial, rather than technical tasks.
- Constraints from downstream are rarely considered.

Widespread use of knowledge-based systems will rapidly be adopted, marking a second generation in which techniques become available that allow first-generation tools to be integrated, networked and coordinated.

Most companies are already fully networked and integrated. The following projections can be made for this second generation of knowledge-based systems and tools:

- Knowledge-based tools are developed to complement and replace first-generation *shells*. These are targeted for *design assistance*, rather than for general design applications, especially tools for design evaluation, selection and review problems that can be enhanced and expanded for a wide range of different engineering applications.
- Various design strategies are built into *expert system shells*, so that knowledge from new areas of engineering design can be utilised appropriately.

Projecting even further, the third generation will arise as there is widespread automation of the application of knowledge-based tools such as *design automation*, which will require advances in the application of machine learning and knowledge acquisition techniques, and the automation of new innovations in design verification and validation such as *evaluation design automation*.

The third generation will also have automated the process of applying these tools in design organisations. With each generation, the key aspects of the previous generations become ever more widespread as technology moves out of the research and development phase and into commercial products and tools.

The above projections and trends are expected in the following areas:

- Degree of integration and networking of intelligent design tools;
- Degree of automation of the application of design tool technology;
- Sophistication of general-purpose tools (shells);
- Degree of usage in engineering design organisations;
- Degree of understanding of the design process of complex systems.

2.2.2 Design Automation and Evaluation Design Automation

Research work on *design automation (DA)* has concentrated on programs that play an active role in the design process, in that they actually create or alter the design. A design automation environment typically contains a design representation or design database through which the design is controlled. Such a design automation environment usually interacts with a predetermined set of resident *computer-aided design (CAD)* tools, and will attempt to act as a manager of the *CAD* tools by handling input/output requirements and possibly automatically sequencing these *CAD* tools. Furthermore, it provides a design platform acting as a framework that, in effect, shields the designer from cumbersome details and allows for design work at a high level of abstraction during the earlier phases of the engineering design process (Schwarz et al. 2001).

Evaluation design automation (EDA) tools, on the other hand, are passive in that they *evaluate a design* in order to determine how well it performs. Evaluation design automation uses a '*frame-based*' knowledge representation to store and process expert knowledge. Frames provide a means of grouping packages of knowledge that are related to each other in some manner, where each knowledge package may have widely differing representations. The packages of knowledge are referred to

as '*slots*' in the frame. The various slots could contain knowledge such as symbolic data indicating performance values, heuristic rules indicating likely failure modes, or procedures for design review routines. The knowledge contained in these slots can be grouped according to a systems hierarchy, and the frames as such can be grouped to form a hierarchy of contexts.

Another important aspect to *EDA* is *constraint propagation*, for it is through constraint propagation that design criteria are aligned with implementation constraints. Usually, constraint propagation is achievable through *data-directed invocation*. Data-directed invocation is the mechanism that allows the design to incrementally progress as the objectives and needs of the design become apparent. In this fashion, the design constraints will change and propagate with each modification to the partial design. This is important, since the design requirements typically cannot be determined a priori (Lee et al. 1993).

The construct of Chapters 3, 4 and 5 in Part II is based upon the prediction, assessment and evaluation of reliability, availability, maintainability and safety, according to the particular engineering design phases of conceptual design, preliminary design and detail design respectively. Besides an initial introduction into engineering design integrity, the chapters are further subdivided into the related topics of theory, analysis and practical application of each of these concepts. Thus, Chapters 3, 4 and 5 include a *theoretical overview*, which gives a certain *breadth* of research into the theory covering each concept in engineering design; an insight into *analytic development*, which gives a certain *depth* of research into up-to-date analytical techniques and methods that have been developed and are currently being developed for analysis of each concept in engineering design; and an exposition of *application modelling*, whereby specific computational models have been developed and applied to the different concepts, particularly *AIB modelling* in which *expert systems* within a networked *blackboard model* are applied to determine engineering design integrity.

Part II
Engineering Design Integrity Application

Chapter 3
Reliability and Performance in Engineering Design

Abstract This chapter considers in detail the concepts of reliability and performance in engineering design, as well as the various criteria essential to designing for reliability. Reliability in engineering design may be considered from the points of view of whether a design has inherently obtained certain attributes of functionality, brought about by the properties of the components of the design, or whether the design has been configured at systems level to meet certain operational constraints based on specific design criteria. Designing for reliability includes all aspects of the ability of a system to perform. Designing for reliability becomes essential to ensure that engineering systems are capable of functioning at the required and specified levels of performance, and to ensure that less costs are expended to achieve these levels of performance. Several techniques for determining reliability are categorised under three distinct definitions, namely reliability prediction, reliability assessment and reliability evaluation, according to their applicability in determining the integrity of engineering design at the conceptual, preliminary or schematic, and detail design stages respectfully. Techniques for reliability prediction are more appropriate during conceptual design, techniques for reliability assessment are more appropriate during preliminary or schematic design, and techniques for reliability evaluation are more appropriate during detail design. This chapter considers various techniques in determining reliability in engineering design at the various design stages, through the formulation of conceptual and mathematical models of engineering design integrity in designing for reliability, and the development of computer methodology whereby the models can be used for engineering design review procedures.

3.1 Introduction

From an understanding of the concept of *integrity* in engineering design—particularly of industrial systems and processes—which includes the criteria of *reliability*, *availability*, *maintainability* and *safety* of the inherent systems and processes and their related equipment, the need arises to examine in detail what each of these

R.F. Stapelberg, *Handbook of Reliability, Availability, Maintainability and Safety in Engineering Design*, © Springer 2009

criteria implies from a theoretical perspective, and how they can be practically and successfully applied. This includes the formulation of conceptual and mathematical models of engineering design integrity in design synthesis, particularly *designing for reliability, availability, maintainability and safety*, as well as the development of intelligent computer automated methodology whereby the conceptual and mathematical models can be practically used for engineering design review procedures.

The criterion of *reliability in engineering design* may be considered from two points of view: first, whether a particular design has inherently obtained certain attributes of reliability, brought about by the properties of the *components* of the design or, second, whether the design has been configured at *systems* level to meet certain reliability constraints based on specific design criteria. The former point of view may be considered as a 'bottom-up' *assessment* in which reliability in engineering design is approached from the design's lowest level (i.e. component level) *up* the systems hierarchy to the design's higher levels (i.e. assembly, system and process levels), whereby the collective effect of all the components' reliabilities on their assemblies and systems in the hierarchy is determined.

Clearly, this approach is feasible only once all the design's components have been identified, which is well into the detail design stage. The latter viewpoint may be considered as a 'top-down' *development* in which *designing for reliability* is considered from the design's highest level (i.e. process level) *down* the systems hierarchy to the design's lowest level (i.e. component level), whereby reliability constraints placed upon systems performance are determined, which will eventually effect the system's assemblies and components in the hierarchy.

This approach does *not* depend on having to initially identify all the design's components, which is particular to the conceptual and preliminary design phases of the engineering design process. Thus, in order to develop the most applicable and practical methodology for determining the integrity of engineering design at different stages of the design process, particularly relating to the *assessment* of *reliability in engineering design*, or to the *development* of *designing for reliability* (i.e. 'bottom-up' or 'top-down' approaches in the systems hierarchy), some of the basic techniques applicable to either of these approaches need to be identified and categorised by definition, and considered for suitability in achieving the goal of reliability in engineering design.

Several techniques for determining reliability are categorised under three distinct definitions, namely *reliability prediction*, *reliability assessment* and *reliability evaluation*, according to their applicability in determining the integrity of engineering design at the conceptual, preliminary/schematic or detail design stages. It must be noted, however, that these techniques do *not* represent the total spectrum of reliability analysis, and their use in determining the integrity of engineering design is considered from the point of view of their practical application, as determined in the theoretical overview. The definitions are fundamentally qualitative in distinction, and indicate significant differences in the approaches to determining the reliability of systems, compared to that of assemblies or of components. They start from a prediction of reliability of *systems* based on a *prognosis of systems performance* under conditions subject to various failure modes (*reliability prediction*), then progress to

an estimation of reliability based on *inferences of failure of equipment* according to their statistical failure distributions (*reliability assessment*) and, finally, to a determination of reliability based on *known values of failure rates* for *components* (*reliability evaluation*).

> *Reliability prediction* in this context can be defined in its simplest form as "*estimation of the probability of successful system performance or operation*".
>
> *Reliability assessment* can be defined as "*estimation of the probability that an item of equipment will perform its intended function for a specified interval under stated conditions*".
>
> *Reliability evaluation* can be defined as "*determination of the frequency with which component failures occur over a specified period of time*".

By grouping selected reliability techniques into these three different qualitative definitions, it can be readily discerned which specific techniques, relating to each of the three terms, can practically and logically be applied to the different phases of engineering design, such as conceptual design, preliminary or schematic design, and detail design. The techniques for *reliability prediction* would be more appropriate during *conceptual design*, when alternative systems in their general context are being identified in preliminary block diagrams, such as first-run process flow diagrams (PFDs), and estimates of the probability of successful performance or operation of alternative designs are necessary. Techniques for *reliability assessment* would be more appropriate during *preliminary or schematic design*, when the PFDs are frozen, process functions defined with relevant specifications relating to specific process design criteria, and process reliability and criticality are assessed according to estimations of probability that items of equipment will perform their intended function for specified intervals under stated conditions. Techniques for *reliability evaluation* are more appropriate during *detail design*, when components of equipment are detailed, such as in pipe and instrument drawings (P&IDs), and are specified according to equipment design criteria. Equipment reliability and criticality are evaluated from a determination of the frequencies with which failures occur over a specified period of time, based on known component failure rates. It is important to note that the distinction of these three terms are not absolutely clear-cut, especially *reliability assessment* and *reliability evaluation*, and that overlap of similar concepts and techniques will occur on the boundaries between these. In general, specific reliability techniques can be logically grouped under each definition and tested for contribution to each phase of the design process.

3.2 Theoretical Overview of Reliability and Performance in Engineering Design

In general, the measure of an item's *reliability* is defined as "*the frequency with which failures occur over a specified period of time*". In the past several years, the concept of reliability has become increasingly important, and a primary concern with engineered installations of technically sophisticated equipment. *Systems reli-*

ability and the study of *reliability engineering* particularly advanced in the military and space exploration arenas in the past two decades, especially in the development of large complex systems. Reliability engineering, as it is being applied in systems and process engineering industries, originated from a military application. Increased emphasis is being placed on the reliability of systems in the current technological revolution. This revolution has been accelerated by the threat of armed conflict as well as the stress on military preparedness, and an ever-increasing development in computerisation, micro-computerisation and its application in space programs, all of which have had a major impact on the need to include reliability in the engineering design process. This accelerated technological development dramatically emphasised the consequences of *unreliability* of systems. The consequences of systems unreliability ranged from operator safety to economic consequences of systems failure and, on a broader scale, to consequences that could affect national security and human lives. A somewhat disturbing fact is that the problem of avoiding these consequences becomes more severe as equipment and systems become more technologically advanced. Reduced operating budgets, especially during global economic cut-backs, further compound the problem of systems failure by limiting the use of *back-up systems* and and units that could take over when needed, requiring primary units to function with minimum possible occurrence of failure. The problem of reliability thus becomes twofold—first, the use of increasingly sophisticated equipment in complex integrated systems and second, a limit on funding for capital investments and operating and maintenance budgets, reducing the convenience of reliance on back-up or redundant equipment. As a result, the development of sound *design for reliability* practices become essential, to ensure that engineering systems are capable of functioning at the required and specified levels of performance, and to ensure that less costs are expended to achieve the required and specified levels of performance. A significant development in the application of the concept of reliability, not only in the context of existing systems and equipment but specifically in engineering design, is *reliability analysis*.

Reliability analysis in engineering design can be applied to determine whether it would be more effective to rely on redundant systems, or to upgrade the reliability of a primary unit in order to achieve the required level of *operational capability*. Reliability analysis can also show which problem design areas are the ones in real need of attention from an operational capability viewpoint, and which ones are less critical. The effect of applying adequate reliability analysis in engineering design would be to reduce the overall procurement and operational costs, and to increase the operational *availability* and physical *reliability* of most engineering systems and processes.

Reliability analysis in engineering design incorporates various techniques that are applied for different purposes. These techniques include the following:

- *Failure definition and quantification (FDQ)*, which defines equipment conditions, analyses existing failure data history of similar systems and equipment, and develops failure frequency matrices, failure distributions, hazard rates, component safe-life limits, and establishes component age-reliability characteristics.

- *Failure modes effects and criticality analysis (FMECA)*, which determines the reliability criticality of components through the identification of the component's functions, identification of different failure modes affecting each function, identification of the consequences and effects of each failure mode on the system's function, and possible causes for each of the failure modes.
- *Fault-tree or root cause analysis (RCA)*, which determines the combinations of events that will lead to the root causes of component failure. It indicates failure modes (in branch-tree structures) and probabilities of failure occurrence.
- *Risk analysis (RA)*, which combines root cause analysis with the effects of the occurrence of catastrophic failures.
- *Failure elimination analysis (FEA)*, which determines expected repetitive failures, analyses the primary causes of these failures, and develops improvements to eliminate or to reduce the possible occurrence of these failures.

Relationship of components to systems The relationship of a *component* to an overall *system* is determined by a technique called *systems breakdown structuring* in systems engineering analysis, which will be considered in greater detail in a later chapter.

As an initial overview to the development of reliability in engineering design, consideration of only the definitions for a *system* and a *component* would suffice at this stage.

> A *system* is defined as "*a complex whole of a set of connected parts or components with functionally related properties that links them together in a systems process*".
>
> A *component* is defined as "*a constituent part or element contributing to the composition of the whole*".

Reliability of a component Reliability can be defined in its simplest form as "*the probability of successful operation*". This probability, in its simplest form, is the ratio of the number of components surviving a failure test to the number of components present at the beginning of the test. A more complete definition of reliability that is somewhat more complex is given in the USA Military Standard (M1L-STD-721B). This definition states: "*Reliability is the probability that an item will perform its intended function for a specified interval under stated conditions*". The definition indicates that reliability may not be quite as simple as previously defined. For example, the reliability of a mechanical component may be subject to added stress from vibrations. Testing for reliability would have to account for this condition as well, otherwise the calculation has no real meaning.

Reliability of a system Further complications in the determination of reliability are introduced when *system reliability* is being considered, rather than component reliability. A system consists of several components of which one or more must be working in order for the system to function. Components of a system may be connected in *series*, as illustrated below in Fig. 3.1, which implies that if one component fails, then the entire system fails.

In this case, reliability of the entire system is considered, and not necessarily the reliability of an individual component. If, in the example of the control-panel

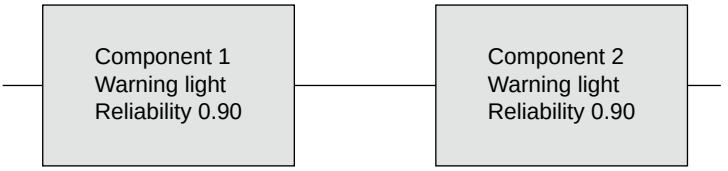


Fig. 3.1 Reliability block diagram of two components in series

warning lights, two warning lights were actually used in series for a total warning system, where each warning light had a reliability of 0.90, then the reliability of the warning system would be

$$R_{\text{System}} = R_{\text{Component 1}} \times R_{\text{Component 2}}$$
$$R_{\text{System}} = 0.90 \times 0.90 = 0.81\,.$$

The system reliability in a *series* configuration is *less* than the reliabilities of each component. This systems reliability makes use of a probability law called the *law of multiplication*.

This law states:

> "*If two or more events are independent, the probability that all events will occur is given by the product of their respective probabilities of individual occurrences*".

Thus, *series reliability* can be expressed in the following relationship

$$R_{\text{Series}} = \prod_{i=1}^{n} R_{\text{Component}\,i} \quad \forall i = 1, \ldots, n\,. \tag{3.1}$$

A realistic example is now described.

A typical high-speed reducer is illustrated below in Fig. 3.2, together with Table 3.1 listing its critical components in sequence according to configuration, and test values for the failure rates as well as the reliability values for each component. What is the overall reliability of the system, considering each component to function in a series configuration?

The consideration of a system's components to function in a series configuration, particularly with simple system configurations where inherent components are usually not redundant or where systems are single, stand-alone units with a limited number of assemblies (usually one to a maximum of three assembly sets), is preferred because systems reliability closely resembles practical usage.

A different type of system arrangement utilising two components in parallel is illustrated below in Fig. 3.3.

This system has two components that represent a *parallel* or *redundant* system where one component can serve as a back-up unit for the other in case of one *or* the other component failing. The system thus requires that only *one* component be working in order for the system to be functional. To calculate the system reliability, the individual reliabilities of each component are added together and then the

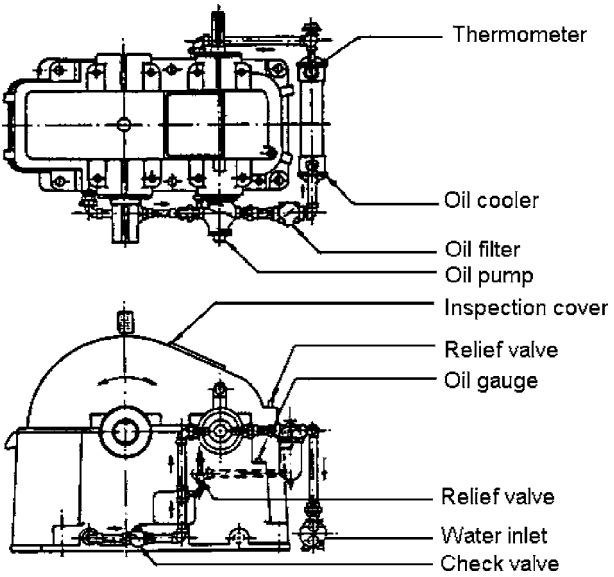


Fig. 3.2 Reliability of a high-speed self-lubricated reducer

Table 3.1 Reliability of a high-speed self-lubricated reducer

Component	Failure rate	Reliability
Gear shaft	0.01	0.99
Helical gear	0.01	0.99
Pinion	0.02	0.98
Pinion shaft	0.01	0.99
Gear bearing	0.02	0.98
Pinion bearing	0.02	0.98
Oil pump	0.08	0.92
Oil filter	0.01	0.99
Oil cooler	0.02	0.98
Housing	0.01	0.99
System	0.21[a]	0.79[b]

[a] System failure rate $= \Sigma$ (component failure rates)
[b] System reliability $= \Pi$ (component reliabilities)

product of the reliabilities in the system are subtracted. Thus, for the two components in Fig. 3.3, each with reliabilities of 0.90

$$R_{\text{System}} = (0.90 + 0.90) - (0.90 \times 0.90) = 0.99\,.$$

The system reliability of a *parallel* configuration is *greater* than the reliabilities of each individual component. This system's reliability makes use of a probability law

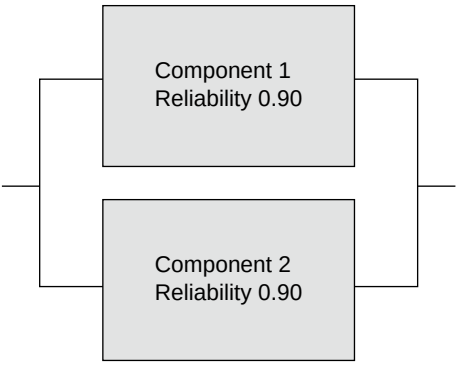


Fig. 3.3 Reliability block diagram of two components in parallel

called the *general law of addition*. This law states:

> "*If two events can occur simultaneously (i.e. in parallel), the probability that either one or both will occur is given by the sum of the individual probabilities of occurrence less the product of the individual probabilities*".

Thus, *parallel reliability* can be expressed in the following relationship

$$R_{\text{Parallel}} = \sum_{i=1}^{n} R_i - \prod_{i=1}^{n} R_i \quad \forall i = 1, \ldots, n \,. \tag{3.2}$$

The event in this case is whether a single component is working. The system is functional as long as either *one* or *both* components are working. An important point illustrated is the fact that *system configuration can have a major impact on overall systems reliability*. Thus, in engineered installations with complex integrations of system configurations, the overall impact on reliability is of critical concern in engineering design.

Parallel (or redundant) system configurations are often used where high reliability is required, as the overall result of reliability is greater than each individual component's reliability.

One of the basic concepts of reliability analysis is the fact that all systems, no matter how complex, can be reduced to a simple *series* system. For example, the two-component *series* configuration and two-component *parallel* configuration can be integrated to yield a relatively more complex system as illustrated below in Fig. 3.4.

Using the results of the previous calculations, and the probability laws of multiplication and addition, the combined system can now be reduced to a two-component system configuration, shown in Fig. 3.5.

The reliability of the *series* portion of the combined system was previously calculated to be 0.81. The reliability of the *parallel* portion of the combined system was previously calculated to be 0.99. These reliabilities are now used to represent an equivalent two-component configuration system, as illustrated in Fig. 3.5. The

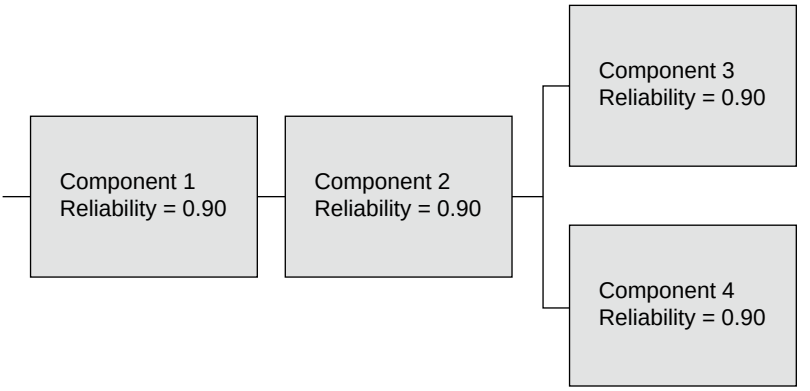


Fig. 3.4 Combination of series and parallel configuration

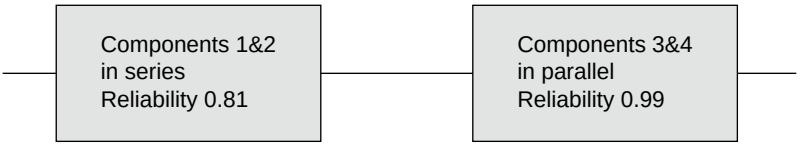


Fig. 3.5 Reduction of combination system configuration

combined systems reliability can be calculated as

$$R_{\text{Combined}} = 0.81 \times 0.99 = 0.80\,.$$

This combined systems configuration (consisting of a two-component series configuration system plus a two-component parallel configuration system), where each component has an individual reliability of 0.90, has an overall reliability that is *less* than each individual component, as well as *less* than each of its inherent two-component configuration systems. It is evident that as systems become more complex in configuration of individual components, so the reliability of the system *decreases*.

Furthermore, the more complex an engineered installation becomes with respect to complex integration of systems, the greater the probability of *unreliability*. Therefore, a greater emphasis must be placed upon the *consequences* of the unreliability of systems, especially complex systems, in designing for reliability. An even greater compounding effect on the essential need for a comprehensive approach to designing for reliability is the fact that these consequences become more severe as equipment and systems become more technologically advanced, in addition to a funding constraint placed on the number of back-up systems and units that could take over when needed.

Difference between single component and system reliabilities The reliability of the total system is of prime importance in reliability analysis for engineering design.

A system usually consists of many different components. As previously observed, these components can be structured in one of two ways, either in series or in parallel.

If components are in series, then all of the components must operate successfully for the system to function. On the other hand, if components are in parallel, only one of the components must operate for the system to be able to function either fully or partially. This is referred to as the system's *level of redundancy*. Both of these configurations need to be considered in determining how each configuration's component reliabilities will affect system reliability. System reliabilities are calculated by means of the laws of probability. To apply these laws to systems, some knowledge of the reliabilities of the inherent components is necessary, since they affect the reliability of the system. Component reliabilities are derived from tests or from actual failure history of similar components, which yield information about component failure rates. When a new component is designed, no quantitative measures of electrical, mechanical, chemical or structural properties reveal the reliability of the component. Reliability can be measured only through testing the component in a realistic simulated environment, or from actual failure history of the component while it is in use. Thus, without a quantitative probability distribution of failure data to statistically determine the measure of uncertainty (or certainty) of a component's reliability, the component's reliability remains undeterminable. This has been the opinion amongst engineers and researchers until relatively recently (Dubois et al. 1990; Bement et al. 2000b; Booker et al. 2000). With the modern application of a concept that has been postulated since the second half of the twentieth century (Zadeh 1965, 1978), the feasibility of modelling uncertainty with insufficient data, and even without any data, became a reality. This concept expounded upon modelling uncertain and vague knowledge using *fuzzy sets* as a basis for the *theory of possibility*. This qualitative concept is considered later, in detail.

The first system configuration to consider in quantitatively determining system reliability, then, is a series configuration of its components. *The problem that is of interest in this case is the manner in which system reliability decreases as the number of its components configured in series increases*.

Thus, the reliabilities of the components grouped together in a series configuration must first be calculated. Quantitative reliability calculations for such a group of components are based on two important considerations:

- Measurement of the reliability of the components must be as precise as possible.
- The way in which the reliability of the series system is calculated.

The probability law that is used for a group of series components is the product of the reliabilities of the individual components.

As an example, consider the power train system of a haul truck, illustrated in Figs. 3.6 and 3.7. The front propeller shaft is one of the components of the output shaft assembly. The output shaft assembly is adjacent to the torque converter and transmission assemblies, and these are all assemblies of the power train system. The power train system is only one of the many systems that make up the total haul truck configuration. For illustrative purposes, and simplicity of calculation, all



Fig. 3.6 Power train system reliability of a haul truck (Komatsu Corp., Japan)

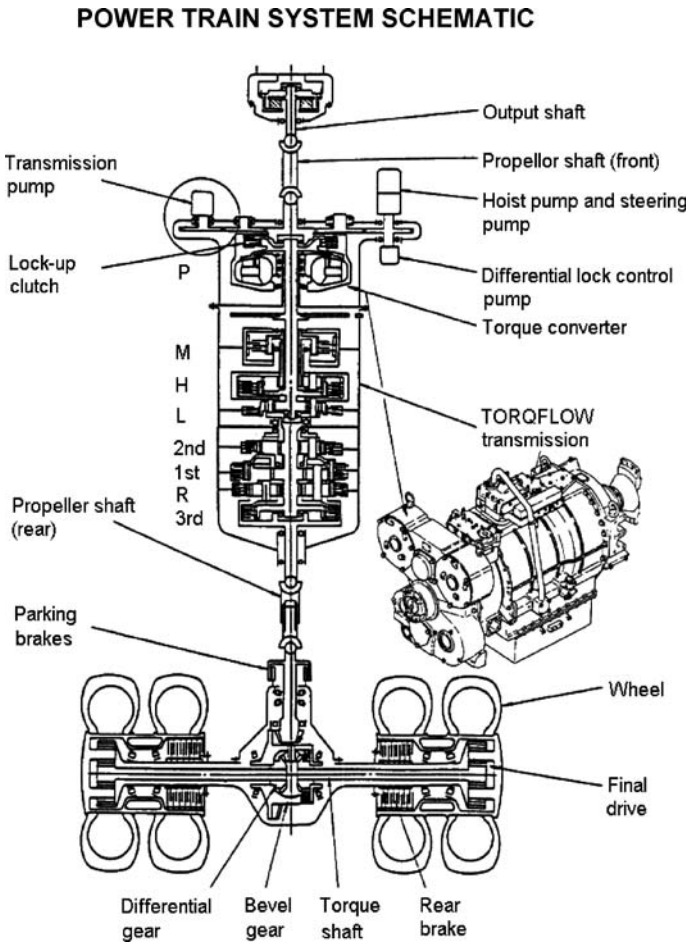


Fig. 3.7 Power train system diagram of a haul truck

Table 3.2 Power train system reliability of a haul truck

	Output shaft assembly	Transmission sub-system	Power train system
No. of components	5	50	100
Group reliability	0.99995	0.99950	0.99900
Output shaft assembly reliability		$= (0.99999)^5$	$= 0.99995$
Transmission sub-system reliability		$= (0.99999)^{50}$	$= 0.99950$
Power train system reliability		$= (0.99999)^{100}$	$= 0.99900$

components are considered to have the same reliability of 0.99999. The reliability calculations are given in Table 3.2.

The series formula of reliability implies that the reliability of a group of series components is the product of the reliabilities of the individual components. If the output shaft assembly had five components in series, then the output shaft assembly reliability would be five times the product of $0.99999 = 0.99995$. If the torque converter and transmission assemblies had a total of 50 different components, belonging to both assemblies all in series, then this sub-system reliability would be 50 times the product of $0.99999 = 0.99950$. If the power train system had a total of 100 different components, belonging to different assemblies, some of which belong to different sub-systems all in series, then the power train system's reliability would be a 100 times the product of $0.99999 = 0.99900$.

The value of a component reliability of 0.99999 implies that out of 100,000 events, 99,999 successes can be expected. This is somewhat cumbersome to envisage and, therefore, it is actually more convenient to illustrate reliability through its converse, *unreliability*. This unreliability is basically defined as

$$\text{Unreliability} = 1 - \text{Reliability}\,.$$

Thus, if component reliability is 0.99999, the unreliability is 0.00001. This implies that only one failure out of a total of 100,000 events can be expected. In the case of the haul truck, an event is when the component is used under gearshift load stress every haul cycle. If a haul cycle was an average of 15 min, then this would imply that a power train component would fail about every 25,000 operational hours. The output shaft assembly reliability of 0.99995 implies that only five failures out of a total of 100,000 events can be expected, or one failure every 20,000 events (i.e. haul cycles). (This means one assembly failure every 20,000 haul cycles, or every 5,000 operational hours.) A sub-system (power converter and transmission) reliability of 0.99950 implies that 50 failures can be expected out of a total of 100,000 events (i.e. haul cycles). (This means one sub-system failure every 2,000 haul cycles, or every 500 operational hours.) Finally, the power train system reliability of 0.99900 implies that 100 failures can be expected out of a total of 100,000 events (i.e. haul shifts). (This means one system failure every 1,000 haul cycles, or every 250 operational hours!) Note how the reliability decreases from a component reliability of only one failure in 100,000 events, or every 25,000 operational hours, to the eventual system reliability, which has 100 components in series, with 100 fail-

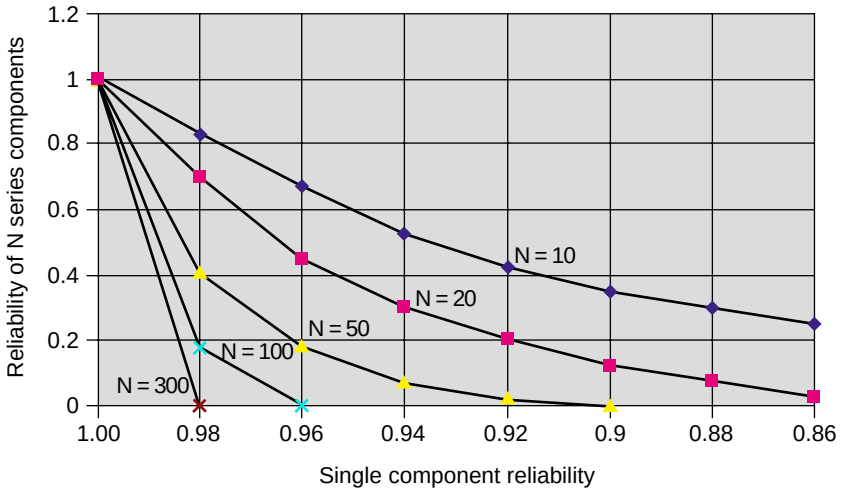


Fig. 3.8 Reliability of groups of series components

ures occurring in a total of 100,000 events, or an average of one failure every 1,000 events, or every 250 operational hours.

This decrease in system reliability is even more pronounced for lower component reliabilities. For example, with identical component reliabilities of 0.90 (in other words, one expected failure out of ten events), the reliability of the power train system with 100 components in series would be practically zero!

$$R_{\text{System}} = (0.90)^{100} \approx 0\,.$$

The following Fig. 3.8 is a graphical portrayal of how the reliability of groups of series components changes for different values of individual component reliabilities, where the reliability of each component is identical. This graph illustrates how close to the reliability value of 1 (almost 0 failures) a component's reliability would have to be in order to achieve high group reliability, when there are increasingly more components in the group.

The effect of redundancy in system reliability When very high system reliabilities are required, the designer or manufacturer must often duplicate components or assemblies, and sometimes even whole sub-systems, to meet the overall system or equipment reliability goals. In systems or equipment such as these, the components are said to be redundant, or in parallel.

Just as the reliability of a group of series components decreases as the number of components increases, so the opposite is true for redundant or parallel components. Redundant components can dramatically increase the reliability of a system. However, this increase in reliability is at the expense of factors such as weight, space, and manufacturing and maintenance costs. When redundant components are being analysed, the term unreliability is preferably used. This is because the calculations

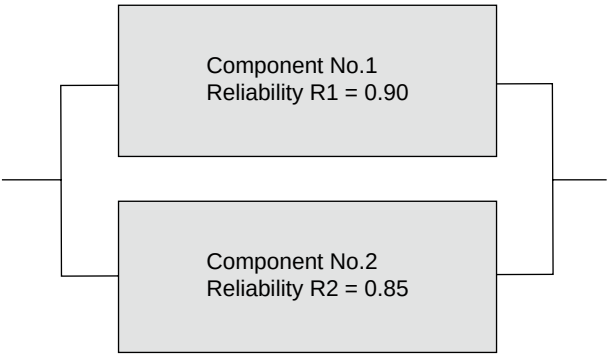


Fig. 3.9 Example of two parallel components

are easier to perform using the unreliability of a component. As a specific example, consider the two parallel components illustrated below in Fig. 3.9, with reliabilities of 0.9 and 0.85 respectively

$$\begin{aligned} \text{Unreliability:} \quad U &= (1-R1)\times(1-R2) \\ &= (0.1)\times(0.15) \\ &= 0.015 \\ \text{Reliability of group:} \quad R &= 1-\text{Unreliability} \\ &= 1-0.015 \\ &= 0.985\,. \end{aligned}$$

With the individual component reliabilities of only 0.9 (i.e. ten failures out of 100 events), and of 0.85 (i.e. 15 failures out of 100 events), the overall system reliability of these two components in parallel is increased to 0.985 (or 15 failures in 1,000 events). The improvement in reliability achieved by components in parallel can be further illustrated by referring to the graphic portrayal below (Fig. 3.10). These curves show how the reliability of groups of parallel components changes for different values of individual component reliabilities.

From these graphs it is obvious that a significant *increase* in system reliability is obtained from *redundancy*.

To cite a few examples from these graphs, if the reliability of one component is 0.9, then the reliability of two such components in parallel is 0.99. The reliability of three such components in parallel is 0.999. This means that, on average, only one *system* failure can be expected to occur out of a total of 1,000 events. Put in more correct terms, only one time out of a thousand will all three components fail in their function, and thus result in system functional failure.

Consider now an example of series and parallel assemblies in an engineered installation, such as the slurry mill illustrated below in Fig. 3.11. The system is shown with some major sub-systems. Table 3.3 gives reliability values for some of the critical assemblies and components. Consider the overall reliability of these sub-

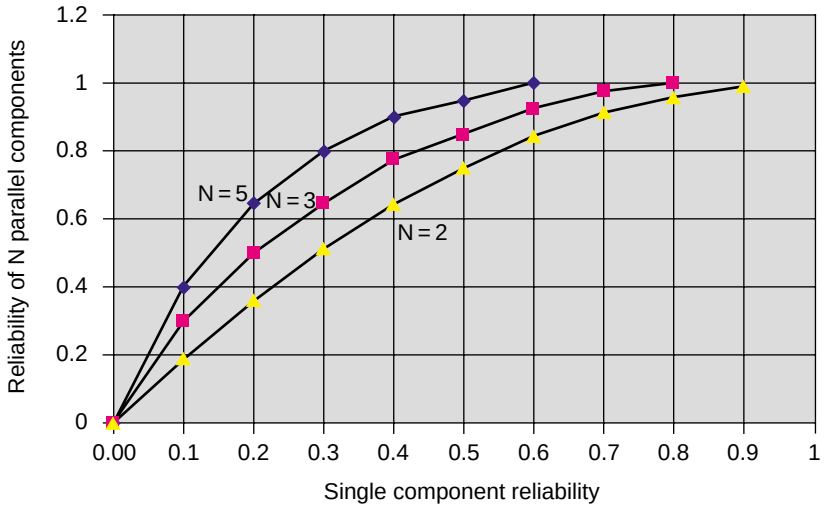


Fig. 3.10 Reliability of groups of parallel components

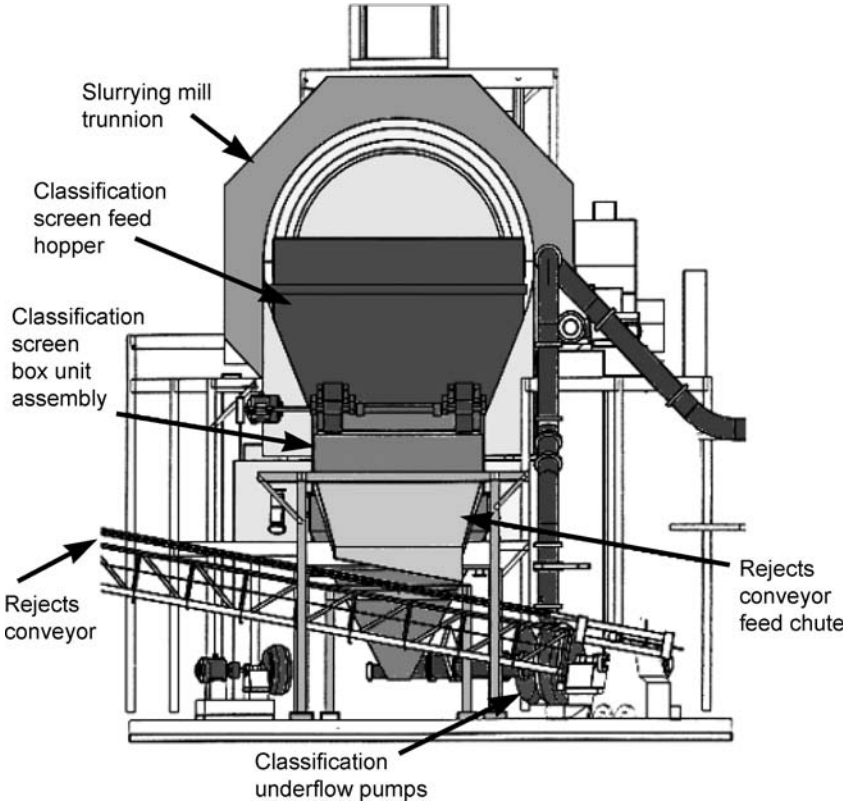


Fig. 3.11 Slurry mill engineered installation

Table 3.3 Component and assembly reliabilities and system reliability of slurry mill engineered installation

Components	Reliability
Mill trunnion	
Slurrying mill trunnion shell	0.980
Trunnion drive gears	0.975
Trunnion drive gears lube ($\times$2 units)	0.975
Mill drive	
Drive motor	0.980
Drive gearbox	0.980
Drive gearbox lube	0.975
Drive gearbox heat exchanger ($\times$2 units)	0.980
Slurry feed and screen	
Classification feed hopper	0.975
Feed hopper feeder	0.980
Feed hopper feeder motor	0.980
Classification screen	0.950
Distribution pumps	
Classification underflow pumps ($\times$2 units)	0.980
Underflow pumps motors	0.980
Rejects handling	
Rejects conveyor feed chute	0.975
Rejects conveyor	0.950
Rejects conveyor drive	0.980
Sub-systems/assemblies	
Slurry mill trunnion	0.955
Slurry mill drive	0.935
Classification	0.890
Slurry distribution	0.979
Rejects handling	0.908
Slurry mill system	
Slurry mill	0.706

systems once all of the parallel assemblies and components have been reduced to a series configuration, similar to Figs. 3.4 and 3.5.

Some of the major sub-systems, together with their major components, are the slurry mill trunnion, the slurry mill drive, classification, slurry distribution, and rejects handling.

The systems hierarchy of the slurry mill first needs to be identified in a top-level systems–assembly configuration, and accordingly is simply structured for illustration purposes:

Systems	Assemblies
Milling	Slurry mill trunnion
	Slurry mill drive
Classification	Slurry feed
	Slurry screen
Distribution	Slurry distribution pumps
	Rejects handling

Slurry mill trunnion:

$$\begin{aligned}&\text{Trunnion shell} \times \text{Trunnion drive gears} \times \text{Gears lube (2 units)}\\&= (0.980 \times 0.975) \times [(0.975 + 0.975) - (0.975 \times 0.975)]\\&= (0.980 \times 0.975 \times 0.999)\\&= 0.955\,,\end{aligned}$$

Slurry mill drive:

$$\begin{aligned}&\text{Motor} \times \text{Gearbox} \times \text{Gearbox lube} \times \text{Heat exchangers (2 units)}\\&= (0.980 \times 0.980 \times 0.975) \times [(0.980 + 0.980) - (0.980 \times 0.980)]\\&= (0.980 \times 0.980 \times 0.975 \times 0.999)\\&= 0.935\,,\end{aligned}$$

Classification:

$$\begin{aligned}&\text{Feed hopper} \times \text{Feeder} \times \text{Feeder motor} \times \text{Classification screen}\\&= (0.975 \times 0.980 \times 0.980 \times 0.950)\\&= 0.890\,,\end{aligned}$$

Slurry distribution:

$$\begin{aligned}&\text{Underflow pumps (2 units)} \times \text{Underflow pumps motors}\\&= [(0.980 + 0.980) - (0.980 \times 0.980)] \times 0.980\\&= (0.999 \times 0.980)\\&= 0.979\,,\end{aligned}$$

Rejects handling:

$$\begin{aligned}&\text{Feed chute} \times \text{Rejects conveyor} \times \text{Rejects conveyor drive}\\&= (0.975 \times 0.950 \times 0.980)\\&= 0.908\,,\end{aligned}$$

Slurry mill system:

$$\begin{aligned}&= (0.955 \times 0.935 \times 0.890 \times 0.979 \times 0.908)\\&= 0.706\,.\end{aligned}$$

The slurry mill system reliability of 0.706 implies that 294 failures out of a total of 1,000 events (i.e. mill charges) can be expected. If a mill charge is estimated to last for 3.5 h, this would mean one system failure every 3.4 charges, or about every 12 operational hours!

The staggering frequency of one expected failure every operational shift of 12 h, irrespective of the relatively high reliabilities of the system's components, has a significant impact on the approach to systems design for integrity (reliability, availability and maintainability), as well as on a proposed maintenance strategy.

3.2.1 Theoretical Overview of Reliability and Performance Prediction in Conceptual Design

Reliability and *performance prediction* attempts to estimate the probability of successful *performance of systems*. Reliability and performance prediction in this context is considered in the *conceptual design* phase of the engineering design process. The most applicable methodology for reliability and performance prediction in the conceptual design phase includes basic concepts of mathematical modelling such as:

- Total cost models for design reliability.
- Interference theory and reliability modelling.
- System reliability modelling based on system performance.

3.2.1.1 Total Cost Models for Design Reliability

In a paper titled 'Safety and risk' (Wolfram 1993), reliability and risk prediction is considered in determining the total potential cost of an engineering project. With increased design reliability (including strength and safety), project costs can increase exponentially to some cut-off point. The tendency would thus be to achieve an 'acceptable' design at the least cost possible.

a) Risk Cost Estimation

The total potential cost of an engineering project compared to its design reliability, whereby a minimum cost point designated the *economic optimum reliability* is determined, is illustrated in Fig. 3.12. Curve ACB is the normal 'first cost curve', which includes capital costs plus operating and maintenance costs. With the inclusion of the 'risk cost curve' (CD), the effect on total project cost is reflected as a concave or parabolic curve. Thus, designs of low reliability are not worth consideration because the *risk cost* is too high.

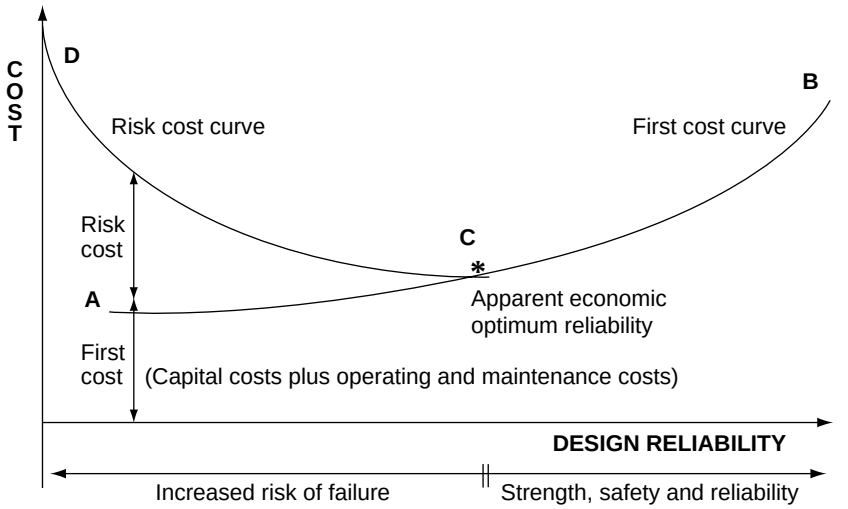


Fig. 3.12 Total cost versus design reliability

The difference between the 'risk cost curve' and the 'first cost curve' in Fig. 3.12 designates this risk cost, which is a function of the probability and consequences of *systems* failure on the project.

Thus, the risk cost can be formulated as

$$\text{Risk cost} = \text{Probability of failure} \times \text{Consequence of failure}\,.$$

This probability and consequence of *systems* failure is related to *process* reliability and criticality at the higher systems levels (i.e. process and system level) that is established in the design's systems hierarchy, or *systems breakdown structure (SBS)*.

According to Wolfram, there would thus appear to be an economically optimum level of process reliability (and safety). However, this is misleading, as the prediction of process reliability and the inherent probability of failure do not reflect reality precisely, and the extent of the error involved is uncertain. In the face of this uncertainty, there is the tendency either to be conservative and move towards higher predicted levels of design reliability, or to rely on previous designs where the individual process systems on their own were adequately designed and constructed. In the first case, this is the same as selecting larger *safety factors* when there is ignorance about how a system or structure will behave. In the latter case, the combination and integration of many previously designed systems inevitably give rise to design complexity and consequent frequent failure, where high risks of the integrity of the design are encountered.

Consequently, there is a need to develop good design models that can reflect reality as closely as possible. Furthermore, Prof. Wolfram contends that these design models need not attempt to explain wide-ranging phenomena, just the criteria relevant to the design. However, the fact that engineering design should be more precise

close to those areas where failure is more likely to occur is overlooked by most design engineers in the early stages of the design process. The questions to be asked then are: which areas are more likely to incur failure, and what would the probability of that likelihood be? The penalty for this uncertainty is a substantial increase in first costs if the project economics are feasible, or a high risk in the consequential risk costs.

b) Project Cost Estimation

Nearly every engineering design project will include some form of first cost estimating. This initial cost estimating may be performed by specific engineering personnel or by separate cost estimators. Occasionally, other resources, such as vendors, will be required to assist in first cost estimating. The engineering design project manager determines the need for cost estimating services and making arrangements for the appropriate services at the appropriate times. Ordinarily, cost estimating services should be obtained from cost estimators employed by the design engineer. First cost estimating is normally done as early as possible, when planning and scheduling the project, as well as finalising the estimating approach and nature of engineering input to be used as the basis for the cost estimate.

Types of first cost estimates First cost estimates consist basically of investment or capital costs, operating costs, and maintenance costs. These types of estimates can be evaluated in a number of ways to suit the needs of the project:

- Discounted cash flow (DCF)
- Return on investment (ROI)
- Internal rate of return (IRR)
- Sensitivity evaluations

Levels of cost estimates The most important consideration in planning cost estimating tasks is the establishment of a clear understanding as to the required level or accuracy of the cost estimate.

Basically, each level of the engineering design process has a corresponding level of cost estimating, whereby first cost estimations are usually performed during the conceptual and preliminary design phases. The following cost estimate accuracies for each engineering design phase are considered typical:

- Conceptual design phase: plus or minus 30%
- Preliminary design phase: plus or minus 20%
- Final detail design phase: plus or minus 10%

The percentages imply that the estimate will be above or below the final construction costs of the engineered installation, by that amount. Conceptual or first cost estimates are generally used for project feasibility, initial cash flow, and funding purposes by the client. Preliminary estimates that include risk costs are used for ‘go-no-go’ decisions by the client. Final estimates are used for control purposes during procurement and construction of the final design.

Cost estimating concepts The two basic categories of costs that must be considered in engineered installations are *recurring costs* and *non-recurring costs*. An example of a non-recurring cost would be the engineering design of a system from its conceptual design through preliminary design to detail design. A typical recurring cost would be the construction, fabrication or installation costs for the system during its construction/installation phase.

Estimating non-recurring costs In making cost estimates for non-recurring costs such as the engineering design of a system from its conceptual design through to final detail design, inclusive of first costs and risk costs, the project manager may assign the task of analysing the scope of *engineering effort* to the cognisant engineering design task force group leaders. This engineering effort would then be divided into two definable categories, namely a conceptual effort, and a design effort.

Conceptual effort The characteristic of conceptual effort during the conceptual design phase is that it requires creative engineering to apply new areas of technology that are probed in feasibility studies, in an attempt to solve a particular design problem. However, creative engineering contains more risk to complete as far as time and cost are concerned, and the estimates must therefore be modified by the proper risk factor.

Design effort The design effort involves straightforward engineering work in which established procedures are used to achieve the design objective. The estimate of cost and time to complete the engineering work during the preliminary design and final detail design phases can be readily derived from past experience of the design engineers, or from the history of similar projects. These estimates should eventually be accurate within 10% of completed construction costs, requiring estimates to be modified by a smaller but still significant risk factor.

Classification of engineering effort In a classification of the type of engineering effort that is required, the intended engineered installation would be subdivided into groups of discrete elements, and analysed according to block diagrams of these basic groups of elements that comprise the proposed design. The elements identified in each block would serve as a logical starting point for the *work breakdown structure (WBS)*, which would then be used for deriving the cost estimate. These elements can be grouped into:

- *Type A: engineered elements*:
  Elements requiring cost estimates for engineering design, as well as for construction/fabrication and installation (i.e. contractor items).
- *Type B: fabricated elements*:
  Elements requiring cost estimates for fabrication and installation only (i.e. vendor items or packages).
- *Type C: procured elements*:
  Elements requiring cost estimates for procurement and drafting to convey systems interface only (i.e. off-the-shelf items).

Each of the elements would then be classified as to the degree of design detail required. (That is to achieve the requirements stipulated by the design baseline identified in a design configuration management plan.) The classification is based on the degree of engineering effort required by the design engineer, and will vary in accordance with the knowledge in a particular field of technology. Those elements that require a significant amount of engineering and drafting effort are the systems and sub-systems that will be designed, built and tested, requiring detailed drawings and specifications. In most engineered installations, *type A* elements represent about 30% of all the items but account for about 70% of the total effort required.

Management review of engineering effort When the estimates for the various elements are submitted by the different engineers, a cost estimate review by task force senior engineers, the team leader, and project manager includes:

- A review of all systems to identify similar or identical elements for which redundant engineering charges are estimated.
- A review of all systems to identify elements for which a design may have been accomplished on other projects, thereby making available an off-the-shelf design instead of expending a duplicating engineering effort on the current project.
- A review of all systems to identify elements that, although different, may be sufficiently similar to warrant adopting one standard element for a maximum number of systems without compromising the performance characteristics of the system.
- A review of all systems to identify elements that may be similar to off-the-shelf designs to warrant adoption of such off-the-shelf designs without compromising the performance characteristics in any significant way.

Estimating recurring costs Some of the factors that comprise recurring cost estimates for the construction/installation phase of a system are the following:

- *Construction costs*, including costs of site establishment, site works, general construction, system support structures, on-site fabrication, inspection, system and facilities construction, water supply, and construction support services.
- *Fabrication costs*, including costs of fabricating specific systems and assemblies, setting up specialised manufacturing facilities, manufacturing costs, quality inspections, and fabrication support services.
- *Procurement costs*, including costs of acquiring material/components, warehousing, demurrage, site storage, handling, transport and inspection.
- *Installation costs*, including costs of auxiliary equipment and facilities, cabling, site inspections, installation instructions, and installation drawings.

The techniques and thinking process required to estimate the cost of engineered installations differ greatly from normal construction cost estimations. Before project engineers can begin to converge on a cost estimate for a system or facility of an engineered installation, it must be properly defined, requiring answers to the following types of questions:

What is the description and specification of each system?
What is the description and specification of each sub-system?

Pitfalls of cost estimating The major pitfalls of estimating costs for engineered installations are errors in applying the mechanics of estimating, as well as judgement errors. In deriving the cost estimate, project engineers should review the work to ensure that none of the following errors has been made:

- *Omissions and incorrect work breakdown*:
  Was any cost element forgotten in addition to the engineering, material or other costs estimated for the engineering effort? Does the work breakdown structure adequately account for all the systems/sub-systems and engineering effort required?
- *Misinterpretation of data*:
  Is the interpretation of the complexity of the engineered installation accurate? Interpretations leading to under-estimations of simplicity or over-estimations of complexity will result in estimates of costs that are either too low or too high.
- *Wrong estimating techniques*:
  The correct estimating techniques must be applied to the project. For example, the use of cost statistics derived from the construction of a similar system, and using such figures for a system that requires engineering will invariably lead to low cost estimates.
- *Failure to identify major cost elements*:
  It has been statistically established that for any system, 20% of its sub-systems will account for 80% of its total cost. Concentration on these identified sub-systems will ensure a reasonable cost estimate.
- *Failure to assess and provide for risks*:
  Engineered installations involving engineering and design effort must be tested for verification. Such tests usually involve a high expenditure to attain the final detail design specification.

3.2.1.2 Interference Theory and Reliability Modelling

Although, at the conceptual and preliminary design phases, the intention is to consider *systems* that fulfil their required performance criteria within specified limits of performance according to the functional characteristics of their constituent *assemblies*, further design considerations of process systems may include the *component* level. This is done by referring to the collective reliabilities and physical configurations of components in assemblies, depending on what level of process definition has been attained, and whether component failure rates are known. However, some component failures are not necessarily dependent upon usage over time, especially in specific cases of electrical components. In such cases, generally a failure occurs when the *stress* exceeds the *strength*. Therefore, to predict reliability of such items, the nature of the stress and strength random variables must be known. This method assumes that the probability density functions of stress and strength are known, and the variables are statistically independent.

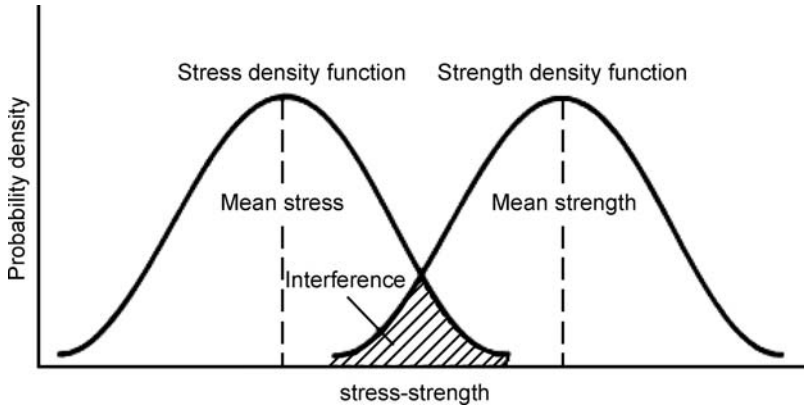


Fig. 3.13 Stress/strength diagram

A stress/strength interference diagram is shown in Fig. 3.13. The darkened area in the diagram represents the *interference* area. Besides such graphical presentation, it is also necessary to define the differences between stress and strength.

Stress is defined as "*the load which will produce a failure of a component or device*". The term *load* may be identified as mechanical, electrical, thermal or environmental effects.

Strength is defined as "*the ability of a component or device to accomplish its required function satisfactorily without a failure when subject to external load*".

Stress–strength *interference reliability* is defined as "*the probability that the failure governing stress will not exceed the failure governing strength*".

In mathematical form, this can be stated as

$$R_{\mathrm{C}} = P(s < S) = P(S > s)\,, \tag{3.3}$$

where:

$R_{\mathrm{C}} =$ the reliability of a component or a device,
$P =$ the probability,
$S =$ the strength,
$s =$ the stress.

Equation (3.3) can be rewritten in the following form

$$R_{\mathrm{C}} = \int_{-\infty}^{+\infty} f_2(s) \left[\int_{S}^{\infty} f_1(S)\,\mathrm{d}S \right] \mathrm{d}s\,, \tag{3.4}$$

where:

$f_2(s)$ is the probability density function of the stress, s
$f_1(S)$ is the probability density function of the strength, S.

Models employed to predict failure in predominantly mechanical systems are quite elementary. They are based largely on techniques developed many years ago for electronic systems and components. These models can be employed effectively for analysis of mechanical systems but they must be used with caution, since they assume that extrinsic factors such as the frequency of random shocks to the system (for example, power surges) will determine the probability of failure—hence, the assumption of Poisson distribution processes and constant hazard rates.

In research conducted into mechanical reliability (Carter 1986), it is shown that intrinsic degradation mechanisms such as fatigue, creep and stress corrosion can have a strong influence on system lifetime and the probability of failure. In highly stressed equipment, cumulative damage to specific components will be the most likely cause of failure. Hence, a review of the factors that influence degradation mechanisms such as *maintenance practice* and *operating environment* becomes a vital element in the evaluation of likely reliability performance.

To predict the probability of *system* failure, it becomes necessary to identify the various degradation mechanisms, and to determine the impact of different maintenance and operating strategies on the expected lifetimes, and level of maintainability, of the different assemblies and components in the system. The load spectrum generated by different operating and maintenance scenarios can have a significant effect on system failure probability.

When these distributions are well separated with small variances (low-stress conditions), the *safety margin* will be large and the failure distribution will tend towards the constant hazard rate (random-failure) model. In this case, the system failure probability can be computed as a function of the hazard rates for all the components in the system. For highly stressed equipment operating in hostile environments, the load and strength distributions may have a significant overlap because of the greater variance of the load distribution and the deterioration in component strength with time. Carter shows that the safety margin will then be smaller, and the tendency will be towards a weakest-link model. The probability of failure in this case can then depend on the resistance of one specific component (the weakest link) in the system.

Carter's research has been published in a number of papers and is summarised in his book *Mechanical reliability* (Carter 1986). Essentially, this work relates failure probability to the effect of the interaction between the system's *load* and *strength* distributions, as indicated in Fig. 3.14. Carter's research work also relates reliability to design (Carter 1997).

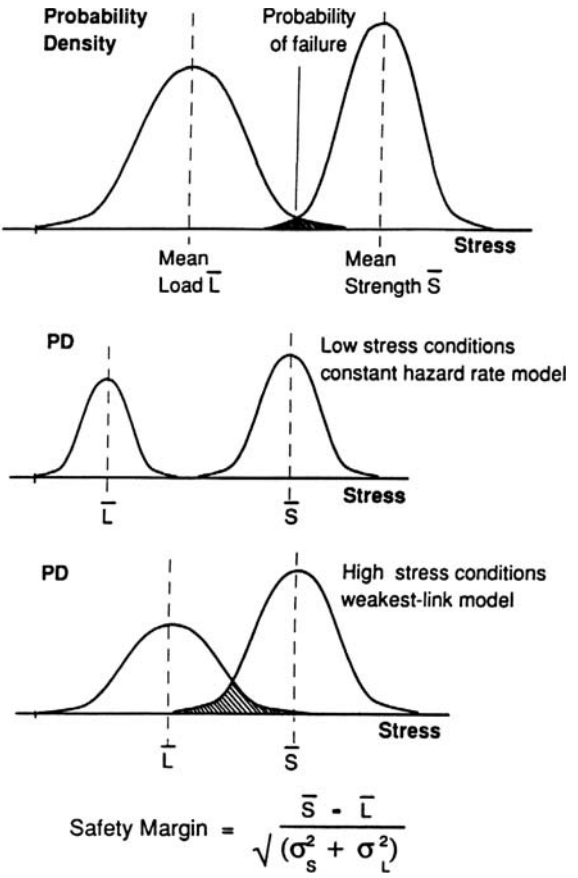


Fig. 3.14 Interaction of load and strength distributions (Carter 1986)

3.2.1.3 System Reliability Modelling Based on System Performance

The techniques for *reliability prediction* have been selected to be appropriate during conceptual design. However, at both the conceptual and preliminary design stages, it is often necessary to consider only *systems*, and not *components*, as most of the system's components have not yet been defined. Although reliability is generally described in terms of probability of failure or a mean time to failure of items of *equipment* (i.e. *assemblies* or *components*), a distinction is sometimes made between the *performance* of a *process* or *system* and its *reliability*. For example, process performance may be measured in terms of output quantities and product quality. However, this distinction is not helpful in process design because it allows for omission of *reliability prediction* from conceptual design considerations, leaving the task of evaluating reliability until detail design, when most of the equipment has been specified.

In a paper 'An approach to design for reliability' (Thompson et al. 1999), it is stated that *designing for reliability* includes all aspects of the ability of a *system* to perform, according to the following definition:

Reliability is defined as "*the probability that a device, machine or system will perform a specified function within prescribed limits, under given environmental conditions, for a specified time*".

It is apparent that a clearer distinction between systems, equipment, assemblies and components (not to mention devices and machines) needs to be made, in order to properly accommodate *reliability predictions* in engineering design reviews. Such a distinction is based upon the essential study and application of *systems engineering analysis*.

Systems engineering analysis is the study of *total systems* performance, rather than the study of the *parts*. It is the study of the complex whole of a set of connected assemblies or components and their related properties. This is feasible only through the establishment of a *systems breakdown structure (SBS)*.

The most important step in *reliability prediction* at the conceptual design stage is to consider the first item given in the list of essential preliminaries to the techniques that should be used by design engineers in determining the integrity of engineering design, namely a *systems breakdown structure* (SBS; refer to Section 1.1.1; Essential preliminaries, page 13).

a) System Breakdown Structure (SBS)

A systems breakdown structure (SBS) is a systematic hierarchical representation of equipment, grouped into its logical systems, sub-systems, assemblies, sub-assemblies and component levels. It provides visibility of process systems and their constituent assemblies and components, and allows for the whole range of reliability analysis, from *reliability prediction* through *reliability assessment* to *reliability evaluation*, to be summarised from process or system level, down to sub-system, assembly, sub-assembly and component levels.

The various levels of a systems breakdown structure are normally determined by a framework of criteria established to logically group similar components into sub-assemblies or assemblies, which are logically grouped into sub-systems or systems. This logical grouping of the constituent parts of each level of an SBS is done by identifying the actual physical design configuration of the various items of one level of the SBS into items of a higher level of systems hierarchy, and by defining common operational and physical functions of the items at each level.

Thus, from a process design integrity viewpoint, the various levels of an SBS can be defined:

- A *process* consists of one or more systems for which *overall availability* can be determined, and is dependent upon the interaction of the performance of its constituent systems.

- A *system* is a collection of sub-systems and assemblies for which *system performance* can be determined, and is dependent upon the interaction of the functions of its constituent assemblies.
- An *assembly or equipment* is a collection of sub-assemblies or components for which the values of reliability and maintainability relating to their functions can be determined, and is dependent upon the interaction of the reliabilities and physical configuration of its constituent components.
- A *component* is a collection of parts that constitutes a functional unit for which the *physical condition* can be measured and reliability can be determined.

Several different terms can be used to describe an SBS in a systems engineering context, specifically a *systems hierarchical structure*, or a *systems hierarchy*. From an engineering design perspective, however, the term SBS is usually preferred.

b) Functional Failure and Reliability

At the component level, physical condition and reliability are in most cases identical. Consider the case of a coupling. Its physical condition may be measured by its ultimate shear strength. However, the reliability of the coupling is also determined by its ability to sustain a given torque. Similar arguments may be put for other cases, such as a bolt—its measure of tensile strength and reliability in sustaining a given load, in which very little difference will be found between reliability and physical condition at the component level. When components are combined to form an assembly, they gain a collective identity and are able to perform in a manner that is usually more than the sum of their parts.

For example, a positive displacement pump is an assembly of components, and performs duties that can be measured in terms such as flow rate, pressure, temperature and power consumption. It is the ability of the assembly to carry out all these collective functions that tends to be described as the *performance*, while the *reliability* is determined by the ability of its components to resist failure. However, if the pump continues to operate but does not deliver the correct flow rate at the right pressure, then it should be regarded as having failed, because it does not fulfil its prescribed duty. It is thus incorrect to describe a pump as reliable if it does not perform the *function* required of it, according to its design. This principle is based upon a concise approach to the concept of *functional failure* whereby *reliability*, *failure* and *function* need to be defined.

According to the US Military Standard MIL-STD-721B, *reliability* is defined as "*the probability that an item will perform its intended function [without failure] for a specified interval under stated conditions*". From the same US Military Standard MIL-STD-721B, *failure* is defined as "*the inability of an item to function within its specified limits of performance*".

This means that *functional performance limits* must be clearly defined before failures can be identified. However, the task of defining functional performance limits is not exactly straightforward, especially at systems level. A complete analysis of complex systems normally requires that the functions of the various assemblies and

components of the system be identified, and that limits of performance be related to these functions.

The definition of *function* is given as "*the work that an item is designed to perform*". Failure of the item's function by definition means failure of the work or duty that the item is designed to perform.

Functional failure can thus be defined as "*the inability of an item to carry-out the work that it is designed to perform within specified limits of performance*".

From the definition, two *degrees of severity* for functional failure can be discerned:

- A *complete loss of function*, where the item cannot carry out any of the work that it was designed to perform.
- A *partial loss of function*, where the item is unable to function within specified limits of performance.

From the definitions, a concise definition of reliability can be considered:

Reliability may be defined as "*the probability that an item is able to carry-out the work that it is designed to perform within specified limits of performance for a specified interval under stated conditions*".

An important part of this definition of reliability *is the ability to perform within specified limits*. Thus, from the point of view of the degrees of severity of functional failure, no distinction is made between *performance* and *reliability* of assemblies where *functional characteristics* and *functional performance limits* can be clearly defined. Design considerations of process systems may refer to the component level and/or to the collective reliabilities and physical configurations of components in assemblies, depending on what level of process definition has been attained. However, at the conceptual or preliminary design stages, the intention is to consider *systems* that fulfil their required performance criteria within specified limits of performance according to the functional characteristics of their constituent *assemblies*.

c) Functional Failure and Functional Performance

A method in which design problems may be formulated in order to achieve maximum reliability (Thompson et al. 1999) has been adapted and expanded to accommodate its use in preliminary design, in which most of the system's components have not yet been defined. The method integrates functional failure and functional performance considerations so that a maximum *safety margin* is achieved with respect to all performance criteria. The most significant advantage of this method is that *it does not rely on failure data*. Also, provided that all the *functional performance limits* can be defined, it is possible to compute a multi-objective optimisation to determine an optimal solution.

The conventional reliability method would be to specify a minimum failure rate and to select appropriate components with individual failure rates that, when combined, achieve the required reliability. This method is, of course, reasonable provided that dependable failure rates are available. In many cases, however, none are

known with confidence, and a quantified approach to *designing for reliability that does not require failure rate data* is proposed. The approach taken is to define performance objectives that, when met, achieve an optimum design with regard to overall reliability by ensuring that the system has no 'weak links', whether the weaknesses are defined functional failures, or a failure of the system to meet the required performance criteria. The choice of *functional performance limits* is made with respect to the knowledge of loading conditions, the consequences of failure, as well as reliability expectations. If the knowledge of loading conditions is incomplete, which would generally be the case for conceptual or preliminary design, the approach to *designing for reliability* would be to use high *safety margins*, and to adopt limits of acceptable performance that are well clear of any failure criteria. Where precise data may not be available, it is clear from the previous consideration of strength and load distributions under interference theory and reliability modelling that the strength should be separated from the load by as much as possible, in order to maximise the *safety margin* in relation to certain performance criteria.

However, in cases where confidence can be placed on accurate loading calculations, as with the modelling situations considered in interference theory or in reliability modelling, then acceptable performance levels can be selected at high stress levels so that all the *components* function near their limits, resulting in a high performance *system*. If, on the other hand, it is required to reduce a *safety margin* with respect to a particular failure criterion in order to introduce a 'weak link', then the limits of acceptable performance can be modified accordingly. By the use of sets of constraints that describe the boundaries of the limits of acceptable performance, a feasible design solution will lie within the space bounded by these constraints. The most reliable design solution would be the solution that is the furthest away from the constraints, and a design that has the highest *safety margin* with respect to all constraints is the most reliable. The objective, then, is to produce a design that has the highest possible *safety margin* with respect to all constraints. However, since these constraints will be defined in different units, and because many different constraints may apply, consideration of a method of measurement is required that will yield common, non-dimensional performance measures that can be meaningfully combined. A method of *data point generation* based on limits of performance has been developed for general design analysis to determine various design alternatives (Liu et al. 1996).

3.2.2 Theoretical Overview of Reliability Assessment in Preliminary Design

Reliability assessment attempts to estimate the expected reliability and criticality values for each individual *system* or *assembly* at the upper systems levels of the systems breakdown structure (SBS). This is done without any difficulty, not only for relatively simple initial system configurations but for progressively more complex integrations of systems as well. Reliability assessment ranges from estimations of

the reliability of relatively simple systems with series and parallel *assemblies*, to estimations of the reliability of multi-state systems with random failure occurrences and repair times (i.e. constant failure and repair rates) of inherent independent assemblies.

Reliability assessment in this context is considered during the *preliminary* or *schematic design* phase of the engineering design process, with an estimation of the probability that items of *equipment* will perform their intended function for specified intervals under stated conditions.

The most applicable methods for reliability assessment in the preliminary design phase include concepts of mathematical modelling such as:

- Markov modelling:
  To estimate the reliability of multi-state *systems* with constant failure and repair rates of inherent independent *assemblies*.
- The binomial method:
  To assess the reliability of simple *systems* of series and parallel *assemblies*.
- Equipment aging models:
  To assess the aging of *equipment* at varying rates of degradation in engineered installations.
- Failure modes and effects analysis/criticality analysis:
  A step-by-step procedure for the assessment of failure effects and criticality in *equipment* design.
- Fault-tree analysis:
  To analyse the causal relationships between *equipment* failures and *system* failure, leading to the identification of specific critical *system* failure modes.

3.2.2.1 Markov Modelling (Continuous Time and Discrete States)

This method can be used in more cases than any other technique (Dhillon 1999a). Markov modelling is applicable when modelling assemblies with dependent failure and repair modes, and can be used for modelling multi-state systems and common-cause failures without any conceptual difficulty.

The method is more appropriate when system failure and repair rates are constant, as problems may arise when solving a set of linear algebraic equations for large systems where system failure and repair rates are variable. The method breaks down for a system that has non-constant failure and repair rates, except in the case of a few special situations that are not relevant to applications in engineering design. In order to formulate a set of Markov state equations, the rules associated with transition probabilities are:

a) The probability of more than one transition in time interval Δt from one state to the next state is negligible.

b) The transitional probability from one state to the next state in the time interval Δt is given by $\lambda \Delta t$, where λ is the *constant failure rate* associated with the Markov states.
c) The occurrences are independent.

A system state space diagram for *system* reliability is shown in Fig. 3.15. The state space diagram represents the transient state of a system, with system transition from state 0 to state 1. A state is transient if there is a positive probability that a system will not return to that state.

As an example, an expression for *system* reliability of the system state space shown in Fig. 3.15 is developed with the following Eqs. (3.5) and (3.6)

$$P_0(t+\Delta t) = P_0(t)[1-\lambda \Delta t]\,, \tag{3.5}$$

where:

$P_0(t)$ is the probability that the system is in operating state 0 at time t.
λ is the constant failure rate of the system.
$[1-\lambda \Delta t]$ is the probability of no failure in time interval Δt when the system is in state t.
$P_0(t+\Delta t)$ is the probability of the system being in operating state 0 at time $t+\Delta t$.

Similarly,

$$P_1(t+\Delta t) = P_0(t)[\lambda \Delta t] + P_1(t)\,, \tag{3.6}$$

where:

$P_0(t)$ denotes the probability that the system is in failed state 0 in time Δt.

In the limiting case, Eqs. (3.5) and (3.6) become

$$\lim_{\Delta t \to 0} \frac{P_0(t+\Delta t) - P_0(t)}{\Delta t} = \frac{\mathrm{d}P_0(t)}{\mathrm{d}t} = \lambda P_0(t)\,. \tag{3.7}$$

The initial condition is that when

$$\lim_{\Delta t \to 0} \frac{P_1(t+\Delta t) - P_1(t)}{\Delta t} = \frac{\mathrm{d}P_1(t)}{\mathrm{d}t} = \lambda P_0(t)\,, \tag{3.8}$$

where: $t = 0$, $P_0(0) = 1$, and $P_1(0) = 0$.

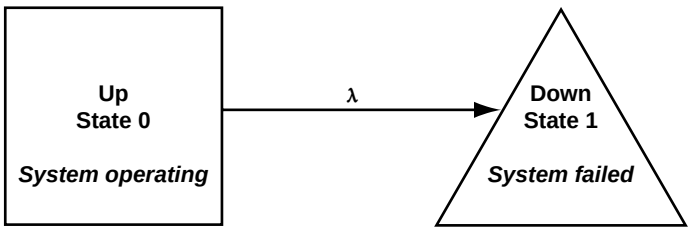


Fig. 3.15 System transition diagram

Solving Eqs. (3.7) and (3.8) by using Laplace transforms

$$P_0(s) = \frac{1}{s+\lambda} \tag{3.9}$$

and

$$P_1(s) = \frac{\lambda}{s+\lambda} \, . \tag{3.10}$$

By using the inverse transforms, Eqs. (3.9) and (3.10) become

$$P_0(t) = \mathrm{e}^{-\lambda_t} \, , \tag{3.11}$$

$$P_1(t) = 1 - \mathrm{e}^{-\lambda_t} \, . \tag{3.12}$$

Markov modelling is a widely used method to assess the reliability of systems in general, when the system's failure rates are constant. For many systems, the assumption of constant failure rate may be acceptable. However, the assumption of a *constant repair rate* may not be valid in just as many cases.

This situation is considered later in Chapter 4, Availability and Maintainability in Engineering Design.

3.2.2.2 The Binomial Method

This technique is used to assess the reliability of relatively simple systems with series and parallel *assemblies*. For reliability assessment of such *equipment*, the *binomial method* is one of the simplest techniques.

However, in the case of complex systems with many configurations of assemblies, the method becomes a trying task. The technique can be applied to systems with independent identical or non-identical assemblies.

Various types of quantitative probability distributions are applied in reliability analysis. The binomial distribution specifically has application in combinatorial reliability problems, and is sometimes referred to as a Bernoulli distribution. The binomial or Bernoulli probability distribution is very useful in assessing the probabilities of outcomes, such as the total number of failures that can be expected in a sequence of trials, or in a number of equipment items.

The mathematical basis for the technique is the following

$$\prod_{i=1}^{k} (R_i + F_i) \, , \tag{3.13}$$

where:

k is the number of non-identical assemblies
R_i is the ith assembly reliability
F_i is the ith assembly unreliability.

This technique is better understood with the following examples:

Develop reliability expressions for (a) a series system network and (b) a parallel system network with two non-identical and independent assemblies each.

Since $k = 2$, from Eq. (3.13) one obtains

$$(R_1 + F_1)(R_2 + F_2) = R_1R_2 + R_1F_2 + R_2F_1 + F_1F_2 \; . \tag{3.14}$$

a) Series Network

For a series network with two assemblies, the reliability R_S is

$$R_S = R_1R_2 \; . \tag{3.15}$$

Equation (3.15) simply represents the first right-hand term of Eq. (3.14).

b) Parallel Network

Similarly, for a parallel network with two assemblies, the reliability R_P is

$$R_P = R_1R_2 + R_1F_2 + R_2F_1 \, . \tag{3.16}$$

Since $(R_1 + F_1) = 1$ and $(R_2 + F_2) = 1$, the above equation becomes

$$R_P = R_1R_2 + R_1(1 - R_2) + R_2(1 - R_1) \; . \tag{3.17}$$

By rearranging Eq. (3.17), we get

$$\begin{aligned} R_P &= R_1R_2 + R_1 - R_1R_2 + R_2 - R_1R_2 \\ R_P &= R_1 + R_2 - R_1R_2 \\ R_P &= 1 - (1 - R_1)(1 - R_2) \; . \end{aligned} \tag{3.18}$$

This progression series can be similarly extended to a k assembly system.

The binomial method is fundamentally a statistical technique for establishing estimated reliability values for series or parallel network systems. The confidence level of *uncertainty* of the estimate is assessed through the *maximum-likelihood* technique. This technique finds good estimates of the parameters of a probability distribution obtained from available data.

Properties of maximum-likelihood estimates include the concept of *efficiency* in its comparability to a 'best' estimate with minimum variance, and *sufficiency* in that the summary statistics upon which the estimate is based essentially contains sufficient available data. This is a problem with many preliminary designs where the estimates are not always unbiased, in that the sum of the squares of the deviations from the mean is, in fact, a biased estimate.

3.2.2.3 Equipment Aging Models

A critical need for high reliability has particularly existed in the design of weapons and space systems, where the lifetime requirement (5 to 10 years) has been relatively short compared to the desired lifetime for systems in process designs such as nuclear power plant (up to 30 years). In-service aging due to stringent operational conditions can lead to simultaneous failure of redundant systems, particularly safety systems, with an essential need for functional operability in high-risk processes and systems, such as in nuclear power plants (IEEE Standard 323-1974). Because it is the most prevalent source of potential *common failure mechanisms*, *equipment aging* merits attention in reviewing reliability models for use in *designing for reliability* and in qualifying equipment for use in *safety systems*.

Although it is acknowledged that *random failures* are not likely to cause simultaneous failure of redundant safety systems, and this type of failure does not automatically lead to rejection of the equipment being tested, great care needs to be taken in understanding *random failure* in order to provide assurance that it is, in fact, not related to a deficiency of design or manufacture. Aging occurs at varying rates in engineering systems, from the time of manufacture to the end of useful life and, under some circumstances, it is important to assess the aging processes.

Accelerated aging is the general term used to describe the simulation of aging processes in the short time. At present, no well-defined accelerated aging methodology exists that may be applied generally to all process equipment. The specific problem is determining the possibility of a link between aging or deterioration of a component, such as a safety-related device, and operational or environmental stress. If such a link is present in the redundant configuration of a safety system, then this can result in a *common failure mode*, where the common factor is aging. Figure 3.16 below illustrates how the risk of *common failure mode* is influenced by stress and time (EPRI 1974). The risk function is displayed by the surface, 0*t*PS. As both stress and time-at-stress increase, the *risk* increases. P is the point of maximum

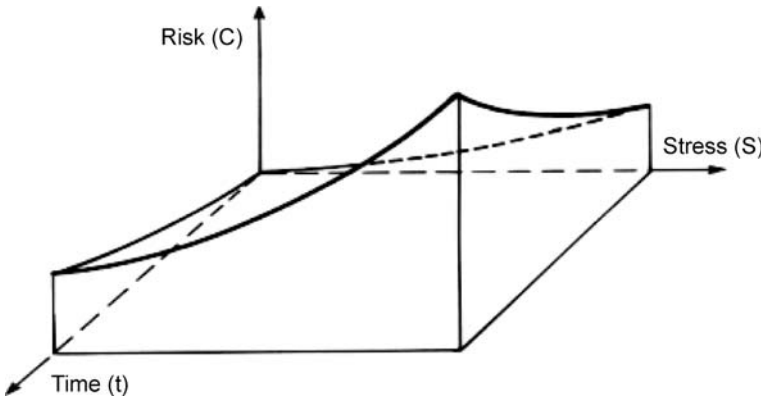


Fig. 3.16 Risk as a function of time and stress

common failure mode risk, which occurs when both stress and time are at a maximum. However, the risk occurring in and around point P cannot be evaluated by either reliability analysis or high-stress exposure tests alone. In this region, it may be necessary to resort to accelerated aging followed by design criteria conditions to evaluate the risk. This requires an understanding of the basic aging process of the equipment's material.

Generally, aging information is found for relatively few materials. Practical methods for the simulation of accelerated aging are limited to a narrow range of applications and, despite research in the field, would not be practically suited for use in *designing for reliability* (EPRI 1974).

3.2.2.4 Failure Modes and Effects Analysis (FMEA)

Failure modes and effect analysis (FMEA) is a powerful reliability assessment technique developed by the USA defence industry in the 1960s to address the problems experienced with complex weapon-control systems. Subsequently, it was extended for use with other electronic, electrical and mechanical equipment. It is a step-by-step procedure for the assessment of failure effects of potential failure modes in equipment design. FMEA is a powerful design tool to analyse engineering systems, and it may simply be described as an analysis of each *failure mode* in the system and an examination of the results or effects of such failure modes on the system (Dhillon 1999a). When FMEA is extended to classify each potential failure effect according to its *severity* (this incorporates documenting catastrophic and critical failures), so that the criticality of the *consequence* or the severity of failure is determined, the method is termed a *failure mode effects and criticality analysis (FMECA)*.

The strength of FMEA is that it can be applied at different systems hierarchy levels. For example, it can be applied to determine the performance characteristics of a gas turbine power-generating *process* or the functional failure probability of its fire protection *system*, or the failure-on-demand probability of the duty of a single pump *assembly*, down to an evaluation of the failure mechanisms associated with a pressure switch *component*. By the analysis of individual failure modes, the *effect* of each failure can be determined on the operational functionality of the relevant systems hierarchy level. FMEAs can be performed in a variety of different ways depending on the objective of the assessment, the extent of systems definition and development, and the information available on a system's assemblies and components at the time of the analysis. A different FMEA focus may dictate a different worksheet format in each case; nevertheless, there are two basic approaches for the application of FMEAs in engineering design (Moss et al. 1996):

- The *functional FMEA*, which recognises that each *system* is designed to perform a number of *functions* classified as outputs. These outputs are identified, and the losses of essential inputs to the item, or of internal failures, are then evaluated with respect to their effects on *system* performance.

- The *equipment FMEA*, which sequentially lists individual *equipment* items and analyses the effect of each *equipment failure mode* on the performance of the system.

In many cases, a combination of these two approaches is employed. For example, a functional analysis at a major *systems* level is employed in the initial functional, 'broad-brush' analysis during the preliminary design phase, which is then followed by more detailed analysis of the *equipment* identified as being more sensitive to the range of uncertainties in meeting certain design criteria during the detail design phase.

a) Types of FMEA and Their Associated Benefits

FMEA may be grouped under three distinct classifications according to application (Grant Ireson et al. 1996):

- *Design-level FMEA*
- *System-level FMEA*
- *Process-level FMEA.*

Design-level FMEA The intention of this type of FMEA is to validate the design parameters chosen for a specified functional performance requirement. The advantages of performing design-level FMEA include identification of potential design-related failure modes at system/sub-system/component level; identification of important characteristics of a given design; documentation of the rationale for design changes to guide the development of future designs; help in the design requirement objective evaluation; and assessment of design alternatives during the preliminary and detail phases of the engineering design process. FMEA is a systematic approach to reduce criticality and risk, and a useful tool to establish priority for design improvement in designing for reliability during the preliminary design phase.

System-level FMEA This is the highest-level FMEA that is performed in a systems hierarchy, and its purpose is to identify and prevent failures related specifically to systems/sub-systems during the early preliminary design phase of the engineering design process. Furthermore, this type of FMEA is carried out to validate that the system design specifications will, in fact, reduce the risk of functional failure to the lowest systems hierarchy level during the detail design phase. A primary benefit of the system-level FMEA is the identification of potential systemic failure modes due to system interaction with other systems in complex integrated designs.

Process-level FMEA This identifies and prevents failures related to the manufacturing/assembly process for certain equipment during the construction/installation stage of an engineering design project. The benefits of this detail design phase FMEA include identification of potential failure modes at equipment level, and the development of priorities and documentation of rationale for any essential design changes, to help guide the manufacturing and assembly process.

b) Steps for Performing FMEA

FMEA can be performed in six steps based on the key concepts of systems hierarchy, operations, functions, failure mode, effects, potential failure and prevention. These steps are given in the following logical sequence (Bowles et al. 1994):

FMEA sequential steps

- Identify the relevant hierarchical levels, and define systems and equipment.
- Establish ground rules and assumptions, i.e. operational phases.
- Describe systems and equipment functions and associated functional blocks.
- Identify possible failure modes and their associated effects.
- Determine the effect of each item's failure for every failure mode.
- Identify methods for detecting potential failures and avoiding functional failures.
- Determine provision for design changes that would prevent functional failures.

c) Advantages and Disadvantages of FMEA

There are many benefits of performing FMEA, particularly in the effective analysis of complex systems design, in comparing similar designs and providing a safeguard against repeating the same mistakes in future designs, and especially to improve communication among design interface personnel (Dhillon 1999a). However, an analysis of several industry-conducted FMEAs (Bull et al. 1995) showed that the timescale involved in properly developing FMEA often exceeds the preliminary/detail design phases. It is common that the results from an FMEA can be delivered to the client only with or, possibly, even after the development of the system itself. An automated approach is therefore essential.

3.2.2.5 Failure Modes and Effects Criticality Analysis (FMECA)

The objective of criticality assessment is to prioritise the failure modes discovered during the FMEA on the basis of their effects and consequences, and likelihood of occurrence. Thus, for making an assessment of equipment criticality during preliminary design, two commonly used methods are the:

- *Risk priority number (RPN) technique* used in general industry,
- *Military standard technique* used in defence, nuclear and aerospace industries.

Both approaches are briefly described below (Bowles et al. 1994).

a) The RPN Technique

This method calculates the risk priority number for a component failure mode using three factors:

- Failure effect severity.
- Failure mode occurrence probability.
- Failure detection probability.

More specifically, the risk priority number is computed by multiplying the rankings (i.e. 1–10) assigned to each of these three factors. Thus, mathematically the risk priority number is expressed by the relationship

$$\mathrm{RPN} = (\mathrm{OR})(\mathrm{SR})(\mathrm{DR}), \tag{3.19}$$

where:

RPN = the risk priority number.
OR = the occurrence ranking.
SR = the severity ranking.
DR = the detection ranking.

Since the three factors are assigned rankings from 1 to 10, the RPN will vary from 1 to 1,000. Failure modes with a high RPN are considered to be more critical; thus, they are given a higher priority in comparison to the ones with lower RPN. Specific ranking values used for the RPN technique are indicated in Tables 3.4, 3.5 and 3.6 for failure detection, failure mode occurrence probability, and failure effect severity respectively (AMCP 706-196 1976).

Table 3.4 Failure detection ranking

Item	Likelihood of detection and meaning	Rank
1	Very high—potential design weakness will be detected	1, 2
2	High—good chance of detecting potential design weakness	3, 4
3	Moderate—possible detection of potential design weakness	5, 6
4	Low—potential design weakness is unlikely to be detected	7, 8
5	Very low—potential design weakness probably not detected	9
6	Uncertain—potential design weakness cannot be detected	10

Table 3.5 Failure mode occurrence probability

Item	Ranking term	Ranking meaning	Occurrence probability	Rank value
1	Remote	Occurrence of failure is quite unlikely	<1 in 10^6	1
2	Low	Relatively few failures are expected	1 in 20,000	2
			1 in 4,000	3
3	Moderate	Occasional failures are expected	1 in 1,000	4
			1 in 400	5
			1 in 80	6
4	High	Repeated failures will occur	1 in 40	7
			1 in 20	8
5	Very high	Occurrence of failure inevitable	1 in 8	9
			1 in 2	10

Table 3.6 Severity of the failure mode effect

Item	Failure effect severity	Severity category description	Rank value
1	Minor	No effect on system performance, and the failure may not even be noticed	1
2	Low	The occurrence of failure will cause only a slight dissatisfaction if observed (i.e. potential loss)	2, 3
3	Moderate	Some dissatisfaction will be caused by failure	4–6
4	High	High degree of dissatisfaction will be caused by failure but the failure itself does not involve safety or even a non-compliance to safety regulations	7, 8
5	Very high	The failure affects safe item operation, and involves significant non-compliance with safety regulations	9, 10

b) The Military Standard Technique

This technique is used in military defence, aerospace and nuclear industries, to prioritise the failure modes of the item under consideration so that appropriate corrective measures can be undertaken (MIL-STD-1629). The technique requires the categorisation of the failure mode effect severity and then the development of a critical ranking. Table 3.7 presents classifications of failure mode effect severity. In order to assess the likelihood of a failure mode occurrence, either a qualitative or a quantitative approach can be used. The qualitative method is used when there are no specific failure rate data. In this approach, the individual occurrence probabilities are grouped into distinct, logically defined levels that establish the qualitative failure probabilities. Table 3.8 presents occurrence probability levels (MIL-STD-1629).

A *criticality matrix* is developed as shown in Fig. 3.17, for identifying and comparing each failure mode to all other failure modes with respect to severity. The criticality matrix is developed by inserting values in matrix locations denoting the severity classification, and either the criticality number K_i for the failure modes of an item, or the occurrence level probability. The distribution of criticality of item failure modes is depicted by the resulting matrix, and serves as a useful tool for assigning design review priorities.

The direction of the arrow originating from the origin, shown in Fig. 3.17, indicates the increasing criticality of the item failure, and the hatching in the figure shows the approximate desirable design region. For severity classifications A and B, the desirable design region has low occurrence probability or criticality number. On the other hand, for severity classifications C and D failures, higher probabilities of occurrence can be tolerated. Nonetheless, failure modes belonging to classifications A and B should be eliminated altogether or at least their probabilities of occurrence be reduced to an acceptable level through design changes. The quantitative approach is used when failure mode and probability of occurrence data are available. Thus, the failure mode critical number is calculated using

$$K_{\mathrm{fm}} = F\theta\lambda T\ , \tag{3.20}$$

Table 3.7 Failure mode effect severity classifications

Item	Classification	Description	No.
1	Catastrophic	The occurrence of failure may result in death or equipment loss	A
2	Critical	The occurrence of failure may result in severe injury or major system damage leading to loss	B
3	Marginal	The occurrence of failure may result in minor injury or minor system damage leading to loss	C
4	Minor	The failure is not serious enough to lead to injury or system damage, but it will result in repair or in unscheduled maintenance	D

Table 3.8 Qualitative failure probability levels

Item	Probability level	Term	Description
1	I	Frequent	High probability of occurrence during the item operational period
2	II	Reasonably probable	Moderate probability of occurrence during the item operational period
3	III	Occasional	Occasion probability of occurrence during the item operational period
4	IV	Remote	Unlikely probability of occurrence during the item operational period
5	V	Extremely unlikely	Zero chance of occurrence during the item operational period

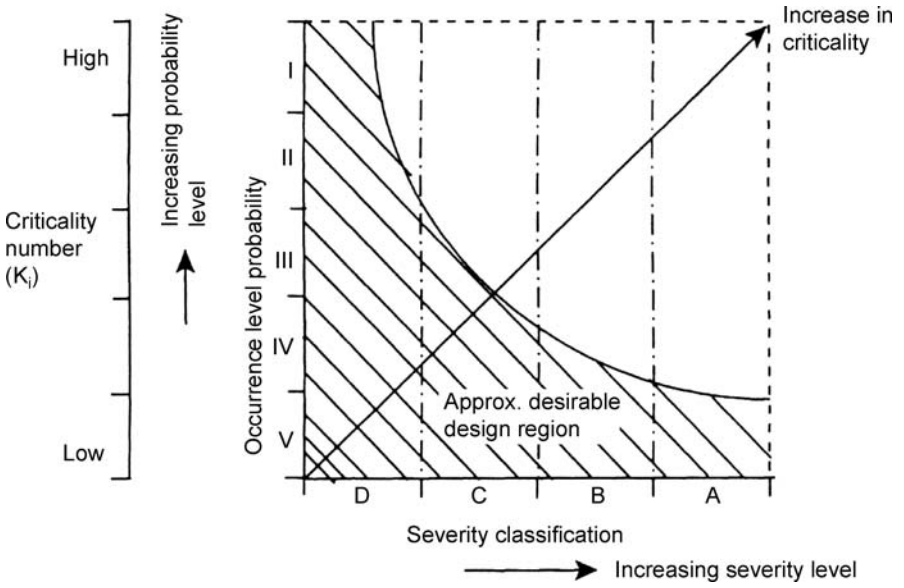


Fig. 3.17 Criticality matrix (Dhillon 1999)

Table 3.9 Failure effect probability guideline values

Item no.	Failure effect description	Probability value of F
1	No effect	0
2	Actual loss	1.0
3	Probable loss	$0.10 < F < 1.00$
4	Possible loss	$0 < F < 0.10$

where:

K_{fm} is the failure mode criticality number.
$\theta =$ the failure mode ratio or the probability that a component will fail in the particular failure mode of interest. More specifically, it is the fraction of the component failure rate that can be allocated to the failure mode under consideration. When all failure modes of a component are specified, the sum of the allocations equals unity.
$F =$ the conditional probability that the failure effect results in the indicated severity classification or category, given that the failure mode occurs. The values of F are based on an analyst's judgment, and these values are quantified according to Table 3.9.
$T =$ is the operational time expressed in hours or cycles.
$\lambda =$ is the component failure rate.

The item criticality number K_i is calculated separately for each severity class. Thus, the total of the criticality numbers of all the failure modes of a component in the severity class of interest is given by the summation of the variables of Eq. (3.20), as indicated in

$$K_i = \sum_{j=1}^{n} (k_{\text{fm}})_j = \sum_{j=1}^{n} (F\theta\lambda T)_j \,, \tag{3.21}$$

where n is the item failure modes that fall under the severity classification under consideration.

When a component's failure mode results in multiple severity class effects, each with its own occurrence probability, then only the most important is used in the calculation of the criticality number K_i (Agarwala 1990).

This can lead to erroneously low K_i values for the less critical severity categories. In order to rectify this error, it is recommended to compute F values for all severity categories associated with a failure mode, and ultimately include only contributions of K_i for category B, C and D failures (Bowles et al. 1994).

c) FMECA Data Sources and Users

Design-related information required for the FMECA includes system schematics, functional block diagrams, equipment detail drawings, pipe and instrument diagrams (P&IDs), design descriptions, relevant specifications, reliability data, avail-

able field service data, effects of operational and environmental stress, configuration management data, operating specifications and limits, and interface specifications. Usually, an FMECA satisfies the needs of many groups during the engineering design process, including not only the different engineering disciplines but quality assurance, reliability and maintainability specialists, systems engineering, logistics support, system safety, various regulatory agencies, and manufacturing contractors as well. Some specific FMECA-related factors and their corresponding data retrieval sources are given as follows (Bowles et al. 1994).

FMECA-related factors and their corresponding data sources:

- Failure modes, causes and rates (manufacturer's database, field experience).
- Failure effects (design engineer, reliability engineer, safety engineer).
- Item identification numbers (parts list).
- Failure detection method (design engineer, maintenance engineer).
- Function (client requirements, design engineer).
- Failure probability/severity classification (safety engineer).
- Item nomenclature/functional specifications (parts list, design engineer).
- Mission phase/operational mode (design engineer).

The *FMEA worksheet* (Moss et al. 1996) is tabular in format to provide a systematic approach to the analysis. The column headings of a standard FMEA worksheet generally are:

- *Item identity/description*: a unique identification code and description of each item.
- *Function*: a brief description of the function performed by the item.
- *Failure mode*: each item failure mode is listed separately, as there may be several for an item.
- *Possible causes*: the likely causes of each postulated failure mode.
- *Failure detection method*: features of the design through which failure can be recognised.
- *Failure effect—local level*: the effect of the failure on the item's function.
- *Compensating provisions*: which could mitigate the effect of the failure.
- *Remarks*: comments on the effect of failure, including any potential design changes.

FMEA extension into FMECA worksheet If the analysis is extended to quantify the severity and probability of failure (or failure rate) of the equipment as defined in a failure modes and effects criticality analysis (FMECA), further columns are added to the FMEA worksheet, such as:

Failure consequence—system level: the consequences of the failure mode on system operation.

Severity: the level of severity of the consequence of each failure mode, classified as:

Level 1—minor, with no consequence on functional performance

Level 2—major, with degradation of system functional performance

Level 3—critical, with a severe reduction in the performance of system function resulting in a change in the system operational state
Level 4—catastrophic, with complete loss of system function.

Loss frequency: the expected frequency of loss resulting from each failure mode, either as a failure rate or as failure probability. The latter is usually estimated for the operating time interval as a proportion of the overall system failure rate or failure probability (FP). The levels generally employed for processes are:

i) Very low probability <0.01 FP
ii) Low probability 0.01–0.l FP
iii) Medium probability 0.1–0.2 FP
iv) High probability >0.2 FP

Component failure rate λ_p: the overall failure rate of the component in its operational mode and environment. Where appropriate, application and environmental factors may be applied to adjust for the difference between the conditions associated with the generic failure rate data and operating stresses under which the item is to be used.

Failure mode proportion α: the fraction of the overall failure rate related to the failure mode under consideration.

Probability of failure consequence β: conditional probability that a failure consequence occurs.

Operational failure rate λ_o: the product of λ_p, α and β.

Data source: the source of the failure rate (or failure probability) data.

For FMECAs, a *criticality matrix* is constructed that relates loss frequency to severity for each failure mode. Failure mode identification numbers are entered in the appropriate cell of the matrix according to their loss frequency and severity to identify each critical item failure mode.

Thus: Criticality = Severity $\times$ Loss frequency,
or: Criticality = Severity $\times$ Operational failure rate.

3.2.2.6 Fault-Tree Analysis in Reliability Assessment

There are two approaches that can be used to analyse the causal relationships between *equipment* and *system* failures (Moss et al. 1996). These are *inductive* or forward analysis, and *deductive* or backward analysis. *FMEA* is an example of inductive analysis. As previously considered, it starts with a set of equipment failure conditions and proceeds forwards, identifying the possible consequences; this is a '*what happens if*' approach.

Fault-tree analysis is a deductive '*what can cause this*' approach, and is used to identify the causal relationships leading to a specific *system* failure mode—the 'top event'. The fault tree is developed from this top, undesired event, in branches showing the different event paths. Equipment failure events represented in the tree are progressively redefined in terms of lower resolution events until the basic events

are encountered on which substantial failure data must be available. The events are combined logically by use of gate symbols as shown in Fig. 3.18, which illustrates the structure of a typical fault tree.

In this case, the basic event combinations are developed that could result in total loss of output from a simple cooling water system. Using this failure logic diagram, the probability of the top event or the top event frequency can then be calculated by providing information on the basic event probabilities. The top event and the system boundary must be chosen with care so that the analysis is not too broad or too narrow to produce the results required. The specification of the system boundary is particularly important to the success of the analysis.

Many cooling water systems have external power supplies and other services such as a water supply. It would not be practical to trace all possible causes of failure of these services back through the distribution and generation systems, nor would this extra detail provide any useful information concerning the system being

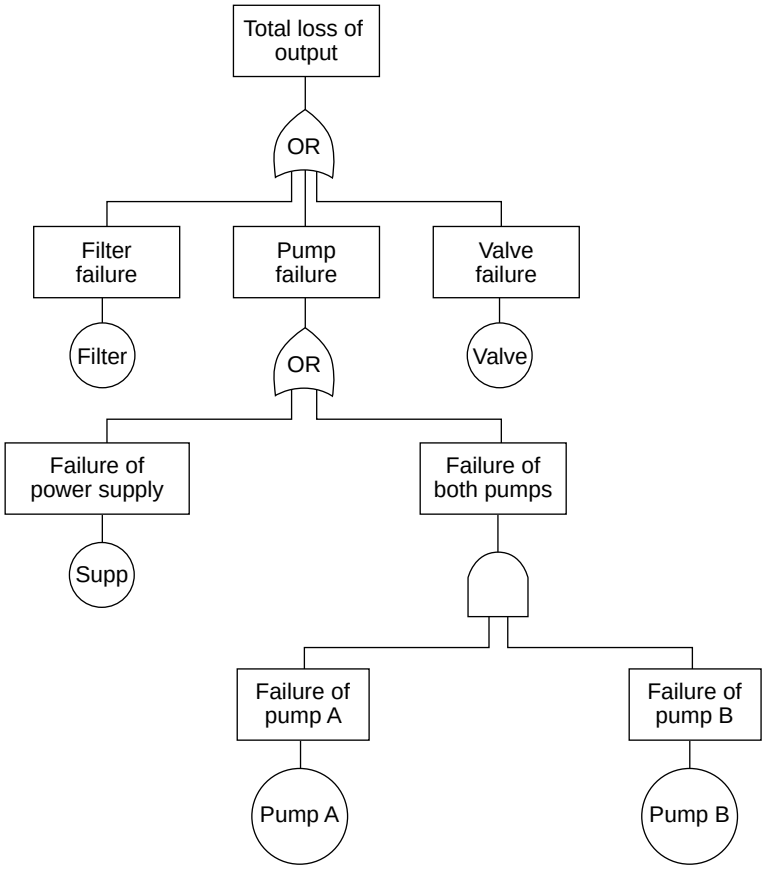


Fig. 3.18 Simple fault tree of cooling water system

assessed. The location of the external boundary will be partially decided by the aspect of system performance that is of interest; however, it is also important to define the external boundary in the time domain. Process start-up or shutdown conditions can generate different hazards from steady-state operation, and it may be necessary to trace any possible faults that could occur.

In Fig. 3.18, basic event combinations are developed of the failures of both pump A *and* pump B *or* failure of the power supply that results in overall pump failure *and/or* failures of the filter *or* valve that could result in total loss of output of the cooling water system. This approach is clearly depicted in the structure of the fault tree of Fig. 3.18, in that the basic events are combined in an *event hierarchy*, from the lower component/sub-assembly levels to the higher assembly/systems levels of the cooling water system systems breakdown structure (SBS).

a) Fault-Tree Analysis Steps

The detailed steps required to perform a fault-tree analysis within the reliability assessment procedure for *equipment* design can be summarised in the following (Andrews et al. 1993):

- Step 1: System configuration understanding.
- Step 2: Identification of system failure states.
- Step 3: Logic model generation.
- Step 4: Qualitative evaluation of the logic model.
- Step 5: Equipment failure analysis.
- Step 6: Quantitative evaluation of the logic model.
- Step 7: Uncertainty analysis.
- Step 8: Sensitivity/importance analysis.

Many of these steps are the same, whatever system and/or equipment is being analysed, though there are some aspects that require special attention, particularly to systems interface when mechanical and electrical equipment is involved. Once the first four steps have been conducted, a qualitative evaluation of the fault-tree logical model is necessary to review whether system configuration and system failure states are correctly understood. The minimal *cut sets* (combinations of *equipment* failures that provide the necessary and sufficient conditions for *system* failure) are then produced.

To progress even further with reliability assessment using fault-tree analysis, the probability of equipment failure, $q(t)$, may be determined *together with equipment maintainability* in the form of a *repair rate*

$$q(t) = \frac{\lambda}{\lambda + \nu}(1 - \mathrm{e}^{-(\lambda+\nu)t}) \,. \tag{3.22}$$

Equation (3.22) is for revealed failures where λ is the failure rate and ν the repair rate. Equation (3.23) is for unrevealed failures, where q_{AV} is the average unavail-

ability, τ is the mean time to repair, and θ is the test interval

$$q_{\mathrm{AV}} = \lambda(\tau + \theta/2)\,. \tag{3.23}$$

For *safety systems* that are normally inactive, failures are revealed only during test or actual use, which means that the unrevealed failure model is appropriate for these systems. However, the underlying assumption in both of these models is that the failure and repair rates are constant, giving a negative exponential distribution for the probability of failure (repair) prior to time t. Constant failure rates are associated with random failure events, as indicated by the useful life period of the hazard rate curve, considered in detail in Section 3.2.3.

However, mechanical equipment subject to wear, corrosion, fatigue, etc. may in many cases not conform to this assumption (Andrews et al. 1993). When either the failure or repair rates are not constant, and the probability density functions for the times to failure $f(t)$ and repair $g(t)$ are available, then they can be combined to give the unconditional failure intensity $w(t)$ and unconditional repair intensity $v(t)$ by solving the following simultaneous integral

$$w(t) = f(t) + \int_0^t f(t-u)v(u)\,\mathrm{d}u\,, \tag{3.24}$$

$$v(t) = \int_0^t g(t-u)w(u)\,\mathrm{d}u\,. \tag{3.25}$$

Having solved these equations, the equipment failure probability is then given by

$$q(t) = \int_0^t [w(u) - v(u)]\,\mathrm{d}u\,. \tag{3.26}$$

For the case of constant failure rates, the probability density functions for the times to failure and repair are given as

$$f(t) = \lambda_{\mathrm{e}}{}^{-\lambda_t}\,, \tag{3.27}$$

$$g(t) = \nu_{\mathrm{e}}{}^{-\nu_t}\,. \tag{3.28}$$

Equations (3.24) and (3.25) can be solved by Laplace transforms. Substituting the solution obtained into Eq. (3.26) yields Eq. (3.27). For more complex distributions of failure and repair times, numerical solutions may be required. With the equipment failure data produced at Step 5, fault-tree quantification gives the system failure probability, the system failure rate, and the expected number of system failures.

Where failure and repair distributions have been specified for the analysis, confidence intervals can be determined at Step 7. Step 8 produces the importance rankings for the basic event identifying the equipment that provides the most significant

contribution to system failure. Fault trees in reliability assessments of integrated engineering systems are significantly more complex than that illustrated in Fig. 3.18.

With complex engineering designs, fault-tree methodology includes the concepts of *availability* and *maintainability*. This is considered in greater detail in Chapter 4, Availability and Maintainability in Engineering Design.

b) Fault-Tree Analysis and Safety and Risk Assessment

The main use of fault trees in *designing for reliability* is in *safety and risk studies*. Fault trees provide a useful representation of the different failure paths, and this can lead to safety and risk assessments of systems and processes even without considering failure and repair data—which does cause some difficulties (Moss et al. 1996).

In many cases, fault trees and failure mode and effect analysis (FMEA) are employed in combination—the FMEA to define the effects and consequences of specific *equipment* failures, and the fault tree (or several fault trees) to identify and quantify the paths that lead to *equipment* failure probability, and high risks of safety.

3.2.3 Theoretical Overview of Reliability Evaluation in Detail Design

Reliability evaluation determines the reliability and criticality values for each individual item of equipment at the *lower* systems levels of the systems breakdown structure. Reliability evaluation determines the failure rates and failure rate *patterns* of components, not only for functional failures that occur at random intervals but for wear-out failures as well.

Reliability evaluation is considered in the *detail design* phase of the engineering design process, to the extent of determination of the frequencies with which failures occur over a specified period of time based on *component* failure rates.

The most applicable methodology for reliability evaluation in the detail design phase includes basic concepts of mathematical modelling such as:

- The hazard rate function.
  (To represent the failure rate *pattern* of a component by evaluating the ratio between its probability of failure and its reliability function.)
- The exponential failure distribution.
  (To define the probability of failure and the reliability function of a component when it is subject only to *functional failures* that occur at *random* intervals.)
- The Weibull failure distribution.
  (To determine component criticality for wear-out failures, rather than random failures.)
- Two-state device reliability networks.
  (A component is said to have two states if it either operates or fails.)

- Three-state device reliability networks.
  (A three-state component derates with one operational and two failure states.)

3.2.3.1 The Hazard Rate Function

The *hazard rate function* is a representation of the failure rate pattern of the ratio between a particular *probability density function (p.d.f.)*, and its *cumulative distribution function (c.d.f.)* or its reliability function.

For continuous random variables, the *cumulative distribution function* is defined by

$$F(t) = \int_{-\infty}^{t} f(x)\,\mathrm{d}x\,, \tag{3.29}$$

where:

$f(x) =$ probability density function of the distribution of value x over the interval $-\infty$ to t.

In the case where $t \to \infty$, the cumulative distribution function is unity

$$F(\infty) = \int_{-\infty}^{\infty} f(x)\,\mathrm{d}x\,. \tag{3.30}$$

The *probability density function* is derived from the derivative of the cumulative distribution function, as follows

$$\frac{\mathrm{d}F(t)}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}\left[\int_{-\infty}^{t} f(x)\,\mathrm{d}x\right]\,. \tag{3.31}$$

The *reliability function* over a period of time t is the difference between the cumulative distribution function where $t \to \infty$ and the cumulative distribution function in the period of time t or, alternately, it is the subtraction of the cumulative distribution function of failure over a period of time t from unity

$$R(t) = 1 - F(t)\,. \tag{3.32}$$

The *hazard rate function* is then defined as

$$\lambda(t) = \frac{f(t)}{R(t)} \tag{3.33}$$

or

$$\lambda(t) = \frac{f(t)}{1 - F(t)}\,.$$

Thus, the hazard rate function can be used to represent the *hazard rate curve* of several different probability density functions, particularly the exponential or Poisson function in which $\lambda(t)$ is a constant, and the Weibull function in which $\lambda(t)$ is either decreasing or increasing.

a) Review of the Hazard Rate Curve

A *hazard rate curve* is shown in Fig. 3.19. This curve is used to represent the failure rate pattern of *equipment* (i.e. assemblies and predominantly components; EPRI 1974). Failure rate representation of electronic *components* is a prime example, in which case only the middle portion (useful life period), or the constant failure rate region of the curve is considered.

As can be seen in Fig. 3.19, the *hazard rate curve* may be divided into three distinct regions or parts (i.e. decreasing, constant, and increasing hazard rate). The *decreasing hazard rate region* of the curve is designated the 'burn-in period', or 'infant mortality period'. The 'burn-in period' failures, known as 'early failures', are the result of design, manufacturing or construction defects in new equipment. As the 'burn-in period' increases, equipment failures decrease, until the beginning of the *constant failure rate region*, which is the middle portion of the curve and designated the 'useful life period' of equipment. Failures occurring during the 'useful life period' are known as 'random failures' because they occur unpredictably. This period starts from the end of the 'burn-in period' and finishes at the beginning of the 'wear-out phase'.

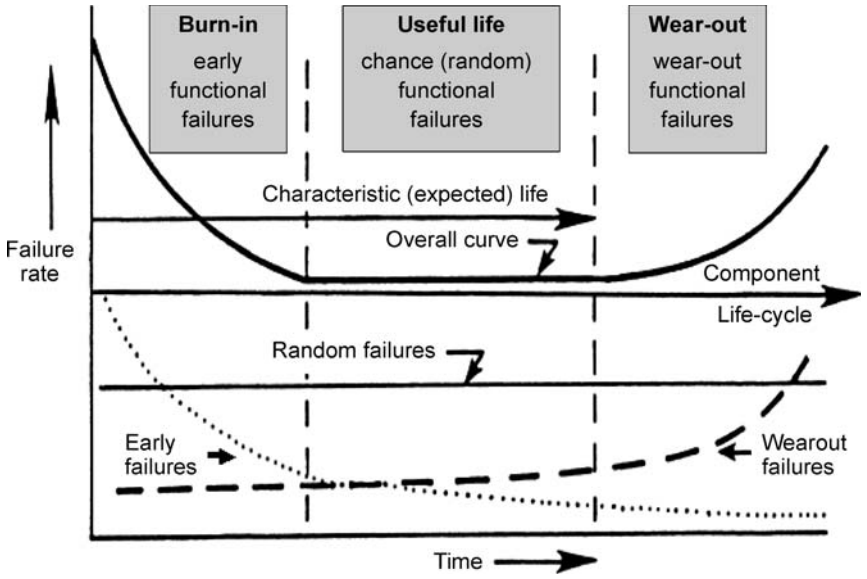


Fig. 3.19 Failure hazard curve (life characteristic curve or risk profile)

The last part of the curve, the *increasing hazard rate region*, is designated the 'wear-out phase' of the equipment. It starts when the equipment has passed its useful life and begins to wear out. During this phase, the number of failures begin to increase exponentially, and are known as 'wear-out failures'.

b) Component Reliability and Failure Distributions

In the calculations for reliability, it is important to note that reliability is an indirect function of the *probability of the occurrence of failure*.

The probability of the occurrence of failure is given by the failure distribution, or *failure probability (FP)* statistic. Thus, the probability of *no* failures occurring over a specific period of time is a measure of the component's or equipment's *reliability* and is given by the *reliability probability (RP)* statistic.

Furthermore, if FP is the *probability of failure* occurring, and RP is the *probability of no failure* occurring, then

$$\mathrm{FP} = 1 - \mathrm{RP}$$

or

$$\mathrm{RP} = 1 - \mathrm{FP}\,. \tag{3.34}$$

Reliability of *components* can thus be determined through the establishment of various *failure distributions*, originating from their *failure density functions.*

Reliability evaluation in designing for reliability assumes that *component reliability* is known, and we are only interested in using this *component reliability* to compute *system reliability*.

However, it is essential to understand how *component reliability* is determined, specifically from two important *failure distributions*, namely:

- *Exponential failure distribution.*
- *Weibull failure distribution.*

3.2.3.2 The Exponential Failure Distribution

When a component is subject only to *functional failures* that occur at *random* intervals, and the expected number of failures is the same for equally long periods of time, its *probability density function* and its *reliability* can be defined by the exponential equation:

Probability density function:

$$f(t,\theta) = \frac{1}{\theta}\,\mathrm{e}^{-t/\theta}\,. \tag{3.35}$$

Reliability:

$$R(t,\theta) = \mathrm{e}^{-t/\theta} \tag{3.36}$$

or, if it is expressed in terms of the *failure rate*, λ

$$f(t,\lambda) = \lambda\, \mathrm{e}^{-\lambda t}\,, \tag{3.37}$$

and the reliability function is

$$R(t,\lambda) = \mathrm{e}^{-\lambda t}\,, \tag{3.38}$$

where:

$f(t,\lambda) =$ probability density function of the Poisson process in terms of time t and failure rate λ.
$R(t,\lambda) =$ reliability of the Poisson process.
t = operating time in the 'useful life period'.
θ = mean time between failures (MTBF).
λ = $1/\theta$, the failure rate for the component.

This equation is applicable for determining *component reliability*, as long as the component is in its 'useful life period'. This is the period during which the *failure rate is constant*, and *failure occurrences* are predominantly *chance or random failures*. The 'useful life period' is considered to be the time after which 'early failures' no longer exist and 'wear-out' failures have not begun.

Note that λ is the distribution *scale* parameter because it scales the exponential function. In reliability terms, λ is the failure rate, which is the reciprocal of the mean time between failure. Because λ is constant for a Poisson process (exponential distribution function), the probability of failure at any time t depends only upon the elapsed time in the component's 'useful life period'.

In complex electro-mechanical systems, the *system failure rate* is effectively constant over the 'useful life period', *regardless of the failure patterns of individual components*. An important point to note about Eqs. (3.37) and (3.38), with respect to *designing for reliability*, is that reliability in this case is a function of *operating time* (t) for the component, as well as the measure of *mean time to failure (MTTF)*.

a) Statistical Properties of the Exponential Failure Distribution

The mean or MTTF The mean, or mean time to fail (MTTF) of the one-parameter exponential distribution is given by the following expression, where $\bar{U}$ is the MTTF

$$\bar{U} = \int_0^\infty t\, f(t)\, \mathrm{d}t\,. \tag{3.39}$$

Relating $f(t)$ to the exponential function gives the relationship

$$\bar{U} = \int_0^\infty t\lambda\, \mathrm{e}^{-\lambda t}\, \mathrm{d}t$$
$$\bar{U} = \frac{1}{\lambda}\,. \tag{3.40}$$

The median The median, $\bar{u}$, of the one-parameter exponential distribution is the value

$$\bar{u} = \frac{1}{\lambda}0.693$$
$$\bar{u} = 0.693\bar{U}\,.$$

The mode The mode, $\mathring{u}$, of the one-parameter exponential distribution is given by

$$\mathring{u} = 0\,. \tag{3.41}$$

For a continuous distribution, the mode is the value of the variate that corresponds to the *maximum* probability density function (p.d.f.). The *modal life*, $\mathring{u}$, is the maximum value of t that satisfies the expression

$$\frac{\mathrm{d}[f(t)]}{\mathrm{d}t} = 0\,.$$

The standard deviation The standard deviation σ_T of the one-parameter exponential distribution is given by

$$\sigma_T = \frac{1}{\lambda} = m\,. \tag{3.42}$$

The reliability function The one-parameter exponential reliability function is given by

$$R(T) = \mathrm{e}^{-\lambda T}$$
$$R(T) = \mathrm{e}^{-T/m}\,.$$

This is the complement of the exponential cumulative distribution function where

$$R(T) = 1 - \int_0^T f(T)\,\mathrm{d}T$$
$$R(T) = 1 - \int_0^T \lambda\, \mathrm{e}^{-\lambda T}\,\mathrm{d}T$$
$$R(T) = \mathrm{e}^{-\lambda T}\,. \tag{3.43}$$

Conditional reliability Conditional reliability calculates the probability of further successful functional duration, given that an item has already successfully functioned for a certain time. In this respect, conditional reliability could be considered to be the reliability of 'used items or components'. This implies that the reliability for an added duration (mission) of t undertaken *after* the equipment or component has already accumulated T hours of operation from age zero is a function only of the added time duration, and not a function of the *age* at the beginning of the mission.

The conditional reliability function for the one-parameter exponential distribution is given by the following expression

$$\begin{aligned} R(T,t) &= \frac{R(T+t)}{R(T)} \\ R(T,t) &= \frac{\mathrm{e}^{-\lambda(T+t)}}{\mathrm{e}^{-\lambda T}} \\ R(T,t) &= \mathrm{e}^{-\lambda t} . \end{aligned} \tag{3.44}$$

Reliable life The reliable life, or the mission duration for a desired reliability goal for the one-parameter exponential distribution is given by

$$\begin{aligned} R(t_R) &= \mathrm{e}^{-\lambda t_R} \\ \ln\{R(t_R)\} &= -\lambda t_R \\ t_R &= \frac{-\ln\{R(t_R)\}}{\lambda} . \end{aligned} \tag{3.45}$$

Residual life Let T denote the time to failure for an item. The *conditional survival function* can then be expressed as

$$R(t) = P(T > t) .$$

The *conditional survival function* is the probability that the item will survive for period t given that it has survived without failure for period T. The *residual life* is thus the extended duration or operational life t where the component has already accumulated T hours of operation from age zero, subject to the conditional survival function.

The *conditional survival function* of an item that has survived (without failure) up to time x is

$$\begin{aligned} R(t|x) &= P(t > t+x|T > x) \\ &= \frac{P(T > t+x)}{P(T > t)} \\ &= \frac{R(t+x)}{R(x)} . \end{aligned} \tag{3.46}$$

$R(t|x)$ denotes the probability that a used item of age x will survive an extra time t.

The *mean residual life (MRL)* of a used item of age x can thus be expressed as

$$\mathrm{MRL}(x) = \int_0^\infty R(t|x)\,\mathrm{d}t\ . \tag{3.47}$$

When $x = 0$, the initial age is zero, implying a new item and, consequently

$$\mathrm{MRL}(0) = \mathrm{MTTF}\ .$$

In considering the reliable life for the one-parameter exponential distribution compared to the residual life, it is of interest to study the function

$$h(x) = \frac{\mathrm{MRL}(x)}{\mathrm{MTTF}}\ . \tag{3.48}$$

There are certain characteristics of comparison, when the initial age is zero (i.e. $x = 0$), between the *mean residual life MRL* (x) and the mean or the *mean time to fail (MTTF)*.

Characteristics of comparison between the *mean residual life MRL* (x) and the mean, or *mean time to fail (MTTF)*, are the following:

- When the time to failure for an item, T, has an exponential distribution, then $h(x) = 1$ for all x.
- When T has a Weibull distribution with shape parameter $\beta < 1$ (i.e. decreasing failure rate), then $h(x)$ is an *increasing* function.
- When T has a Weibull distribution with shape parameter $\beta > 1$ (i.e. increasing failure rate), then $h(x)$ is a *decreasing* function.

Failure rate function The exponential failure rate function is given by

$$\lambda t = \frac{f(T)}{R(T)} = \frac{\lambda\,\mathrm{e}^{-\lambda T}}{\mathrm{e}^{-\lambda T}} = \lambda \tag{3.49}$$

$$\frac{f(T)}{R(T)} = \text{ hazard rate } h(t), \text{ and } \lambda(t) \text{ is constant } \lambda.$$

The *hazard rate* is a constant with respect to time for the exponential failure distribution function. For other distributions, such as the Weibull distribution or the log-normal distribution, the hazard rate is not constant with respect to time.

3.2.3.3 The Weibull Failure Distribution

Although the determination of *equipment reliability* and corresponding *system reliability* during the period of the equipment's useful life period is based on the *exponential failure distribution*, the *failure rate* of the equipment may *not* be constant throughout the period of its use or operation. In most engineering installations, particularly with the integration of complex systems, the purpose of determining

equipment criticality, or combinations of critical equipment, is predominantly to *assess the times to wear-out failures, rather than to assess the times to chance or random failures*.

In such cases, the *exponential failure distribution* does *not* apply, and it becomes necessary to substitute a general failure distribution, such as the *Weibull distribution*. The Weibull distribution is particularly useful because it can be applied to *all three of the phases* of the *hazard rate curve*, which is also called the equipment 'life characteristic curve'.

The equation for the two-parameter *Weibull cumulative distribution function (c.d.f.)* is given by

$$F(t) = \int_0^1 f(t|\beta\mu)\,\mathrm{d}t \ . \tag{3.50}$$

The equation for the two-parameter *Weibull probability density function (p.d.f.)* is given by

$$f(t) = \frac{\beta t^{(\beta-1)}\,\mathrm{e}^{-t/\mu^{\beta}}}{\mu^{\beta}} \ , \tag{3.51}$$

where:

t = the *operating time* for which the *reliability* $R(t)$ of the component must be determined.
β = parameter of the Weibull distribution referred to as the *shape parameter*.
μ = parameter of the Weibull distribution referred to as the *scale parameter*.

a) Statistical Properties of the Weibull Distribution

The mean or MTTF The mean, $\bar{U}$, of the two-parameter Weibull probability density function (p.d.f.) is given by

$$\bar{U} = \mu\Gamma(1/\beta + 1) \ , \tag{3.52}$$

where $\Gamma(1/\beta+1)$ is the gamma function, evaluated at $(1/\beta+1)$.

The median The median, $\bar{u}$, of the two-parameter Weibull distribution is given by

$$\bar{u} = \mu(\ln 2)^{1/\beta} \ . \tag{3.53}$$

The mode The mode or value with maximum probability, $\mathring{u}$, of the two-parameter Weibull distribution is given by

$$\mathring{u} = \mu\left(1 - \frac{1}{\beta}\right)^{1/\beta} \ . \tag{3.54}$$

The standard deviation The standard deviation, σ_T, of the two-parameter Weibull is given by

$$\sigma_T = \mu \sqrt{\Gamma\left(\frac{2}{\beta}+1\right) - \Gamma\left(\frac{1}{\beta}+\right)^2}\,. \tag{3.55}$$

The cumulative distribution function (c.d.f.) The c.d.f. of the two-parameter Weibull distribution is given by

$$F(T) = 1 - \mathrm{e}^{-(T/\mu)^{\beta}}\,. \tag{3.56}$$

Reliability function The Weibull reliability function is given by

$$R(T) = 1 - F(t) = \mathrm{e}^{-(T/\mu)^{\beta}}\,. \tag{3.57}$$

The conditional reliability function Equation (3.58) gives the reliability for an extended operational period, or *mission duration* of t, having already accumulated T hours of operation up to the start of this mission duration, and estimates whether the component will begin the next mission successfully.

It is termed *conditional* because the reliability of the following operational period or new mission can be estimated, based on the fact that the component has already successfully accumulated T hours of operation.

The Weibull conditional reliability function is given by

$$\begin{aligned} R(T,t) &= \frac{R(T+t)}{R(T)} \\ &= \frac{\mathrm{e}^{-(T+t/\mu)^{\beta}}}{\mathrm{e}^{-(T/\mu)^{\beta}}} \\ &= \mathrm{e}^{-\left[(T+t/\mu)^{\beta}-(T/\mu)^{\beta}\right]}\,, \end{aligned} \tag{3.58}$$

The reliable life For the two-parameter Weibull distribution, the reliable life, T_{R}, of a component for a specified reliability, starting at age zero, is given by

$$T_{\mathrm{R}} = \mu \left\{-\ln\left[R(T_{\mathrm{R}}\right]\right\}^{1/\beta} \tag{3.59}$$

b) The Weibull Shape Parameter

The range of shapes that the *Weibull density function* can take is very broad, depending on the value of the *shape parameter* β. This value is usually indicated as $\beta < 1$, $\beta = 1$ and $\beta > 1$. Figure 3.20 illustrates the shape of the Weibull c.d.f. $F(t)$ for different values of β. The amount the curve is spread out along the abscissa or *x*-axis depends on the parameter μ, thus being called the *Weibull scale parameter*.

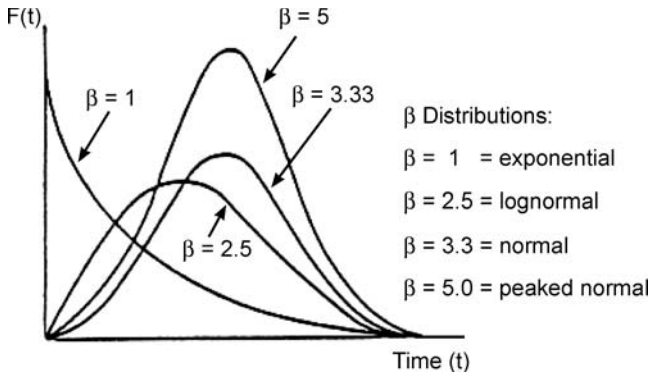


Fig. 3.20 Shape of the Weibull density function, $F(t)$, for different values of β

For $\beta < 1$, the Weibull curve is asymptotic to both the *x*-axis and the *y*-axis, and is skewed.
For $\beta = 1$, the Weibull curve is identical to the exponential density function.
For $\beta > 1$, the Weibull curve is 'bell shaped' but skewed.

c) The Weibull Distribution Function, Reliability and Hazard

Integrating out the *Weibull cumulative distribution function (c.d.f.)* given in Eq. (3.50) gives the following

$$F(t) = \int_0^1 f(t|\beta\mu)\,\mathrm{d}t$$

$$F(t) = 1 - \mathrm{e}^{-t/\mu^{\beta}} \,. \tag{3.60}$$

The mathematical model of *reliability* for the Weibull density function is

$$R(t) = 1 - F(t)$$

$$R = \mathrm{e}^{-t/\mu^{\beta}} \,, \tag{3.61}$$

where:

R is the 'probability of success' or reliability.
t is the equipment age.
μ is the characteristic life or scale parameter.
β is the slope or shape parameter.

The Weibull *hazard rate function*, $\lambda(t)$, is derived from a ratio between the Weibull *probability density function (p.d.f.)* and the Weibull *reliability function*

$$\lambda(t) = \frac{f(t)}{R(t)}$$
$$\lambda(t) = \frac{\beta(t)^{\beta-1}}{\mu^{\beta}}, \tag{3.62}$$

where:

$\mu =$ the scale parameter,
$\beta =$ the shape parameter.

To use this model, one must estimate the values of μ and β. Estimates of these parameters from the Weibull probability density function are computationally difficult to obtain. There are analytical methods for estimating these parameters but they involve the solution of a system of transcendental equations. An easier and commonly used method is based on a graphical technique that makes use of the *Weibull graph chart*.

d) The Weibull Graph Chart

The values of the *failure distribution*, expressed as percentage values of failure occurrences, are plotted against the *y*-axis of the chart displayed in Fig. 3.21, and the corresponding *time between failures* plotted against the *x*-axis. If the plot is a straight

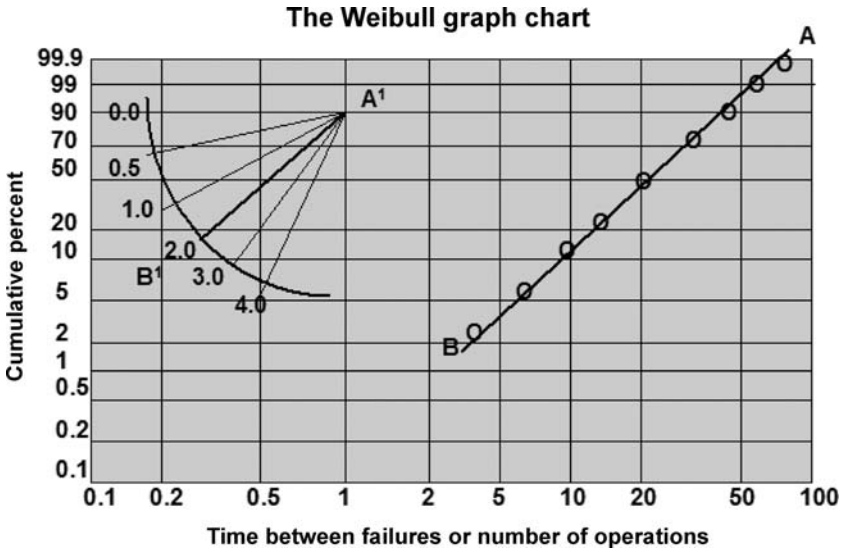


Fig. 3.21 The Weibull graph chart for different percentage values of the failure distribution

line, then the Weibull distribution is applicable and the relevant parameters are determined. If the plot is *not* a straight line, then the *two-parameter* Weibull distribution is *not* applicable and more detailed analysis is required. Such detailed analysis is presented in Section 3.3.3. To explain the format of the chart in Fig. 3.21, each axis of the chart is considered.

- The scale of the *x*-axis is given as a log scale.
- The description given along the *y*-axis is:
- '*cumulative percent*' for '*cumulative distribution function* (%)'
- The scale of the *y*-axis is given as a log–log scale.

3.2.3.4 Reliability Evaluation of Two-State Device Networks

The following models present reliability evaluation of series and parallel two-state device networks (Dhillon 1983):

a) Series Network

This network denotes an assembly of which the components are connected in series. If any one of the components malfunctions, it will cause the assembly to fail. For the *k* non-identical and independent component series, which are time *t*-dependent, the formula for $R_S(t)$, the network reliability, is given in

$$\begin{aligned} R_S(t) &= \{1-F_1(t)\}\cdot\{1-F_2(t)\}\cdot\{1-F_3(t)\}\cdot\ldots\cdot\{1-F_k(t)\} \\ \text{And:}\quad &\{1-F_i(t)\}\approx R_i(t)\,. \end{aligned} \tag{3.63}$$

The *i*th component cumulative distribution function (failure probability) is defined by

$$F_i(t) = \int_0^t f_i(t)\,\mathrm{d}t\,, \tag{3.64}$$

where:

$F_i(t)$ is the *i*th component failure probability for $i = 1,2,3,\ldots,k$.
$R_i(t)$ is the *i*th component reliability, for $i = 1,2,3,\ldots,k$.

By definition:

$$f_i(t) = \lim_{\Delta t\to 0}\frac{\alpha_S(t)-\alpha_S(t+\Delta t)}{\alpha_0\Delta t}$$

$$f_i(t) = \frac{\mathrm{d}F_i(t)}{\mathrm{d}t}\,,$$

where:

Δt = the time interval,

$\alpha_0 =$ the total number of items put on test at time $t = 0$,
$\alpha_S =$ the number of items surviving at time t or at $t + \Delta t$.

Substituting Eq. (3.64) into Eq. (3.63) leads to

$$R_i(t) = 1 - \int_0^t f_i(t)\,\mathrm{d}t \ . \tag{3.65}$$

A more common notation for the *i*th component reliability is expressed in terms of the mathematical constant e. The mathematical constant e is the unique real number, such that the value of the derivative of $f(x) = \mathrm{e}x$ at the point $x = 0$ is exactly 1. The function so defined is called the exponential function. Thus, the alternative, commonly used expression for $R_i(t)$ is

$$R_i(t) = \mathrm{e}^{-\int_0^t \lambda_i(t)\,\mathrm{d}t} \ , \tag{3.66}$$

where $\lambda_i(t)$ is the *i*th component hazard rate or instantaneous failure rate.

In this case, component failure time can follow any statistical distribution function of which the hazard rate is known. The expression $R_i(t)$ is reduced to

$$\begin{aligned} R_i(t) &= 1 - F_i(t) \\ R_i(t) &= \mathrm{e}^{-\lambda_i t} \ . \end{aligned} \tag{3.67}$$

A redundant configuration or single component MTBF is defined by

$$\mathrm{MTBF} = \int_0^\infty R(t)\,\mathrm{d}t \ . \tag{3.68}$$

Thus, substituting Eq. (3.67) into Eq. (3.66), and integrating the results in the series gives the model for MTBF, which in effect is the sum of the inverse values of the component hazard rates, or instantaneous failure rates of all the components in the series

$$\mathrm{MTBF} = \left[\sum_{i=1}^{n} \lambda_i\right]^{-1} \tag{3.69}$$

$$\begin{aligned} \mathrm{MTBF} &= \text{sum of inverse values of component hazard rates} \\ &= \text{instantaneous failure rates of all the components.} \end{aligned}$$

b) Parallel Network

This type of redundancy can be used to improve system and equipment reliability. The redundant system or equipment will fail only if all of its components fail. To develop this mathematical model for application in *reliability evaluation*, it is

assumed that all units of the system are active and load sharing, and units are statistically independent. The unreliability, $F_{\mathrm{P}}(t)$, at time t of a parallel structure with non-identical components is

$$F_{\mathrm{P}}(t) = \prod_{i=1}^{k} F_i(t) \tag{3.70}$$

$F_i(t) =$ ith component unreliability (failure probability).

Since $R_{\mathrm{P}}(t) + F_{\mathrm{P}}(t) = 1$, utilising Eq. (3.70) the parallel structure reliability, $R_{\mathrm{P}}(t)$, becomes

$$R_{\mathrm{P}}(t) = 1 - \prod_{i=1}^{k} F_i(t)\ . \tag{3.71}$$

Similarly, as was done for the series network components with constant failure rates, substituting for $F_i(t)$ in Eq. (3.71) we get

$$R_{\mathrm{P}}(t) = 1 - \prod_{i=1}^{k} \left(1 - \mathrm{e}^{-\lambda_i t}\right)\ . \tag{3.72}$$

In order to obtain the series MTBF, substitute Eq. (3.69) for identical components and integrate as follows

$$\mathrm{MTBF} = \int_0^{\infty} \left[1 - \sum_{j=0}^{k} (n_j)(-1)^j \mathrm{e}^{-\lambda_j t}\right] \mathrm{d}t$$

$$\mathrm{MTBF} = \frac{1}{\lambda} + \frac{1}{2\lambda} + \frac{1}{3\lambda} + \ldots + \frac{1}{k\lambda} \tag{3.73}$$

$\lambda =$ the component hazard or instantaneous failure rate.

c) A k-out-of-m Unit Network

This type of redundancy is used when a certain number k of components in an active parallel redundant system or assembly must work for the system's or assembly's success. The *binomial distribution*, system or assembly *reliability* of the independent and identical components at time t is $R_{k/m}(t)$, where $R(t)$ is the *component reliability*

$$R_{k/m}(t) = \sum_{i=k_t}^{m} (m)[R(t)]^i [1 - R(t)]^{k-i} \tag{3.74}$$

$m =$ the total number of system/assembly components

$k =$ the number of components required for system/assembly success at time t.

Special cases of the *k*-out-of-*m* unit system are:

$k = 1$: = parallel network
$k = m$: = series network.

For exponentially distributed failure times (constant failure rate) of a component, substituting in Eq. (3.74) for $k = 2$ and $m = 4$, the equation becomes

$$R_{2/4}(t) = 3\,\mathrm{e}^{-4\lambda t} - 8\,\mathrm{e}^{-3\lambda t} + 6\,\mathrm{e}^{-2\lambda t}\,. \tag{3.75}$$

d) Standby Redundant Systems

$$R_{\mathrm{S}}(t) = \sum_{i=0}^{K} \left[\int_0^t \lambda(t)\,\mathrm{d}t \right]^i \mathrm{e}^{-\int_0^t \lambda(t)\,\mathrm{d}t} (i!)^{-1}\,. \tag{3.76}$$

In this case (Eq. 3.76), one component is functioning, and K components are on standby, or are not active. To develop a system/assembly reliability model, the components must be identical and independent, and the standby components as new. The general components hazard rate, λ, is assumed.

3.2.3.5 Reliability Evaluation of Three-State Device Networks

A three-state device (component) has one operational and two failure states. Devices such as a fluid flow valve and an electronic diode are examples of a three-state device. These devices have failure modes that can be described as failure in the closed or open states. Such a device can have the following functional states (Dhillon 1983):

State 1 = Operational
State 2 = Failed in the closed state
State 3 = Failed in the open state

a) Parallel Networks

A parallel network composed of active independent three-state components will fail only if all the components fail in the open mode, or at least one of the devices must fail in the closed mode. The network (with non-identical devices) time-dependent reliability, $R_{\mathrm{P}}(t)$, is

$$R_{\mathrm{P}}(t) = \prod_{i=1}^{k} [1 - F_{\mathrm{C}_i}(t)] - \prod_{i=1}^{k} F_{\mathrm{O}_i}(t)\,, \tag{3.77}$$

where:

t = time
k = the number of three-state devices in parallel
$F_{C_i}(t)$ = the *closed mode* probability of device *i* at time *t*
$F_{O_i}(t)$ = the *open mode* probability of device *i* at time *t*

b) Series Networks

A series network is the reverse of the parallel network. A series system will fail only if all of its independent elements fail in a *closed mode* or any one of the components fails in *open mode*. Thus, because of duality, the time-dependent reliability of the series network with non-identical and independent devices is the difference of the summations of the respective values for the *open mode probability*, $[1-F_{O_i}(t)]$, and the *closed mode probability*, $[F_{C_i}(t)]$, of device *i* at time *t*.

The series network with non-identical and independent devices time-dependent reliability, $R_S(t)$, is

$$R_S(t) = \prod_{i=1}^{k}[1-F_{O_i}(t)] - \prod_{i=1}^{k}F_{C_i}(t)\ , \tag{3.78}$$

where:

t = time
k = the number of devices in the series configuration
$F_{C_i}(t)$ = the closed mode probability of device *i* at time *t*
$F_{O_i}(t)$ = the open mode probability of device *i* at time *t*

Closing comments to theoretical overview

It was stated earlier, and must be iterated here, that these techniques do *not* represent the total spectrum of reliability calculations, and have been considered as the most applicable for their application in determining the *integrity of engineering design* during the *conceptual, preliminary* and *detail design* phases of the engineering design process, based on an extensive study of the available literature. Furthermore, the techniques have been grouped according to significant differences in the approaches to the determination of reliability of *systems*, compared to that of *assemblies* or of *components*. This supports the premise that:

- *predictions of the reliability of systems* are based on *prognosis of systems performance* under conditions subject to failure modes (*reliability prediction*);
- *assessments of the reliability of equipment* are based upon *inferences of failure* according to various statistical failure distributions (*reliability assessment*); and
- *evaluations of the reliability of components* are based upon *known values of failure rates* (*reliability evaluation*).

3.3 Analytic Development of Reliability and Performance in Engineering Design

Some of the techniques identified for reliability prediction, assessment and evaluation, in the *conceptual*, *preliminary* and *detail* design phases respectively, have been considered for further analytic development. This has been done on the basis of their transformational capabilities in developing intelligent computer automated methodology. The techniques should be suitable for application in artificial intelligence-based modelling, i.e. *AIB modelling* in which *knowledge-based expert systems* within a *blackboard model* can be applied in determining the integrity of engineering design. The *AIB model* should be suited to applied *concurrent engineering design* in an online and integrated *collaborative engineering design* environment in which automated *continual design reviews* are conducted throughout the engineering design process by remotely located design groups communicating via the internet.

Engineering designs are usually composed of highly integrated, tightly coupled systems with complex interactions, essential to the functional performance of the design. Therefore, *concurrent*, rather than *sequential* considerations of specific requirements are essential, such as meeting the design criteria together with design integrity constraints. The traditional approach in industry for designing engineered installations has been the implementation of a sequential consideration of requirements for process, thermal, power, manufacturing, installation and/or structural constraints. In recent years, *concurrent engineering design* has become a widely accepted concept, particularly as a preferred alternative to the sequential engineering design process. Concurrent engineering design in the context of design integrity is a systematic approach to integrating the various *continual design reviews* within the engineering design process, such as reliability prediction, assessment, and evaluation throughout the preliminary, schematic, and detail design phases respectively. The objective of concurrent engineering design with respect to design integrity is to assure a reliable design throughout the engineering design process. Parallelism is the prime concept in concurrent engineering design, and design integrity (i.e. designing for reliability) becomes the central issue. Integrated *collaborative engineering design* implies information sharing and decision coordination for conducting the *continual design reviews*.

3.3.1 Analytic Development of Reliability and Performance Prediction in Conceptual Design

Techniques for *reliability and performance prediction* in determining the integrity of engineering design during the *conceptual design* phase include system reliability modelling based on:

i. System performance measures
ii. Determination of the most reliable design

iii. Conceptual design optimisation and
iv. Comparison of conceptual designs
v. Labelled interval calculus and
vi. Labelled interval calculus in designing for reliability

3.3.1.1 System Performance Measures

For each process system, there is a set of performance measures that require particular attention in design—for example, temperature range, pressure rating, output and flow rate. Some measures such as pressure and temperature rating may be common for different items of equipment inherent to each process system. Some measures may apply only to one system. The performance measures of each system can be described in matrix form in a *parameter profile matrix* (Thompson et al. 1998), as shown in Fig. 3.22 where:

i = number of performance measure parameters
j = number of process systems
x = a data point that measures the performance of a system with respect to a particular parameter.

It is not meaningful to use actual performance—for example, an operating temperature—as the value of x_{ij}. Rather, it is the *proximity of the actual performance to the limit of process capability of the system* that is useful.

In engineering design review, the proximity of performance to a limit closely relates to a measure of the *safety margin*. In the case of process enhancement, the proximity to a limit may even indicate an inhibitor to proposed changes. For a process system, a non-dimensional numerical value of x_{ij} may be obtained by determining the *limits of capability*, such as $C_{\max}$ and $C_{\min}$, with respect to each performance parameter, and specifying the nominal point or range at which the system's performance parameter is required to operate.

The limits may be represented diagrammatically as shown in Figs. 3.23, 3.24 and 3.25, where an example of two performance limits, of one upper performance limit, and of one lower performance limit is given respectively (Thompson et al. 1998).

The data point x_{ij} that is entered into the performance of systems with two performance limits is the lower value of A and B ($0 <$ score < 10), which is the closest

	Process systems					
Performance parameters	$\mathbf{x}_{11}$	$\mathbf{x}_{12}$	$\mathbf{x}_{13}$	$\mathbf{x}_{14}$	...	$\mathbf{x}_{1i}$
	$\mathbf{x}_{21}$	$\mathbf{x}_{22}$	$\mathbf{x}_{23}$	$\mathbf{x}_{24}$	...	$\mathbf{x}_{2i}$
	$\mathbf{x}_{31}$	$\mathbf{x}_{32}$	$\mathbf{x}_{33}$	$\mathbf{x}_{34}$	...	$\mathbf{x}_{3i}$
	$\mathbf{x}_{j1}$	$\mathbf{x}_{j2}$	$\mathbf{x}_{j3}$	$\mathbf{x}_{j4}$	...	$\mathbf{x}_{ji}$

Fig. 3.22 Parameter profile matrix

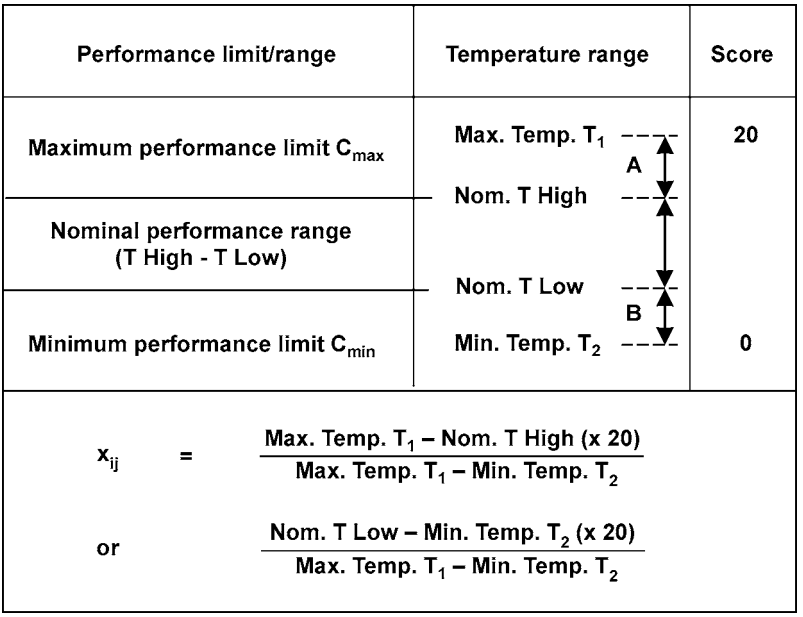


Fig. 3.23 Determination of a data point: two limits

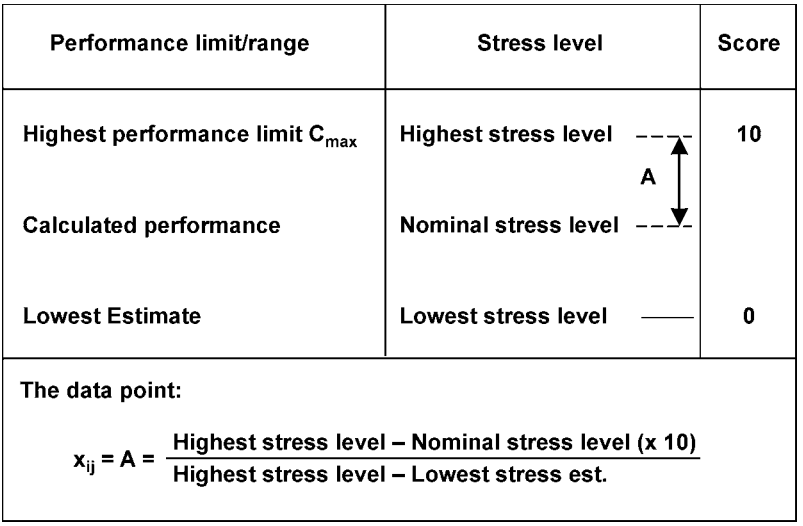


Fig. 3.24 Determination of a data point: one upper limit

the nominal design condition does approach a limit. The value of x_{ij} always lies in the range 0–10. Ideally, when design condition is a single point at the mid-range, then the data point is 10.

Performance limit/range	Capacity level	Score
Highest estimate	Max. capacity est.	10
Calculated performance	Nominal capacity	
	B	
Lowest performance limit C_{min}	Min. capacity level	0

The data point:

$$x_{ij} = B = \frac{\text{Nominal capacity} - \text{Min. capacity level (x 10)}}{\text{Max. capacity est.} - \text{Min. capacity level}}$$

Fig. 3.25 Determination of a data point: one lower limit

It is obvious that this process of data point determination can be generated quickly by computer modelling with inputs from process system performance measures and ranges of capability. If there is one operating limit only, then the data point is obtained as shown in Figs. 3.24 and 3.25, where the upper or lower limits respectively are known.

Therefore, a set of data points can be obtained for each system with respect to the performance parameters that are relevant to that system. Furthermore, a method can be adopted to allow *designing for reliability* to be quantified, which can lead to optimisation of design reliability.

Figures 3.23, 3.24 and 3.25 illustrate how a data point can be generated to measure performance with respect to the best and the worst limits of performance.

3.3.1.2 Determination of the Most Reliable Design in the Conceptual Design Phase

Reliability prediction through system reliability modelling based on system performance may be carried out by the following method (Thompson et al. 1999):

a) Identify the criteria against which the process design is measured.
b) Determine the maximum and minimum acceptable limits of performance for each criterion.
c) Calculate a set of measurement data points of x_{ij} for each criterion according to the algorithms indicated in Figs. 3.23, 3.24 and 3.25.

d) A design proposal that has good reliability will exhibit uniformly high scores of the data points x_{ij}. Any low data point represents system performance that is close to an unacceptable limit, indicating a low safety margin.
e) The conceptual design may then be reviewed and revised in an iterative manner to improve low x_{ij} scores.

When a uniformly high set of scores has been obtained, then the design, or alternative design that is most reliable, will conform to the *equal strength principle*, also referred to as *unity*, in which there are no 'weak links' (Pahl et al. 1996).

3.3.1.3 Comparison of Conceptual Designs

If it is required to compare two or more conceptual designs, then an overall rating of reliability may be obtained to compare these designs. An overall reliability may be determined by calculating a *systems performance index (SP)* as follows

$$\mathrm{SP} = N \left(\sum_{i=1}^{N} 1 / d_i \right)^{-1} \tag{3.79}$$

where

$N =$ the sum of the performances considered
$d_i =$ the scores of the performances considered.

The overall SP score lies in the range from 0 to 10. The inverse method of combination of scores readily identifies low safety margins, unlike normal averaging through addition where almost no safety margin with respect to one criterion may be compensated for by high safety margins elsewhere—which is unacceptable. Alternative designs can therefore be compared with respect to reliability, by comparing their SP scores; the highest score is the most reliable. In a proposed method for using this overall rating approach (Liu et al. 1996), caution is required because simply choosing the highest score may *not* be the best solution. This requires that each design should always be reviewed to see whether weaknesses can be improved upon, which tends to defeat the purpose of the method. Although other factors such as costs may be the final selection criterion for conceptual or preliminary design proposals with similar overall scores (which oft is the case), the objective is to achieve a design solution that is the most reliable from the viewpoint of meeting the required performance criteria. This shortcoming in the overall rating approach may be avoided by supplementing performance measures obtained from mathematical models in the form of mathematical algorithms of process design integrity for the values of x_{ij}, rather than the 'direct' performance parameters such as temperature range, pressure rating, output or flow rate.

The performance measures obtained from these mathematical models consider the *prediction*, *assessment* or *evaluation* of parameters particular to each specific stage of the design process, whether it is conceptual design, preliminary design or detail design respectively.

The approach defines performance measures that, when met, achieve an optimum design with regard to overall *integrity*. It seeks to maximise the integrity of design by ensuring that the criteria of *reliability*, *availability*, *maintainability* and *safety* are concurrently being met. The choice of limits of performance for such an approach is generally made with respect to the *consequences* and *effects* of failure, and *reliability expectations* based on the propagation of single maximum and minimum values of acceptable performance for each criterion. If the consequences and/or effects of failure are high, then limits of acceptable performance with high safety margins that are well clear of failure criteria are chosen. Similarly, if failure criteria are imprecise, then high safety margins are adopted.

These considerations have been further expanded to represent sets of systems that function under sets of failures and performance intervals, applying *labelled interval calculus* (Boettner et al. 1992).

The most significant advantage of this expanded method is that, besides not having to rely on the propagation of single estimated values of failure data, *it also does not have to rely on the determination of single values of maximum and minimum acceptable limits of performance for each criterion*. Instead, *constraint propagation* of intervals about sets of performance values is applied. As these intervals are defined, it is possible to compute a multi-objective optimisation of performance values, in order to determine optimal solution sets for different sets of performance intervals.

3.3.1.4 Conceptual Design Optimisation

The process described attempts to improve reliability continually towards an optimal result (Thompson et al. 1999). If the design problem can be modelled so that it is possible to compute all the x_{ij} scores, then it is possible to optimise mathematically in order to maximise the SP function, as a result of which the x_{ij} scores will achieve a uniformly high score. Typically in engineering design, several conceptual design alternatives need to be optimised for different design criteria or constraints.

To deal with multiple design alternatives, the *parameter profile matrix*, in which the scores for each system's performance measure of x_{ij} is calculated, needs to be modified. Instead of a one-variable matrix, in which the scores x_{ij} are listed, the analysis is completed for each specific criterion y_j. Thus, a two-variable matrix of c_{ij} is constructed, as shown in Fig. 3.26 (Liu et al. 1996).

Design alternatives		$\mathbf{y}_1$	$\mathbf{y}_2$	$\mathbf{y}_3$	$\mathbf{y}_4$	$\mathbf{y}_n$
Performance parameters	$\mathbf{x}_1$	$\mathbf{c}_{11}$	$\mathbf{c}_{12}$	$\mathbf{c}_{13}$	$\mathbf{c}_{14}$	$\mathbf{c}_{1n}$
	$\mathbf{x}_2$	$\mathbf{c}_{21}$	$\mathbf{c}_{22}$	$\mathbf{c}_{23}$	$\mathbf{c}_{24}$	$\mathbf{c}_{2n}$
	$\mathbf{x}_3$	$\mathbf{c}_{31}$	$\mathbf{c}_{32}$	$\mathbf{c}_{33}$	$\mathbf{c}_{34}$	$\mathbf{c}_{3n}$
	$\mathbf{x}_m$	$\mathbf{c}_{m1}$	$\mathbf{c}_{m2}$	$\mathbf{c}_{m3}$	$\mathbf{c}_{m4}$	$\mathbf{c}_{mn}$

Fig. 3.26 Two-variable parameter profile matrix

Determination of an optimum conceptual design is carried out as follows:

a) A performance *parameter profile index (PPI)* is calculated for each performance parameter x_i. This constitutes an analysis of the *rows* of the matrix, in which

$$\text{PPI} = n\left(\sum_{j=1}^{n} 1/c_{ij}\right)^{-1} \tag{3.80}$$

where n is the number of design alternatives.

b) Similarly, a design *alternative performance index (API)* is calculated for each design alternative y_j. This constitutes an analysis of the *columns* of the matrix, in which

$$\text{API} = m\left(\sum_{i=1}^{m} 1/c_{ij}\right)^{-1} \tag{3.81}$$

where m is the number of performance parameters.

c) An *overall performance index (OPI)* is then calculated as

$$\text{OPI} = \frac{100}{mn}\left[\sum_{i=1}^{m}\sum_{j=1}^{n}(\text{PPI})(\text{API})\right] \tag{3.82}$$

where m is the number of performance parameters, n is the number of design alternatives, and OPI lies in the range 0–100 and can thus be indicated as a percentage value.

d) Optimisation is then carried out iteratively to maximise the *overall performance index*.

3.3.1.5 Labelled Interval Calculus

Interval calculus is a method for *constraint propagation* whereby, instead of designating single values, information about sets of values is propagated. Constraint propagation of intervals is comprehensively dealt with by Moore (1979) and Davis (1987). However, this standard notion of interval constraint propagation is not sufficient for even simple design problems, which require expanding the interval constraint propagation concept into a new formalism termed "labelled interval calculus" (Boettner et al. 1992).

Descriptions of conceptual as well as preliminary design represent sets of *systems* or *assemblies* interacting under sets of operating conditions. Descriptions of detail designs represent sets of *components* functioning under sets of operating conditions.

The *labelled interval calculus (LIC)* formalises a system for reasoning about sets. LIC defines a number of operatives on intervals and equations, some of which can be thought of as inverses to the usual notion of interval propagation by the question 'what do the intervals mean?' or, more precisely, 'what kinds of relationships are

possible between a set of values, a variable, and a set of systems or components, each subject to a set of operating conditions?'. The usual notion of an interval constraint is supplemented by the use of labels to indicate relationships between the interval and a set of inferences in the design context. LIC is a fundamental step to understanding fuzzy sets and possibility theory, which will be considered later in detail.

a) Constraint Labels

A *constraint label* describes how a variable is constrained with respect to a given interval of values. The constraint label describes what is known about the values that a variable of a system, assembly, or its components can have under a single set of operating conditions.

There are four constraint labels: *only*, *every*, *some* and *none*. The best approach to understanding the application of these four constraint labels is to give sample descriptions of the values that a particular operating variable would have under a particular set of operating conditions, such as a simple example of a pump assembly that operates under normal operating conditions at pressures ranging from 1,000 to 10,000 kPa.

Only:
$<$ *only p 1000, 10000* $>$ means that the pressure, under the specified operating conditions, takes values only in the interval between 1,000 and 10,000 kPa. Pressure does not take any values outside this interval.

Every:
$<$ *every p 1000, 10000* $>$ means that the pressure, under the specified operating conditions, takes every value in the interval 1,000 to 10,000 kPa. Pressure may or may not take values outside the given interval.

Some:
$<$ *some p 1000, 10000* $>$ means that the pressure, under the specified operating conditions, takes at least one of the values in the interval 1,000 to 10,000 kPa. Pressure may or may not take values outside the given interval.

None:
$<$ *none p 1000, 10000* $>$ means that the pressure, under the specified operating conditions, never takes any of the values in the interval 1,000 to 10,000 kPa.

b) Set Labels

A *set label* consolidates information about the variable values for the entire set of systems or components under consideration. There are two set labels, *all-parts* and *some-part*.

All-parts:
All-parts means the constraint interval is true for every system or component in each selectable subset of the set of systems under consideration. For example, in the case of a series of pumps,

< All-parts only pressure 0, 10000 >
Every pump in the selected subset of the set of systems under consideration operates only under pressures between 0 and 10,000 kPa under the specified operating conditions.

Some-part:
Some-part means the constraint interval is true for at least some system, assembly or component in each selectable subset of the set of systems under consideration.

< Some-part every pressure 0, 10000 >
At least one pump in the selected subset of the set of systems under consideration operates only under pressures between 0 and 10,000 kPa under the specified operating conditions.

c) Labelled Interval Inferences

A method (labelled intervals) is defined for describing sets of systems or equipment being considered for a design, as well as the operatives that can be applied to these intervals. These labelled intervals and operatives can now be used to create inference rules that draw conclusions about the sets of systems under consideration. There are five types of inferences in the labelled interval calculus (Moore 1979):

- Abstraction rules
- Elimination conditions
- Redundancy conditions
- Translation rule
- Propagation rules

Based on the specifications and connections defined in the *conceptual* and *preliminary* design phases, these five labelled interval inferences can be used to reach certain conclusions about the integrity of engineering design.

Abstraction Rules

Abstraction rules are applied to labelled intervals to create subset labelled intervals for selectable items. These subset descriptions can then be used to reason about the design.

There are three abstraction rules:

Abstraction rule 1:

$$(\text{only } X_i)(A_{\text{s},i}, S_i) \rightarrow (\text{only } x \min_i x_{\text{l},i} \max_i x_{\text{h},i})(A \cap_i S_i)$$

Abstraction rule 2:

$$(\text{every } X_i)(A_{\text{s},i}, S_i) \rightarrow (\text{every } x \max_i x_{\text{l},i} \min_i x_{\text{h},i})(A \cap_i S_i)$$

Abstraction rule 3:

$$(\text{some } X_i)(A_{\text{s},i}, S_i) \rightarrow (\text{some } x \min_i x_{\text{l},i} \max_i x_{\text{h},i})(A \cap_i S_i)$$

where

X = variable or operative interval
i = index over the subset
A = set of selectable items
$A_{\text{s},i}$ = ith selectable subset within set of selectable items
S_i = set of states under which the ith subset operates
x = variable or operative
$x_{\text{l},i}$ = lowest x in interval X of the ith selectable subset
$\min_i x_{\text{l},i}$ = the minimum lowest value of x over all subsets i
$\max_i x_{\text{l},i}$ = the maximum lowest value of x over all subsets i
$x_{\text{h},i}$ = highest x in interval X of the ith selectable subset
$\min_i x_{\text{h},i}$ = the minimum highest value of x over all subsets i
$\max_i x_{\text{h},i}$ = the maximum highest value of x over all subsets i
$\cap_i S_i$ = intersection over all i subsets of the set of states.

Again, the best approach to understanding the application of labelled interval inferences for describing sets of systems, assemblies or components being considered for engineering design is to give sample descriptions of the labelled intervals and their computations.

Description of Example

In the conceptual design of a typical engineering process, most sets of systems include a single process vessel that is served by a subset of three centrifugal pumps in parallel. Any two of the pumps are continually operational while the third functions as a standby unit. A basic design problem is the sizing and utilisation of the pumps in order to determine an optimal solution set with respect to various different sets of performance intervals for the pumps. The system therefore includes a subset of three centrifugal pumps in parallel, any two of which are continually operational while one is in reserve, with each pump having the following required pressure ratings:

Pressure ratings:

Pump	Min. pressure	Max. pressure
1	1,000 kPa	10,000 kPa
2	1,000 kPa	10,000 kPa
3	2,000 kPa	15,000 kPa

Labelled intervals:

$X_1 =$ $<$ all-parts every kPa 1000 10000 $>$ (normal)
$X_2 =$ $<$ all-parts every kPa 1000 10000 $>$ (normal)
$X_3 =$ $<$ all-parts every kPa 2000 15000 $>$ (normal)

where
$x_{\mathrm{l},1} = 1{,}000$
$x_{\mathrm{l},2} = 1{,}000$
$x_{\mathrm{l},3} = 2{,}000$
$x_{\mathrm{h},1} = 10{,}000$
$x_{\mathrm{h},2} = 10{,}000$
$x_{\mathrm{h},3} = 15{,}000$

Computation: abstraction rule 2:

(every X_i)($A_{\mathrm{s},i}$, S_i) $\rightarrow$ (every x $\max_i x_{\mathrm{l},i}$ $\min_i x_{\mathrm{h},i}$)($A \cap_i S_i$)
$\max_i x_{\mathrm{l},i} = 2{,}000$
$\min_i x_{\mathrm{h},i} = 10{,}000$

Subset interval:

$<$ all-parts every kPa 2000 10000 $>$ (normal)

Description:

Under normal conditions, *all* the pumps in the subset must be able to operate under *every* value of the interval between 2,000 and 10,000 kPa. The subset interval value must be contained within all of the selectable items' interval values.

Elimination Conditions

Elimination conditions determine those items that do not meet given specifications. In order for these conditions to apply, at least one interval must have an all-parts label, and the state sets must intersect. Each specification is formatted such that there are two labelled intervals and a condition. One labelled interval describes a variable for system requirements, while the other labelled interval describes the same variable of a selectable subset or individual item in the subset.

There are three elimination conditions:

Elimination condition 1:

$$(\text{only } X_1) \text{ and } (\text{only } X_2) \text{ and Not } (X_1 \cap X_2)$$

Elimination condition 2:

$$(\text{only } X_1) \text{ and } (\text{every } X_2) \text{ and Not } (X_2 \subseteq X_1)$$

Elimination condition 3:

$$(\text{only } X_1) \text{ and } (\text{some } X_2) \text{ and Not } (X_1 \cap X_2)$$

Consider the example The system includes a subset of three centrifugal pumps in parallel, any two of which are continually operational, with the following specifications requirement and subset interval:

Specifications:
System requirement: < all-parts only kPa 5000 10000 >

Labelled intervals:
Subset interval: < all-parts every kPa 2000 10000 >

where:
Pump 1 interval: < all-parts every kPa 1000 10000 >
Pump 2 interval: < all-parts every kPa 1000 10000 >
Pump 3 interval: < all-parts every kPa 2000 15000 >

Computation: elimination condition 2:
(only X_1) and (every X_2) and Not ($X_2 \subseteq X_1$)

Subset interval:
System requirement: $X_1 =<$ kPa 5000 10000 >
Subset interval: $X_2 =<$ kPa 2000 10000 >

Elimination result:
Condition: Not ($X_2 \subseteq X_1$) $\Rightarrow$true

Description:
The elimination condition result is true in that the pressure interval of the subset of pumps does not meet the system requirement, where
$X_1 =<$ kPa 5000 10000 >
and the subset interval
$X_2 =<$ kPa 2000 10000 >
A minimum pressure of the subset of pumps (kPa 2,000) cannot be less than the minimum system requirement (kPa 5,000), prompting a review of the conceptual design.

Redundancy Conditions

Redundancy conditions determine if a subset's labelled interval (X_1) is not significant because another subset's labelled interval (X_2) is dominant.

In order for the redundancy conditions to apply, the items set and the state set of the labelled interval (X_1) must be a subset of the items set and state set of the labelled interval (X_2). X_1 must have either an all-parts label or a some-parts label that can be redundant with respect to X_2, which in turn has an all-parts label.

Redundancy conditions do not apply to X_1 having an all-parts label while X_2 has a some-parts label. Each redundancy condition is formatted so that there are two subset labelled intervals and a condition.

There are five redundancy conditions:

Redundancy condition 1:

$$(\text{every } X_1) \text{ and } (\text{every } X_2) \text{ and } (X_1 \subseteq X_2)$$

Redundancy condition 2:

$$(\text{some } X_1) \text{ and } (\text{every } X_2) \text{ and } (X_1 \cap X_2)$$

Redundancy condition 3:

$$(\text{only } X_1) \text{ and } (\text{only } X_2) \text{ and } (X_2 \subseteq X_1)$$

Redundancy condition 4:

$$(\text{some } X_1) \text{ and } (\text{only } X_2) \text{ and } (X_2 \subseteq X_1)$$

Redundancy condition 5:

$$(\text{some } X_1) \text{ and } (\text{some } X_2) \text{ and } (X_2 \subseteq X_1)$$

Consider the example The system includes a subset of three centrifugal pumps in parallel, any two of which are continually operational, with the following specifications requirement and different subset configurations for the two operational units, while the third functions as a standby unit:

Specifications:
System requirement: < all-parts only kPa 1000 10000 >
Pump 1 interval: < all-parts every kPa 1000 10000 >
Pump 2 interval: < all-parts every kPa 1000 10000 >
Pump 3 interval: < all-parts every kPa 2000 15000 >

Labelled intervals:
Subset configuration 1:
Subset1 interval: < all-parts every kPa 1000 10000 >
where:
Pump 1 interval: < all-parts every kPa 1000 10000 >
Pump 2 interval: < all-parts every kPa 1000 10000 >

Subset configuration 2:
Subset2 interval: < all-parts every kPa 2000 10000 >
where:
Pump 1 interval: < all-parts every kPa 1000 10000 >
Pump 3 interval: < all-parts every kPa 2000 15000 >

Subset configuration 3:
Subset3 interval: < all-parts every kPa 2000 10000 >
where:
Pump 2 interval: < all-parts every kPa 1000 10000 >
Pump 3 interval: < all-parts every kPa 2000 15000 >

Computation:
(every X_i)($A_{s,i}$, S_i) $\rightarrow$ (every x $\max_i$ $x_{l,i}$ $\min_i$ $x_{h,i}$)($A \cap_i S_i$)
(every X_1) and (every X_2) and ($X_1 \subseteq X_2$)

For the three subset intervals:
1) Subset intervals:
Subset1 interval: $X_1 =<$ kPa 1000 10000 $>$
Subset2 interval: $X_2 =<$ kPa 2000 10000 $>$

Redundancy result:
Condition: ($X_1 \subseteq X_2$) $\Rightarrow$false

Description:
The redundancy condition result is false in that the pressure interval of the pump subset's labelled interval (X_1) is not a subset of the pump subset's labelled interval (X_2).

2) Subset intervals:
Subset1 interval: $X_1 =<$ kPa 1000 10000 $>$
Subset3 interval: $X_2 =<$ kPa 2000 10000 $>$

Redundancy result:
Condition: ($X_1 \subseteq X_2$) $\Rightarrow$false

Description:
The redundancy condition result is false in that the pressure interval of the pump subset's labelled interval (X_1) is not a subset of the pump subset's labelled interval (X_2).

3) Subset intervals:
Subset2 interval: $X_1 =<$ kPa 2000 10000 $>$
Subset3 interval: $X_2 =<$ kPa 2000 10000 $>$

Redundancy result: Condition: ($X_1 \subseteq X_2$) $\Rightarrow$true

Description:
The redundancy condition result is true in that the pressure interval of the pump subset's labelled interval (X_1) is a subset of the pump subset's labelled interval (X_2).

Conclusion
Subset2 and/or subset3 combinations of pump 1 with pump 3 as well as pump 2 with pump 3 respectively are redundant in that pump 3 is redundant in the configuration of the three centrifugal pumps in parallel.

Translation Rule

The translation rule generates new labelled intervals based on various interrelationships among systems or subsets of systems (equipment). Some components have variables that are directional. (Typically in the case of RPM, a motor produces RPM-out while a pump accepts RPM-in.) When a component such as a motor has a labelled interval that is being considered, the translation rule determines whether it should be translated to a connected component such as a pump if the connected components form a set with matching variables, and the labelled interval for the motor is not redundant in the labelled interval for the pump.

Consider the example A system includes a subset with a motor, transmission and pump where the motor and transmission have the following RPM ratings:

Component	Min. RPM	Max. RPM
Motor	750	1,500
Transmission	75	150

Labelled intervals:
Motor $= <$ all-parts every rpm 750 1500 $>$ (normal)
Transmission $= <$ all-parts every rpm 75 150 $>$ (normal)

Translation rule:
Pump $= <$ all-parts every rpm 75 150 $>$ (normal)

Propagation Rules

Propagation rules generate new labelled intervals based on previously processed labelled intervals and a given relationship *G*, which is implicit among a minimum of three variables. Each rule is formatted so that there are two antecedent subset labelled intervals, a given relationship *G*, and a resultant subset labelled interval. The resultant labelled interval contains a constraint label and a labelled interval calculus operative. The resultant labelled interval is determined by applying the operative to the variables. If the application of the operative on the variables can produce a labelled interval, a new labelled interval is propagated. If the application of the operative on the variables cannot produce a labelled interval, the propagation rule is not valid.

An item's set and state set of the new labelled interval are the intersection of the item's set and state set of the two antecedent labelled intervals. If both of the antecedent labelled intervals have an all-parts set label, the new labelled interval

will have an all-parts set label. If the two antecedent labelled intervals have any other combination of set labels (such as one with a some-part set label, and the other with an all-parts set label; or both with a some-part set label), then the new labelled interval will have a some-part set label (Davis 1987).

There are five propagation rules:

Propagation rule 1:

$$(\text{only } X) \text{ and } (\text{only } Y) \text{ and } G \Rightarrow (\text{only Range } (G, X, Y))$$

Propagation rule 2:

$$(\text{every } X) \text{ and } (\text{every } Y) \text{ and } G \Rightarrow (\text{every Range } (G, X, Y))$$

Propagation rule 3:

$$(\text{every } X) \text{ and } (\text{only } Y) \text{ and state variable } (z) \text{ or parameter } (x) \text{ and } G \Rightarrow (\text{every domain } (G, X, Y))$$

Propagation rule 4:

$$(\text{every } X) \text{ and } (\text{only } Y) \text{ and parameter } (x) \text{ and } G \Rightarrow (\text{only SuffPt } (G, X, Y))$$

Propagation rule 5:

$$(\text{every } X) \text{ and } (\text{only } Y) \text{ and } G \Rightarrow (\text{some SuffPt } (G, X, Y))$$

Consider the example Determine whether the labelled interval of flow for dynamic hydraulic displacement pumps meets the system specifications requirement where the pumps run at revolutions in the interval of 75 to 150 RPM, and the pumps have a displacement capability in the interval 0.5×10^{-3} to 6×10^{-3} cubic metre per revolution. Displacement is the volume of fluid that moves through a hydraulic line per revolution of the pump impellor, and RPM is the revolution speed of the pump. The flow is the rate at which fluid moves through the lines in cubic metres per minute or per hour.

Specifications:
System requirement: $<$ all-parts only flow 1.50 60 $>$ m^3/h

Given relationship:
Flow (m^3/h) = (Displacement $\times$ RPM) $\times C$
where C is the pump constant based on specific pump characteristics.

Labelled intervals:
Displacement (η) $=$ $<$ all-parts only η 0.5×10^{-3} 6×10^{-3} $>$
RPM (ω) $=$ $<$ all-parts only ω 75 150 $>$

Computation:
(only X) and (only Y) and $G \Rightarrow$ (only Range (G, X, Y))
Flow [corners (Q, η, ω)] = (0.0375, 0.075, 0.45, 0.9) $\mathrm{m}^3/\mathrm{min}$
Flow [range (Q, η, ω)] $= <$ flow 2.25 54 $>$ m^3/h

Propagation result: Flow (Q) $= <$ all-parts only flow 2.25 54 $>$

Elimination condition:
(only X_1) and (only X_2) and Not ($X_1 \cap X_2$)

Subset interval:
System requirement: $X_1 = <$ flow 1.50 60 $>$ m^3/h
Subset interval: $X_2 = <$ flow 2.25 54 $>$ m^3/h

Computation:
($X_1 \cap X_2$) $= <$ flow 2.25 54 $>$ m^3/h

Elimination result:
Condition: Not ($X_1 \cap X_2$) $\Rightarrow$true

Description:
With the labelled interval of displacement between 0.5×10^{-3} and 6×10^{-3} cubic metre per revolution and the labelled interval of RPM in the interval of 75 to 150 RPM, the pumps can produce flows only in the interval of 2.25 to 54 m^3/h. The elimination condition is true in that the labelled interval of flow does not meet the system requirement of:
System requirement: $X_1 = <$ flow 1.50 60 $>$ m^3/h
Subset interval: $X_2 = <$ flow 2.25 54 $>$ m^3/h

3.3.1.6 Labelled Interval Calculus in Designing for Reliability

An approach to *designing for reliability* that integrates functional failure as well as functional performance considerations so that a maximum safety margin is achieved with respect to all performance criteria is considered (Thompson et al. 1999). This approach has been expanded to represent sets of systems functioning under sets of failure and performance intervals. The labelled interval calculus (LIC) formalises an approach for reasoning about these sets. The application of LIC in designing for reliability produces a design that has the highest possible safety margin with respect to intervals of performance values relating to specific system datasets. The most significant advantage of this expanded method is that, besides not having to rely on the propagation of single estimated values of failure data, it also does not have to rely on the determination of single values of maximum and minimum acceptable limits of performance for each criterion. Instead, constraint propagation of intervals about sets of performance values is applied, making it possible to compute a multi-objective optimisation of conceptual design solution sets to different sets of performance intervals.

Multi-objective optimisation of conceptual design problems can be computed by applying *LIC inference rules*, which draw conclusions about the sets of systems under consideration to determine optimal solution sets to different intervals of performance values. Considering the performance limits represented diagrammatically in Figs. 3.23, 3.24 and 3.25, where an example of two performance limits, one upper performance limit, and one lower performance limit is given, the determination of datasets using LIC would include the following.

a) Determination of a Data Point: Two Sets of Limit Intervals

The proximity of actual performance to the minimum, nominal or maximum sets of limit intervals of performance for each performance criterion relates to a measure of the safety margin *range*.

The data point x_{ij} is the value closest to the nominal design condition that approaches either minimum or maximum limit interval. The value of x_{ij} always lies in the range 0–10. Ideally, when the design condition is at the mid-range, then the data point is 10. A set of data points can thus be obtained for each system with respect to the performance parameters that are relevant to that system. In this case, the data point x_{ij} approaching the *maximum limit interval* is the performance variable of temperature

$$x_{ij} = \frac{\text{Max. Temp. } T_1 - \text{Nom. } T \text{ High } (\times 20)}{\text{Max. Temp. } T_1 - \text{Min. Temp. } T_2} \tag{3.83}$$

Given relationship: dataset:
$(\text{Max. Temp. } T_1 - \text{Nom. } T \text{ High})/(\text{Max. Temp. } T_1 - \text{Min. Temp. } T_2) \times 20$

where
Max. Temp. T_1 = maximum performance interval
Min. Temp. T_2 = minimum performance interval
Nom. T High = nominal performance interval high

Labelled intervals:
Max. Temp. $T_1 = <$ all-parts only $T_1 t_{1l} t_{1h} >$
Min. Temp. $T_2 = <$ all-parts only $T_2 t_{2l} t_{2h} >$
Nom.T High $= <$ all-parts only $T_H t_{Hl} t_{Hh} >$

where
t_{1l} = lowest temperature value in interval of maximum performance interval.
t_{1h} = highest temperature value in interval of maximum performance interval.
t_{2l} = lowest temperature value in interval of minimum performance interval.
t_{2h} = highest temperature value in interval of minimum performance interval.

t_{Hl} = lowest temperature value in interval of nominal performance interval high.
t_{Hh} = highest temperature value in interval of nominal performance interval high.

Computation: propagation rule 1:
(only X) and (only Y) and $G \Rightarrow$ (only Range (G, X, Y))

$$
\begin{aligned}
& x_{ij} \text{ [corners (Max. Temp. } T_1\text{, Nom. } T \text{ High, Min. Temp. } T_2\text{)]} \\
&= (t_{1\mathrm{h}} - t_{\mathrm{Hl}}/t_{1\mathrm{l}} - t_{2\mathrm{h}}) \times 20\,, \quad (t_{1\mathrm{h}} - t_{\mathrm{Hl}}/t_{1\mathrm{l}} - t_{2\mathrm{l}}) \times 20\,, \\
&\quad (t_{1\mathrm{h}} - t_{\mathrm{Hl}}/t_{1\mathrm{h}} - t_{2\mathrm{h}}) \times 20\,, \quad (t_{1\mathrm{h}} - t_{\mathrm{Hl}}/t_{1\mathrm{h}} - t_{2\mathrm{l}}) \times 20\,, \\
&\quad (t_{1\mathrm{l}} - t_{\mathrm{Hl}}/t_{1\mathrm{l}} - t_{2\mathrm{h}}) \times 20\,, \quad (t_{1\mathrm{l}} - t_{\mathrm{Hl}}/t_{1\mathrm{l}} - t_{2\mathrm{l}}) \times 20\,, \\
&\quad (t_{1\mathrm{l}} - t_{\mathrm{Hl}}/t_{1\mathrm{h}} - t_{2\mathrm{h}}) \times 20\,, \quad (t_{1\mathrm{l}} - t_{\mathrm{Hl}}/t_{1\mathrm{h}} - t_{2\mathrm{l}}) \times 20\,, \\
&\quad (t_{1\mathrm{h}} - t_{\mathrm{Hh}}/t_{1\mathrm{l}} - t_{2\mathrm{h}}) \times 20\,, \quad (t_{1\mathrm{h}} - t_{\mathrm{Hh}}/t_{1\mathrm{l}} - t_{2\mathrm{l}}) \times 20\,, \\
&\quad (t_{1\mathrm{h}} - t_{\mathrm{Hh}}/t_{1\mathrm{h}} - t_{2\mathrm{h}}) \times 20\,, \quad (t_{1\mathrm{h}} - t_{\mathrm{Hh}}/t_{1\mathrm{h}} - t_{2\mathrm{l}}) \times 20\,, \\
&\quad (t_{1\mathrm{l}} - t_{\mathrm{Hh}}/t_{1\mathrm{l}} - t_{2\mathrm{h}}) \times 20\,, \quad (t_{1\mathrm{l}} - t_{\mathrm{Hh}}/t_{1\mathrm{l}} - t_{2\mathrm{l}}) \times 20\,, \\
&\quad (t_{1\mathrm{l}} - t_{\mathrm{Hh}}/t_{1\mathrm{h}} - t_{2\mathrm{h}}) \times 20\,, \quad (t_{1\mathrm{l}} - t_{\mathrm{Hh}}/t_{1\mathrm{h}} - t_{2\mathrm{l}}) \times 20\,,
\end{aligned}
$$

$$
\begin{aligned}
& x_{ij} \text{ [range (Max. Temp. } T_1\text{, Nom. } T \text{ High, Min. Temp. } T_2\text{)]} \\
&= (t_{1\mathrm{l}} - t_{\mathrm{Hh}}/t_{1\mathrm{h}} - t_{2\mathrm{l}}) \times 20\,, \quad (t_{1\mathrm{h}} - t_{\mathrm{Hl}}/t_{1\mathrm{l}} - t_{2\mathrm{h}}) \times 20
\end{aligned}
$$

Propagation result:
$x_{ij} = <$ all-parts only
$x_{ij}\ (t_{1\mathrm{l}} - t_{\mathrm{Hh}}/t_{1\mathrm{h}} - t_{2\mathrm{l}}) \times 20\,, \quad (t_{1\mathrm{h}} - t_{\mathrm{Hl}}/t_{1\mathrm{l}} - t_{2\mathrm{h}}) \times 20 >$
where x_{ij} is dimensionless.

Description:
The generation of data points with respect to performance limits using the labelled interval calculus, approaching the *maximum limit interval*.
This is where the data point x_{ij} approaching the *maximum limit interval*, with x_{ij} in the range (Max. Temp. T_1, Nom. T High, Min. Temp. T_2), and the data point x_{ij} being dimensionless, has a propagation result equivalent to the following labelled interval:
$<$ all-parts only $x_{ij}\ (t_{1\mathrm{l}} - t_{\mathrm{Hh}}/t_{1\mathrm{h}} - t_{2\mathrm{l}}) \times 20\,, \quad (t_{1\mathrm{h}} - t_{\mathrm{Hl}}/t_{1\mathrm{l}} - t_{2\mathrm{h}}) \times 20 >$, which represents the relationship:

$$
x_{ij} = \frac{\text{Max. Temp. } T_1 - \text{Nom. } T \text{ High } (\times 20)}{\text{Max. Temp. } T_1 - \text{Min. Temp. } T_2}
$$

In the case of the data point x_{ij} approaching the *minimum limit interval*, where the performance variable is temperature

$$
x_{ij} = \frac{\text{Nom. } T \text{ Low} - \text{Min. Temp. } T_2\ (\times 20)}{\text{Max. Temp. } T_1 - \text{Min. Temp. } T_2} \tag{3.84}
$$

Given relationship: dataset:
$(\text{Max. Temp. } T_1 - \text{Nom. } T \text{ High})/(\text{Max. Temp. } T_1 - \text{Min. Temp. } T_2) \times 20$

where
Max. Temp. T_1 = maximum performance interval
Min. Temp. T_2 = minimum performance interval
Nom. T Low = nominal performance interval low

Labelled intervals:
Max. Temp. $T_1 = <$ all-parts only $T_1 t_{1l} t_{1h} >$
Min. Temp. $T_2 = <$ all-parts only $T_2 t_{2l} t_{2h} >$
Nom. T Low $= <$ all-parts only $T_L t_{Ll} t_{Lh} >$

where
t_{1i} = lowest temperature value in interval of maximum performance interval
t_{1h} = highest temperature value in interval of maximum performance interval
t_{2l} = lowest temperature value in interval of minimum performance interval
t_{2h} = highest temperature value in interval of minimum performance interval
t_{Ll} = lowest temperature value in interval of nominal performance interval low
t_{Lh} = highest temperature value in interval of nominal performance interval low

Computation: propagation rule 1:
(only X) and (only Y) and $G \Rightarrow$ (only Range (G, X, Y))

$$
\begin{aligned}
& x_{ij} \text{ [corners (Max. Temp. } T_1\text{, Nom. } T \text{ High, Min. Temp. } T_2\text{)]} \\
&= (t_{Lh} - t_{2l}/t_{1l} - t_{2h}) \times 20\,, \quad (t_{Lh} - t_{2l}/t_{1l} - t_{2l}) \times 20\,, \\
&\quad (t_{Lh} - t_{2l}/t_{1h} - t_{2h}) \times 20\,, \quad (t_{Lh} - t_{2l}/t_{1h} - t_{2l}) \times 20\,, \\
&\quad (t_{Ll} - t_{2l}/t_{1l} - t_{2h}) \times 20\,, \quad (t_{Ll} - t_{2l}/t_{1l} - t_{2l}) \times 20\,, \\
&\quad (t_{Ll} - t_{2l}/t_{1h} - t_{2h}) \times 20\,, \quad (t_{Ll} - t_{2l}/t_{1h} - t_{2l}) \times 20\,, \\
&\quad (t_{Lh} - t_{2h}/t_{1l} - t_{2h}) \times 20\,, \quad (t_{Lh} - t_{2h}/t_{1l} - t_{2l}) \times 20\,, \\
&\quad (t_{Lh} - t_{2h}/t_{1h} - t_{2h}) \times 20\,, \quad (t_{Lh} - t_{2h}/t_{1h} - t_{2l}) \times 20\,, \\
&\quad (t_{Ll} - t_{2h}/t_{1l} - t_{2h}) \times 20\,, \quad (t_{Ll} - t_{2h}/t_{1l} - t_{2l}) \times 20\,, \\
&\quad (t_{Ll} - t_{2h}/t_{1h} - t_{2h}) \times 20\,, \quad (t_{Ll} - t_{2h}/t_{1h} - t_{2l}) \times 20\,,
\end{aligned}
$$

$$
\begin{aligned}
& x_{ij} \text{ [range (Max. Temp. } T_1\text{, Nom.}T \text{ High, Min. Temp. } T_2\text{)]} \\
&= (t_{Ll} - t_{2h}/t_{1h} - t_{2l}) \times 20\,, \quad (t_{Lh} - t_{2l}/t_{1l} - t_{2h}) \times 20
\end{aligned}
$$

Propagation result:
$x_{ij} = <$ all-parts only
$x_{ij}(t_{\mathrm{Ll}} - t_{2\mathrm{h}}/t_{1\mathrm{h}} - t_{2\mathrm{l}}) \times 20\,, \quad (t_{\mathrm{Lh}} - t_{2\mathrm{l}}/t_{1\mathrm{l}} - t_{2\mathrm{h}}) \times 20 >$
where x_{ij} is dimensionless.

Description:
The generation of data points with respect to performance limits using the labelled interval calculus, in the case of the data point x_{ij} approaching the *minimum limit interval*, with x_{ij} in the range (Max. Temp. T_1, Nom. T High, Min. Temp. T_2), and x_{ij} dimensionless, has a propagation result equivalent to the following labelled interval:
$<$ all-parts only $x_{ij}(t_{\mathrm{Ll}} - t_{2\mathrm{h}}/t_{1\mathrm{h}} - t_{2\mathrm{l}}) \times 20\,, \quad (t_{\mathrm{Lh}} - t_{2\mathrm{l}}/t_{1\mathrm{l}} - t_{2\mathrm{h}}) \times 20 >$
which represents the relationship:

$$x_{ij} = \frac{\text{Nom. } T \text{ Low} - \text{Min. Temp. } T_2\ (\times 20)}{\text{Max. Temp. } T_1 - \text{Min. Temp. } T_2}$$

b) Determination of a Data Point: One Upper Limit Interval

If there is one operating limit set only, then the data point is obtained as shown in Figs. 3.24 and 3.25, where the upper or lower limit is known. A set of data points can be obtained for each system with respect to the performance parameters that are relevant to that system. In the case of the data point x_{ij} approaching the upper limit interval

$$x_{ij} = \frac{\text{Highest Stress Level} - \text{Nominal Stress Level}\ (\times 10)}{\text{Highest Stress Level} - \text{Lowest Stress Est.}} \tag{3.85}$$

Given relationship: dataset:
$(\mathrm{HSL} - \mathrm{NSL})/(\mathrm{HSL} - \mathrm{LSL}) \times 10$

Labelled intervals:
HSI = highest stress interval $<$ all-parts only HSI $s_{1\mathrm{l}} s_{1\mathrm{h}} >$
LSI = lowest stress interval $<$ all-parts only LSI $s_{2\mathrm{l}} s_{2\mathrm{h}} >$
NSI = nominal stress interval $<$ all-parts only NSI $s_{\mathrm{Hl}} s_{\mathrm{Hh}} >$

where:
$s_{1\mathrm{l}}$ = lowest stress value in interval of highest stress interval
$s_{1\mathrm{h}}$ = highest stress value in interval of highest stress interval
$s_{2\mathrm{l}}$ = lowest stress value in interval of lowest stress interval
$s_{2\mathrm{h}}$ = highest stress value in interval of lowest stress interval
s_{Hl} = lowest stress value in interval of nominal stress interval
s_{Hh} = highest stress value in interval of nominal stress interval

Computation: propagation rule 1:
(only X) and (only Y) and $G \Rightarrow$(only Range (G, X, Y))

$$
\begin{aligned}
& x_{ij} \text{ [corners (HSL, NSL, LSL)]} \\
&= (s_{1\mathrm{h}} - s_{\mathrm{Hl}}/s_{1\mathrm{l}} - s_{2\mathrm{h}}) \times 10\,, \quad (s_{1\mathrm{h}} - s_{\mathrm{Hl}}/s_{1\mathrm{l}} - s_{2\mathrm{l}}) \times 10\,, \\
&\quad (s_{1\mathrm{h}} - s_{\mathrm{Hl}}/s_{1\mathrm{h}} - s_{2\mathrm{h}}) \times 10\,, \quad (s_{1\mathrm{h}} - s_{\mathrm{Hl}}/s_{1\mathrm{h}} - s_{2\mathrm{l}}) \times 10\,, \\
&\quad (s_{1\mathrm{l}} - s_{\mathrm{Hl}}/s_{1\mathrm{l}} - s_{2\mathrm{h}}) \times 10\,, \quad (s_{1\mathrm{l}} - s_{\mathrm{Hl}}/s_{1\mathrm{l}} - s_{2\mathrm{l}}) \times 10\,, \\
&\quad (s_{1\mathrm{l}} - s_{\mathrm{Hl}}/s_{1\mathrm{h}} - s_{2\mathrm{h}}) \times 10\,, \quad (s_{1\mathrm{l}} - s_{\mathrm{Hl}}/s_{1\mathrm{h}} - s_{2\mathrm{l}}) \times 10\,, \\
&\quad (s_{1\mathrm{h}} - s_{\mathrm{Hh}}/s_{1\mathrm{l}} - s_{2\mathrm{h}}) \times 10\,, \quad (s_{1\mathrm{h}} - s_{\mathrm{Hh}}/s_{1\mathrm{l}} - s_{2\mathrm{l}}) \times 10\,, \\
&\quad (s_{1\mathrm{h}} - s_{\mathrm{Hh}}/s_{1\mathrm{h}} - s_{2\mathrm{h}}) \times 10\,, \quad (s_{1\mathrm{h}} - s_{\mathrm{Hh}}/s_{1\mathrm{h}} - s_{2\mathrm{l}}) \times 10\,, \\
&\quad (s_{1\mathrm{l}} - s_{\mathrm{Hh}}/s_{1\mathrm{l}} - s_{2\mathrm{h}}) \times 10\,, \quad (s_{1\mathrm{l}} - s_{\mathrm{Hh}}/s_{1\mathrm{l}} - s_{2\mathrm{l}}) \times 10\,, \\
&\quad (s_{1\mathrm{l}} - s_{\mathrm{Hh}}/s_{1\mathrm{h}} - s_{2\mathrm{h}}) \times 10\,, \quad (s_{1\mathrm{l}} - s_{\mathrm{Hh}}/s_{1\mathrm{h}} - s_{2\mathrm{l}}) \times 10\,,
\end{aligned}
$$

$$
\begin{aligned}
& x_{ij} \text{ [range (HSL, NSL, LSL)]} \\
&= (s_{1\mathrm{l}} - s_{\mathrm{Hh}}/s_{1\mathrm{h}} - s_{2\mathrm{l}}) \times 10\,, \quad (s_{1\mathrm{h}} - s_{\mathrm{Hl}}/s_{1\mathrm{l}} - s_{2\mathrm{h}}) \times 10
\end{aligned}
$$

Propagation result:
$x_{ij} = <$ all-parts only
$x_{ij}(s_{1\mathrm{l}} - s_{\mathrm{Hh}}/s_{1\mathrm{h}} - s_{2\mathrm{l}}) \times 10\,, \quad (s_{1\mathrm{h}} - s_{\mathrm{Hl}}/s_{1\mathrm{l}} - s_{2\mathrm{h}}) \times 10 >$
where x_{ij} is dimensionless.

Description:
The data point x_{ij} approaching the *upper limit interval*, with x_{ij} in the range (High Stress Level, Nominal Stress Level, Lowest Stress Level), and x_{ij} dimensionless, has a propagation result equivalent to the following labelled interval:
$<$ all-parts only $x_{ij}(s_{\mathrm{Ll}} - s_{2\mathrm{h}}/s_{1\mathrm{h}} - s_{2\mathrm{l}}) \times 20\,, \quad (s_{\mathrm{Lh}} - s_{2\mathrm{l}}/s_{1\mathrm{l}} - s_{2\mathrm{h}}) \times 20 >$,
which represents the relationship:

$$
x_{ij} = \frac{\text{Highest Stress Level} - \text{Nominal Stress Level } (\times 10)}{\text{Highest Stress Level} - \text{Lowest Stress Est.}}
$$

c) Determination of a Data Point: One Lower Limit Interval

In the case of the data point x_{ij} approaching the lower limit interval

$$
x_{ij} = \frac{\text{Nominal Capacity} - \text{Min. Capacity Level } (\times 10)}{\text{Max. Capacity Est.} - \text{Min. Capacity Level}} \tag{3.86}
$$

Given relationship: dataset:
$(\text{Nom. Cap. L} - \text{Min. Cap. L})/(\text{Max. Cap. L} - \text{Min. Cap. L}) \times 10$

where
Max. Cap. C_1 = maximum capacity interval
Min. Cap. C_2 = minimum capacity interval
Nom. Cap. C_{L} = nominal capacity interval low

Labelled intervals:
Max. Cap. $C_1 = <$ all-parts only $C_1 c_{1l} c_{1h} >$
Min. Cap. $C_2 = <$ all-parts only $C_2 c_{2l} c_{2h} >$
Nom. Cap. $C_L = <$ all-parts only $C_L c_{Ll} c_{Lh} >$

where
c_{1l} = lowest capacity value in interval of maximum capacity interval
c_{1h} = highest capacity value in interval of maximum capacity interval
c_{2l} = lowest capacity value in interval of minimum capacity interval
c_{2h} = highest capacity value in interval of minimum capacity interval
c_{Ll} = lowest capacity value in interval of nominal capacity interval low
c_{Lh} = highest capacity value in interval of nominal capacity interval low

Computation: propagation rule 1:
(only X) and (only Y) and $G \Rightarrow$ (only Range (G, X, Y))

$$
\begin{aligned}
& x_{ij} \text{ [corners (Max. Cap. Min. Cap. } C_2\text{, Nom. Cap. } C_L\text{)]} \\
&= (c_{Lh} - c_{2l}/c_{1l} - c_{2h}) \times 10\,, \quad (c_{Lh} - c_{2l}/c_{1l} - c_{2l}) \times 10\,, \\
&\quad (c_{Lh} - c_{2l}/c_{1h} - c_{2h}) \times 10\,, \quad (c_{Lh} - c_{2l}/c_{1h} - c_{2l}) \times 10\,, \\
&\quad (c_{Ll} - c_{2l}/c_{1l} - c_{2h}) \times 10\,, \quad (c_{Ll} - c_{2l}/c_{1l} - c_{2l}) \times 10\,, \\
&\quad (c_{Ll} - c_{2l}/c_{1h} - c_{2h}) \times 10\,, \quad (c_{Ll} - c_{2l}/c_{1h} - c_{2l}) \times 10\,, \\
&\quad (c_{Lh} - c_{2h}/c_{1l} - c_{2h}) \times 10\,, \quad (c_{Lh} - c_{2h}/c_{1l} - c_{2l}) \times 10\,, \\
&\quad (c_{Lh} - c_{2h}/c_{1h} - c_{2h}) \times 10\,, \quad (c_{Lh} - c_{2h}/c_{1h} - c_{2l}) \times 10\,, \\
&\quad (c_{Ll} - c_{2h}/c_{1l} - c_{2h}) \times 10\,, \quad (c_{Ll} - c_{2h}/c_{1l} - c_{2l}) \times 10\,, \\
&\quad (c_{Ll} - c_{2h}/c_{1h} - c_{2h}) \times 10\,, \quad (c_{Ll} - c_{2h}/c_{1h} - c_{2l}) \times 10\,,
\end{aligned}
$$

$$
\begin{aligned}
& x_{ij} \text{ [range (Max. Cap. Min. Cap. } C_2\text{, Nom. Cap. } C_L\text{)]} \\
&= (c_{Ll} - c_{2h}/c_{1h} - c_{2l}) \times 10\,, \quad (c_{Lh} - c_{2l}/c_{1l} - c_{2h}) \times 10
\end{aligned}
$$

Propagation result:
$x_{ij} = <$ all-parts only
$x_{ij}(c_{Ll} - c_{2h}/c_{1h} - c_{2l}) \times 10\,, \quad (c_{Lh} - c_{2l}/c_{1l} - c_{2h}) \times 10 >$
where x_{ij} is dimensionless.

Description:
The generation of data points with respect to performance limits using the labelled interval calculus for the *lower limit interval* is the following:

The data point x_{ij} approaching the *lower limit interval*, with x_{ij} in the range (Max. Capacity Level, Min. Capacity Level, Nom. Capacity Level), and x_{ij} dimensionless, has a propagation result equivalent to the following labelled interval:
$<$ all-parts only $x_{ij}(c_{\mathrm{Ll}} - c_{2\mathrm{h}}/c_{1\mathrm{h}} - c_{2\mathrm{l}}) \times 10\,, \quad (c_{\mathrm{Lh}} - c_{2\mathrm{l}}/c_{1\mathrm{l}} - c_{2\mathrm{h}}) \times 10\ >$
with x_{ij} in the range (Max. Cap. Min. Cap. C_2, Nom. Cap. C_L), representing the relationship:

$$x_{ij} = \frac{\text{Nominal Capacity} - \text{Min. Capacity Level}(\times 10)}{\text{Max. Capacity Est.} - \text{Min. Capacity Level}}$$

d) Analysis of the Interval Matrix

In Fig. 3.26, the performance measures of each system of a process are described in matrix form containing *data points* relating to process systems and *single parameters* that describe their performance. The matrix can be analysed by rows and columns in order to evaluate the performance characteristics of the process. Each *data point* of x_{ij} refers to a single parameter. Similarly, in the expanded method using labelled interval calculus (LIC), the performance measures of each system of a process are described in an *interval matrix* form, containing *datasets* relating to systems and *labelled intervals* that describe their performance. Each row of the interval matrix reveals whether the process has a consistent safety margin with respect to a specific set of performance values.

A *parameter performance index, PPI*, can be calculated for each *row*

$$\mathrm{PPI} = n\left(\sum_{j=1}^{n} 1/x_{ij}\right)^{-1} \tag{3.87}$$

where n is the number of systems in row i.

The calculation of PPI is accomplished using LIC inference rules that draw conclusions about the system *datasets* of each matrix row under consideration. The numerical value of PPI lies in the range 0–10, irrespective of the number of *datasets* in each row (i.e. the number of process systems). A comparison of PPIs can be made to judge whether specific performance criteria, such as reliability, are acceptable.

Similarly, a *system performance index*, *SPI*, can be calculated for each *column* as

$$\mathrm{SPI} = m\left(\sum_{i=1}^{m} 1/x_{ij}\right)^{-1} \tag{3.88}$$

where m is the number of parameters in column i.

The calculation of SPI is accomplished using LIC inference rules that draw conclusions about performance *labelled intervals* of each matrix column under consideration. The numerical value of SPI also lies in the range 0–10, irrespective of the number of *labelled intervals* in each column (i.e. the number of performance

parameters). A comparison of SPIs can be made to assess whether there is acceptable performance with respect to any performance criteria of a specific system.

Finally, an *overall performance index*, *OPI*, can be calculated (Eq. 3.89). The numerical value of OPI lies in the range 0–100 and can be indicated as a percentage value.

$$\mathrm{OPI} = \frac{1}{mn}\left(\sum_{i=1}^{m}\sum_{j=1}^{n}(\mathrm{PPI})(\mathrm{SPI})\right) \tag{3.89}$$

where m is the number of performance parameters, and n is the number of systems.

Description of Example

Acidic gases, such as sulphur dioxide, are removed from the combustion gas emissions of a non-ferrous metal smelter by passing these through a reverse jet scrubber. A reverse jet scrubber consists of a scrubber vessel containing jet-spray nozzles adapted to spray, under high pressure, a caustic scrubbing liquid counter to the high-velocity combustion gas stream emitted by the smelter, whereby the combustion gas stream is scrubbed and a clear gas stream is recovered downstream. The reverse jet scrubber consists of a scrubber vessel and a subset of three centrifugal pumps in parallel, any two of which are continually operational, with the following labelled intervals for the specific performance parameters (Tables 3.10 and 3.11):

Propagation result:
$x_{ij} = <$ all-parts only
$x_{ij}(x_{1l} - x_{Hh}/x_{1h} - x_{2l}) \times 10\,, \quad (x_{1h} - x_{Hl}/x_{1l} - x_{2h}) \times 10\ >$

Table 3.10 Labelled intervals for specific performance parameters

Parameters	Vessel	Pump 1	Pump 2	Pump 3
Max. flow	< 65 75 >	< 55 60 >	< 55 60 >	< 65 70 >
Min. flow	< 30 35 >	< 20 25 >	< 20 25 >	< 30 35 >
Nom. flow	< 50 60 >	< 40 50 >	< 40 50 >	< 50 60 >
Max. pressure	< 10000 12500 >	< 8500 10000 >	< 8500 10000 >	< 12500 15000 >
Min. pressure	< 1000 1500 >	< 1000 1250 >	< 1000 1250 >	< 2000 2500 >
Nom. pressure	< 5000 7500 >	< 5000 6500 >	< 5000 6500 >	< 7500 10000 >
Max. temp.	< 80 85 >	< 85 90 >	< 85 90 >	< 80 85 >
Min. temp.	< 60 65 >	< 60 65 >	< 60 65 >	< 55 60 >
Nom. temp.	< 70 75 >	< 75 80 >	< 75 80 >	< 70 75 >

Table 3.11 Parameter interval matrix

Parameters	Vessel	Pump 1	Pump 2	Pump 3
Flow (m^3/h)	< 1.1 8.3 >	< 1.3 6.7 >	< 1.3 6.7 >	< 1.1 8.3 >
Pressure (kPa)	< 2.2 8.8 >	< 2.2 6.9 >	< 2.2 6.9 >	< 1.9 7.5 >
Temp. (°C)	< 2.0 10.0 >	< 1.7 7.5 >	< 1.7 7.5 >	< 1.7 5.0 >

Labelled intervals—flow:
Vessel interval: $= <$ all-parts only x_{ij} 1.1 8.3 $>$
Pump 1 interval: $= <$ all-parts only x_{ij} 1.3 6.7 $>$
Pump 2 interval: $= <$ all-parts only x_{ij} 1.3 6.7 $>$
Pump 3 interval: $= <$ all-parts only x_{ij} 1.1 8.3 $>$

Labelled intervals—pressure:
Vessel interval: $= <$ all-parts only x_{ij} 2.2 8.8 $>$
Pump 1 interval: $= <$ all-parts only x_{ij} 2.2 6.9 $>$
Pump 2 interval: $= <$ all-parts only x_{ij} 2.2 6.9 $>$
Pump 3 interval: $= <$ all-parts only x_{ij} 1.9 7.5 $>$

Labelled intervals—temperature:
Vessel interval: $= <$ all-parts only x_{ij} 2.0 10.0 $>$
Pump 1 interval: $= <$ all-parts only x_{ij} 1.7 7.5 $>$
Pump 2 interval: $= <$ all-parts only x_{ij} 1.7 7.5 $>$
Pump 3 interval: $= <$ all-parts only x_{ij} 1.7 5.0 $>$

The *parameter performance index, PPI*, can be calculated for each *row*

$$\text{PPI} = n\left(\sum_{j=1}^{n} 1/x_{ij}\right)^{-1} \tag{3.90}$$

where n is the number of systems in row i.

Labelled intervals:
Flow (m^3/h) PPI $= <$ all-parts only PPI 1.2 7.4 $>$
Pressure (kPa) PPI $= <$ all-parts only PPI 2.1 7.5 $>$
Temp. ($^\circ$C) PPI $= <$ all-parts only PPI 1.8 7.1 $>$

The *system performance index, SPI*, can be calculated for each *column*

$$\text{SPI} = m\left(\sum_{i=1}^{m} 1/x_{ij}\right)^{-1} \tag{3.91}$$

where m is the number of parameters in column i.

Labelled intervals:
Vessel SPI $= <$ all-parts only 1.6 9.0 $>$
Pump 1 SPI $= <$ all-parts only 1.7 7.0 $>$
Pump 2 SPI $= <$ all-parts only 1.7 7.0 $>$
Pump 3 SPI $= <$ all-parts only 1.5 6.6 $>$

Description:
The *parameter performance index, PPI*, and the *system performance index, SPI*, indicate whether there is acceptable overall performance of the operational parameters (PPI), and what contribution an item makes to the overall effectiveness of the system (SPI).

The *overall performance index, OPI,* can be calculated as

$$\mathrm{OPI} = \frac{1}{mn}\left(\sum_{i=1}^{m}\sum_{j=1}^{n}(\mathrm{PPI})(\mathrm{SPI})\right) \tag{3.92}$$

where m is the number of performance parameters, and n is the number of systems.

Computation: propagation rule 1:
(only X) and (only Y) and $G \Rightarrow$ (only Range (G, X, Y))

$$\begin{aligned}
&\text{OPI [corners (PPI, SPI)]}\\
&\quad = [1/12 \times ((1.2 \times 1.6) + (1.2 \times 1.7) + (1.2 \times 1.7) + (1.2 \times 1.5)\\
&\qquad + (2.1 \times 1.6) + (2.1 \times 1.7) + (2.1 \times 1.7) + (2.1 \times 1.5)\\
&\qquad + (1.8 \times 1.6) + (1.8 \times 1.7) + (1.8 \times 1.7) + (1.8 \times 1.5))]\,,\\
&\qquad [1/12 \times ((7.4 \times 9.0) + (7.4 \times 7.0) + (7.4 \times 7.0) + (7.4 \times 6.6)\\
&\qquad + (7.5 \times 9.0) + (7.5 \times 7.0) + (7.5 \times 7.0) + (7.5 \times 6.6)\\
&\qquad + (7.1 \times 9.0) + (7.1 \times 7.0) + (7.1 \times 7.0) + (7.1 \times 6.6))]
\end{aligned}$$

$$\begin{aligned}
&\text{OPI [range (PPI, SPI)]}\\
&\quad = < [1/12 \times 33.2]\,, \quad [1/12 \times 651.2] >\\
&\text{and:}\\
&\text{OPI} = < \text{all-parts only \%2.8 54.3} >
\end{aligned}$$

Description:
The *overall performance index, OPI,* is a combination of the *parameter performance index, PPI,* and the *system performance index, SPI,* and indicates the overall performance of the operational parameters (PPI), and the overall contribution of the system's items on the system (SPI) itself.
The numerical value of OPI lies in the range 0–100 and can thus be indicated as a percentage value, which is a useful measure for conceptual design optimisation. The reverse jet scrubber system has an overall performance in the range of 2.8 to 54%, which is not optimal.
The critical minimum performance level of 2.8% *as well as the upper performance level of* 54% *indicate design review.*

3.3.2 Analytic Development of Reliability Assessment in Preliminary Design

The most applicable techniques selected as tools for *reliability assessment* in intelligent computer automated methodology for determining the integrity of engineering

design during the *preliminary* or *schematic design* phase are failure modes and effects analysis (FMEA), failure modes and effects criticality analysis (FMECA), and fault-tree analysis. However, as the main use of fault-tree analysis is perceived to be in *designing for safety*, whereby fault trees provide a useful representation of the different failure paths that can lead to safety and risk assessments of systems and processes, this technique will be considered in greater detail in Chap. 5, Safety and Risk in Engineering Design. Thus, only FMEA and FMECA are further developed at this stage with respect to the following:

i. *FMEA and FMECA in engineering design analysis*
ii. *Algorithmic modelling in failure modes and effects analysis*
iii. *Qualitative reasoning in failure modes and effects analysis*
iv. *Overview of fuzziness in engineering design analysis*
v. *Fuzzy logic and fuzzy reasoning*
vi. *Theory of approximate reasoning*
vii. *Overview of possibility theory*
viii. *Uncertainty and incompleteness in design analysis*
ix. *Modelling uncertainty in FMEA and FMECA*
x. *Development of a qualitative FMECA.*

3.3.2.1 FMEA and FMECA in Engineering Design Analysis

Systems can be described in terms of hierarchical *system breakdown structures (SBS)*. These system structures are comprised of many sub-systems, assemblies and components (and parts), which can fail at one time or another. The effect of *functional failure* of the system structures on the system as a whole can vary, and can have a direct, indirect or no adverse effect on the performance of the system. In a systems context, any direct or indirect effect of equipment functional failures will result in a change to the *reliability* of the system or equipment, but may not necessarily result in a change to the *performance* of the system.

Equipment (i.e. assemblies and components) showing functional failures that degrade system performance, or render the system inoperative, is termed *system-critical*. Equipment functional failures that degrade the reliability of the system are classified as *reliability-critical* (Aslaksen et al. 1992).

a) Reliability-Critical Items

Reliability-critical items are those items that *can* have a quantifiable impact on *system performance* but predominantly on *system reliability*. These items are usually identified by appropriate reliability analysis techniques. The identification of reliability-critical items is an essential portion of engineering design analysis, especially since the general trend in the design of process engineering installations is towards increasing system complexity. It is thus imperative that a systematic method for identifying reliability-critical items is implemented during the

engineering design process, particularly during *preliminary design*. Such a systematic method is failure modes and effects criticality analysis (FMECA). In practice, however, development of FMECA procedures have often been considered to be arduous and time consuming. As a result, the benefits that can be derived have often been misunderstood and not fully appreciated. The FMECA procedure consists of three inherent sub-methods:

- Failure modes and effects analysis (FMEA).
- Failure hazard analysis.
- Criticality analysis.

The methods of *failure modes and effects analysis*, *failure hazard analysis* and *criticality analysis* are interrelated. Failure hazard analysis and criticality analysis cannot be effectively implemented without the prior preparations for failure modes and effects analysis. Once certain groundwork has been completed, all of these analysis methods should be applied. This groundwork includes a detailed understanding of the *functions* of the system under consideration, and the *functional relationships* of its constituent components. Therefore, two necessary additional techniques are imperative *prior* to developing FMEA procedures, namely:

- Systems breakdown structuring.
- Functional block diagramming.

As previously indicated, a *systems breakdown structure (SBS)* can be defined as "*a systematic hierarchical representation of equipment, grouped into its logical systems, sub-systems, assemblies, sub-assemblies, and component levels*".

A *functional block diagram (FBD)* can be defined as "*an orderly and structured means for describing component functional relationships for the purpose of systems analysis*".

An FBD is a combination of an SBS and concise descriptions of the *operational* and *physical functions* and *functional relationships* at component level. Thus, the FBD need only be done at the lowest level of the SBS, which in most cases is at component level. It is from this relation between the FBD and the SBS that the combined result is termed a *functional systems breakdown structure (FSBS)*.

Some further concepts essential to a proper basic understanding of FSBS are considered in the following definitions:

A *system* is defined as "*a complete whole of a set of connected parts or components with functionally related properties that links them together in a system process*".

A *function* is defined as "*the work that an item is designed to perform*".

This definition indicates, through the terms *work* and *design*, that any item contains both *operational* and *physical* functions. *Operational functions* are related to the item's *working* performance, and *physical functions* are related to the item's *design*.

Functional relationships, on the other hand, describe the *actions* or *changes* in a system that are derived from the various ways in which the system's *components* and their properties are linked together *within* the system. Functional relationships

thus describe the *complexity* of a system at the component level. *Component functional relationships* describe the *actions* internal in a system, and can be regarded as the *interactive work* that the system's components are designed to perform. Component functional relationships may therefore be considered from the point of view of their internal *interactive functions.* Furthermore, *component functional relationships* may also be considered from the point of view of their different *cause and effect changes*, or change *symptoms*, or in other words, their internal *symptomatic functions*.

In order to fully understand *component functional relationships*, concise descriptions of the *operational* and *physical functions* of the system must first be defined, and then the *functional relationships* at *component* level are defined. The descriptions of the system's *operational* and *physical functions* need to be quantified with respect to their *limits of performance*, so that the *severity* of functional failures can be defined at a later stage in the FMECA procedure. The first step, then, is to list the components in a functional systems breakdown structure (FSBS).

b) Functional Systems Breakdown Structure (FSBS)

The identification of the constituent items of each level of a functional systems breakdown structure (FSBS) is determined from the top down. This is done by identifying the actual physical design configuration of the system, in lower-level items of the systems hierarchy. The various levels of an FSBS are identified from the bottom up, by logically grouping items or components into sub-assemblies, assemblies or sub-systems. Operational and physical functions and limits of performance are then defined in the FSBS. Once the functions in the FSBS have been described and limits of performance quantified, then the various *functional relationships* of the components are defined, either in a *functional block diagram* (*FBD*) or through *functional modelling*.

The functional block diagram (FBD) is a structured means for describing *component functional relationships* for design analysis. However, in the development of an FBD, the descriptions of these component functional relationships should be limited to two words if possible: a verb to describe the *action* or *change*, and a noun to describe the *object* of the action or change. In most cases, if the component functional relationships cannot be stated using two words, then *more than one functional relationship exists.*

A verb–noun combination cannot be repeated in any one branch of the FBD's descriptions of the component functional relationships. If, however, repetition is apparent, then review of the component functional relationships in the functional block diagram (FBD) becomes necessary (Blanchard et al. 1990).

As an example, some verb–noun combinations are given for describing component functional relationships for design analysis during the *preliminary design* phase in the engineering design process.

The following semantic list represents some verb–noun combinations:

Verb	*Noun*
Circulate	Current
Close	Overflow
Compress	Gas
Confine	Liquids
Contain	Lubricant
Control	Flow
Divert	Fluid
Generate	Power
Provide	Seal
Transfer	Signal
Transport	Material

It is obvious that the most appropriate verb must be combined with a corresponding noun. Thus, the verb 'control' can be used in many combinations with different nouns. It can be readily discerned that these actions can be either *operational functional relationships* that are related to the item's required *performance*, or *physical functional relationships* that are related to the item's *design*. For instance, current can be controlled *operationally*, through the use of a regulator, or *physically* through the internal physical *resistance* properties of a conductor.

What becomes essential is to ask the question '*how?*' after the verb–noun combination has been established in describing *functional relationships*. The question is directed towards an answer of either '*operational*' or '*physical*'. In the case of an uncertain decision concerning whether the verb–noun description of the *functional relationship* is achieved either *operationally* (i.e. related to the item's performance) or *physically* (i.e. related to the item's material design), then the basic principles used in defining the item's *functions* can be referred to.

These principles indicate that the item's *functions* can be identified on the basis of the fundamental criteria relating to operational and physical functions, which are:

- *movement* and *work*, in the case of operational functions, and
- *shape* and *consistence*, in the case of physical functions.

c) Failure Modes and Effects Analysis (FMEA)

Failure modes and effects analysis (FMEA) is one of the most commonly used techniques for assessing the reliability of engineering designs. The analysis at *systems* level involves identifying potential *equipment failure modes* and assessing the *consequences* they might have on the system's performance. Analysis at *equipment* level involves identifying potential *component failure modes* and assessing the *effects* they might have on the functional reliability of neighbouring components, and then propagating these up to the system level. This propagation is usually done in a failure modes and effects criticality analysis (FMEA).

The criticality of components and component failure modes can therefore be assessed by the extent the effects of failure might have on equipment functional

reliability, and the appropriate steps taken to amend the design so that critical failure modes become sufficiently improbable.

With the completion of the functional block diagram (FBD), development of the failure modes and effects analysis (FMEA) can proceed. The initial steps of FMEA considers criteria such as:

- System performance specifications
- Component functional relationships
- Failure modes
- Failure effects
- Failure causes.

A complex system can be analysed at different levels of resolution and the appropriate performance or functions defined at each level. The top levels of the *system breakdown structure* are the process and system levels where *performance* specifications are defined, and the lower levels are the assembly, component and part levels where not only primary equipment but also individual components have a role to play in the overall *functions* of the system. An FMEA consists of a combined top-down and bottom-up analysis. From the top, the process and system *performance* specifications are decomposed into assembly and component performance requirements and, from the bottom, these assembly and component performance requirements are translated into *functions* and *functional relationships* for which system performance specifications can be met.

After determining assembly and component functions and functional relationships through application of the techniques of system breakdown structures (SBS) and functional block diagrams (FBD), the remaining steps in developing an FMEA consider determining *failure modes*, *failure effects*, *failure causes* as well as *failure detection*.

Engineering systems are designed to achieve predefined performance criteria and, although the FMEA will provide a comparison between a system's normal and faulty behaviour through the identification of failure modes and related descriptions of possible failures, it is only when this behavioural change affects one of the performance criteria that a failure effect is deemed to have occurred. The failure effect is then described in terms of system performance that has been either reduced or not achieved at all.

A survey of applied FMEA has shown that the greatest criticism is the inability of the FMEA to sufficiently influence the engineering design process, because the timescale of the analysis often exceeds the design process (Bull et al. 1995b). It is therefore often the case that FMEA is seen not as a design tool but solely as a deliverable to the client. To reduce the total time for the FMEA, an approach is required whereby the methodology is not only automated but also integrated into the engineering design process through intelligent computer automated methodology. Such an approach would, however, require consideration of qualitative reasoning in engineering design analysis. In order to be able to develop the reliability technique of FMEA (and its extension of criticality considerations into a FMECA) for application in intelligent computer automated methodology, particularly for artificial

intelligence-based (AIB) modelling, it is essential to carefully consider each progressive step with respect to its related definitions. It is obvious that the best point of departure would be an appropriate definition for *failure*.

According to the US Military Standard (MIL-STD-721B), a *failure* is defined as "*the inability of an item to function within its specified limits of performance*". This implies that *system functional performance limits* must be clearly defined before any functional failures can be identified. The task of defining system functional performance limits is not straightforward, especially with complex integration of systems. A thorough analysis of systems integration complexity requires that the FMEA not only considers the functions of the various systems and their equipment but that limits of performance be related to these functions as well.

As previously indicated, the definition of a *function* is given as "*the work that an item is designed to perform*". Thus, failure of the item's function means failure of the work that the item is designed to perform.

Functional failure can thus be defined as "*the inability of an item to carry-out the work that it is designed to perform within specified limits of performance*".

It is obvious from this definition that there are two degrees of severity of functional failure:

i) A *complete loss of function*, where the item cannot carry out any of the work that it was designed to perform.
ii) *A partial loss of function*, where the item is unable to function within specified limits of performance.

Potential failure may be defined as "*the identifiable condition of an item indicating that functional failure can be expected*". In other words, potential failure is an *identifiable condition* or *state* of an item on which its function depends, indicating that the occurrence of functional failure can be expected.

From an essential understanding of the implications of these definitions, the various steps in the development of an FMEA can now be considered.

STEP 1: the first criterion to consider in the FMEA is *failure mode*.

The definition of *mode* is given as "*method or manner*".
Failure mode can be defined as "*the method or manner of failure*".
If *failure* is considered from the viewpoint of either *functional failure or potential failure*, then *failure mode* can be determined as:

i) The *method or manner* in which an item is *unable* to carry out the *work* that it is designed to perform within limits of performance. This would imply either the mode of failure in which the item cannot carry out any of the work that it is designed to perform (i.e. *complete loss of function*), or the mode of failure in which the item is unable to function within specified limits of performance (i.e. *partial loss of function*).
ii) The *method or manner* in which an item's identifiable *condition* could arise, indicating that functional failure can be expected. This would imply a failure mode only when the item's identifiable condition is such that a functional failure can be expected.

Thus, *failure mode* can be described from the points of view of:

- A complete functional loss.
- A partial functional loss.
- An identifiable condition.

For reliability assessment during the preliminary engineering design phase, the first two failure modes, namely a complete functional loss, and a partial functional loss, can be practically considered. The determination of an identifiable condition is considered when contemplating the possible *causes* of a complete functional loss or of a partial functional loss.

STEP 2: the following step in developing an FMEA is to consider the criteria of *failure effects*.

The definition of *effect* is given as "*an immediate result produced*".
Failure effects can be defined as "*the immediate results produced by failure*".
Failure consequence can be defined as "*the overall result or outcome of failures*".
It is clear that from these definitions that there are two levels—firstly, an *immediate effect* and, secondly, an *overall consequence* of failure.

i) The *effects of failure* are associated with analysis at *component* level of the immediate results that initially occur within the component's or assembly's environment.
ii) The *consequences of failure* are associated with analysis at *systems* level of the overall results that eventually occur in the system or process as a whole.

For the purpose of developing an FMEA at the higher systems level, some of the basic principles of *failure consequences* need to be described. The *consequences* of failure need *not* have *immediate* results. However, as indicated before, typical FMEA analysis of failure effects on functional reliability at component level and propagated up to the system level is usually done in a failure modes and effects criticality analysis (FMEA).

Operational and *physical consequences of failure* can be grouped into five significant categories:

- Safety consequences.
  Safety operational and physical consequences of functional failure are alternately termed *critical functional failure consequences.* These functional failures affect either the operational or physical functions of systems, assemblies or components that could have a direct adverse effect on *safety*, with respect to catastrophic incidents or accidents.
- Economic consequences.
  Economic operational and physical consequences of functional failure involve an *indirect economic loss*, such as the loss in production, as well as the *direct cost of corrective action*.
- Environmental consequences.
  Environmental operational and physical consequences of functional failure in engineered installations relate to environmental problems predominantly associ-

ated with treatment of wastes from mineral processing operations, hydrometallurgical processes, high-temperature processes, and processing operations from which by-products are treated. Any *functional failures* in these processes would most likely result in *environmental operational and physical consequences*.

- Maintenance consequences.
  Maintenance operational and physical consequences of functional failure involve only the *direct cost of corrective maintenance action.*
- Systems consequences.
  Systems operational and physical consequences of functional failure involve integrated failures in the functional relationships of components in process engineering systems with regard to their internal *interactive functions*, or internal *symptomatic functions*.

STEP 3: the following step in developing an FMEA is to consider the criteria of *failure causes.*

The definition of *cause* is "*that which produces an effect*".
Failure causes can be defined as "*the initiation of failures which produce an effect*".

The definition of *functional failure* was given as "*the inability of an item to carry-out the work that it is designed to perform within specified limits of performance*". Considering the *causes* of *functional failure*, it is practical to place these into *hazard categories* of component functional failure *incidents or events*. These *hazard categories* are determined through the reliability evaluation technique of *failure hazard analysis (FHA)*, which is considered later.

The definition of potential failure was given as "*the identifiable condition of an item indicating that functional failure can be expected*". The *effects of potential failure* could result in *functional failure*. In other words, the *causes of functional failure* can be found in *potential failure conditions*. The most significant aspect of *potential failure* is that it is a condition or state, and not an incident or event such as with *functional failure*.

In being able to define *potential failure* in an item of equipment, the *identifiable conditions or state* of the item upon which its functions depend must then also be identified. The *operational* and *physical* conditions of the item form the basis for defining potential failures arising in the item's functions. This implies that an item, which may have several functions and is meant to carry out work that it is designed to perform, will be subject to several conditions or states on which its functions depend, *from the moment that it is working or put to use.* In other words, the item is subject to *potential failure* the moment it is in *use*.

Potential failure is related to the identifiable condition or state of the item, based upon the work it is designed to perform, and the result of its use. The *causes* of potential failure are thus related to the *extent of use* under which the system or equipment is placed.

In summary, then, developing an FMEA includes considering the criteria of failure causes—the *causes of functional failure* can be found in potential failure condi-

tions and, in turn, the *causes of potential failure* can be related to the extent of use of the system or equipment.

Despite the fairly comprehensive and sound theoretical approach to the definitions of the relevant criteria and analysis steps in developing an FMEA, it still does not provide exhaustive lists of causes and effects for full sets of failure modes. A complete analysis, down to the smallest detail, is generally too expensive (and often impossible). The central objective of FMEA in engineering design therefore is more for *design verification*. This would require an approach to FMEA that concentrates on failure modes that can be represented in terms of simple linguistic or logic statements, or by *algorithmic modelling* in the case of more complicated failure modes. In the design of integrated engineering systems, however, most failure modes are not simple but complex, requiring an analytic approach such as algorithmic modelling.

3.3.2.2 Algorithmic Modelling in Failure Modes and Effects Analysis

All engineering systems can be broken down into sub-systems and/or assemblies and components, but at which level should they be modelled? At one extreme, if the FMEA is concerned with the process as a whole, it may be sufficient to represent the inherent equipment as single entities. Conversely, it may be necessary to consider the effects of failure of single components of the equipment. Less detailed analysis could be justified for a system based on previous designs, with relatively high reliability and safety records. Alternatively, greater detail and a correspondingly lower system-level analysis is required for a new design or a system with unknown reliability history (Wirth et al. 1996).

The British Standard on FMEA and FMECA (BS5760, 1991) requires failure modes to be considered at the lowest practical level. However, in considering the use of FMEA for *automated continual design reviews* in the engineering design process, it is prudent to initially concentrate on failure modes that could be represented in terms of simple linguistic or logic statements. Once this has been accomplished, the problem of how to address complicated failure modes can be addressed. This is considered in the following algorithmic approaches (Bull et al. 1995b):

- Numerical analysis
- Order of magnitude
- Qualitative simulation
- Fuzzy techniques.

a) Numerical Analysis

There are several numerical and symbolic algorithms that can be used to solve dynamic systems. However, many of these algorithms have two major drawbacks: firstly, they might not be able to reach a reliable steady-state solution, due to convolutions in the numerical solution of their differential equations, or because of the

presence of non-linear properties (for example, in the modelling of performance characteristics of relief valves, non-return valves, end stops, etc.).

Secondly, the solutions may be very specific. They are typically produced for a system at a certain pressure, flow, load condition, etc. In engineering design, and in particular in the FMEA, it is common not to know the precise values of quantities, especially in the early design stages. It would thus be more intuitive to be able to relate design criteria in terms of *ranges* of values, as considered in the *labelled interval calculus* method for system performance measures.

b) Order of Magnitude

The problem of how to address complicated failure modes can be approached through order of magnitude reasoning, developed by Raiman (1986) and extended by Mavrovouniotis and Stephanopoulis (Mavrovouniotis et al. 1988). Order of magnitude is primarily concerned with considering the relative sizes of quantities. A variable in this formalism refers to a specific physical quantity with known dimensions but unknown numerical values. The fundamental concept is that of a *link*—the ratio of two quantities, only one of which can be a *landmark*. Such a landmark is a variable with known (and constant) sign and value. There are seven possible primitive relations between these two quantities:

$A << B$	A is much smaller than B
$A - < B$	A is moderately smaller than B
$A \sim < B$	A is slightly smaller than B
$A == B$	A is exactly equal to B
$A > \sim B$	A is slightly larger than B
$A > - B$	A is moderately larger than B
$A >> B$	A is much larger than B.

The formalism itself involves representing these primitives as real intervals centred around unity (which represents exact equality). They allow the data to be represented either in terms of a precise value or in terms of intervals, depending upon the information available and the problem to be solved. Hence, the algorithmic model will encapsulate all the known features of the system being simulated. Vagueness is introduced only by lack of knowledge in the initial conditions. A typical analysis will consist of asking questions of the form:

- What happens if the pressure rises significantly higher than the operating pressure?
- What is the effect of the flow significantly being reduced?

c) Qualitative Simulation

Qualitative methods have been devised to simulate physical systems whereby quantities are represented by their sign only, and differential equations are reinterpreted

as *logical predicates*. The simulation involves finding values that satisfy these constraints (de Kleer et al. 1984).

This work was further developed to represent the quantities by *intervals* and *landmark* values (Kuipers 1986). Collectively, variables and landmarks are described as the quantities of the system. The latter represent important values of the quantities such as maximum pressure, temperature, flow, etc.

The major drawback with these methods is that the vagueness of the input data leads to ambiguities in the predictions of system behaviour, whereby many new constraints can be chosen that correspond to many physical solutions. In general, it is not possible to deduce which of the myriad of solutions is correct. In terms of FMEA, this would mean there could be a risk of failure effects being generated that are a result of the inadequacy of the algorithm, and not of a particular failure mode.

d) Fuzzy Techniques

Kuiper's work was enhanced by Shen and Leitch (Shen et al. 1993) to allow for *fuzzy intervals* to be used in *fuzzy simulation*.

In qualitative simulation, it is possible to describe quantities (such as pressure) as 'low' or 'high'. However, typical of engineering systems, these fuzzy intervals may be divided by a landmark representing some critical quantity, with consequent uncertainty where the resulting point should lie, as 'low' and 'high' are not absolute terms.

The concept of *fuzzification* allows the boundary to be blurred, so that for a small range of values, the quantity could be described as *both* 'low' and 'medium'. The problem with this approach (and with fuzzy simulation algorithms in general) is that it introduces further ambiguity.

For example, it has been found that in the dynamic simulation of an actuator, there are 19 possible values for the solution after only three steps (Bull et al. 1995b). This result is even worse than it appears, as the process of fuzzification removes the guarantee of converging on a physical solution. Furthermore, it has been shown that it is possible to develop fuzzy Euler integration that allows for qualitative states to be predicted at absolute time points. This solves some of the problems but there is still ambiguity in predicted behaviour of the system (Steele et al. 1996, 1997; Coghill et al. 1999a,b).

3.3.2.3 Qualitative Reasoning in Failure Modes and Effects Analysis

It would initially appear that qualitative reasoning algorithms are not suitable for FMEA or FMECA, as this formalism of analysis requires unique predictions of system behaviour. Although some vagueness is permissible due to *uncertainty*, it cannot be ambiguous, and ambiguity is an inherent feature of computational qualitative reasoning. In order, then, to consider the feasibility of qualitative reasoning in FMEA and FMECA without this resulting in ambiguity, it is essential to investigate further the concept of uncertainty in engineering design analysis.

a) The Concept of Uncertainty in Engineering Design Analysis

Introducing the concept of uncertainty in reliability assessment by utilising the techniques of FMEA and FMECA requires that some issues and concepts relating to the physical system being designed must first be considered.

A typical engineering design can be defined using the concepts introduced by Simon (1981), in terms of its inner and outer environment, whereby an interface between the substance and organisation of the design itself, and the surroundings in which it operates is defined. The design engineer's task is to establish a complete definition of the design and, in many cases, the manufacturing details (i.e. the inner environment) that can cope with supply and delivery (i.e. the outer environment) in order to satisfy a predetermined set of design criteria. Many of the issues that are often referred to as uncertainty are related to the ability of the design to meet the design criteria, and are due to characteristics associated with *both* the inner and outer environments (Batill et al. 2000). This is especially the case when several systems are integrated in a complex process with multiple (often conflicting) characteristics.

Engineering design is associated with decisions based upon information related to this interface, which considers uncertainty in the complex integration of systems in reality, compared to the concept of uncertainty in systems analysis and modelling. From the perspective of the designer, a primary concern is the source of variations in the inner environment, and the need to reduce these variations in system performance through decisions made in the design process. The designer is also concerned with how to reduce the sensitivity of the system's performance to variations in the outer environment (Simon 1981). Furthermore, from the designer's perspective, the system being designed exists only as an abstraction, and any information related to the system's characteristics or behaviour is approximate prior to its physical realisation. Dealing with this incomplete description of the system, and the approximate nature of the information associated with its characteristics and behaviour are key issues in the design process (Batill et al. 2000).

The intention, however, is to focus on the *integrity* of engineering design using the extensive capabilities now available with modelling and digital computing. With the selection of a basic concept of the system at the beginning of the conceptual phase of the engineering design process, the next step is to identify (though not necessarily quantify) a finite set of *design variables* that will eventually be used to uniquely specify the design. The identification and quantification of this set of design variables are central to, and will evolve with the design throughout the design process. It is this quantitative description of the system, based upon information developed, using algorithmic models or simulation, that becomes the focus of *preliminary* or *schematic design*.

Though there is great benefit in providing quantitative descriptions as early in the design process as possible, this depends upon the availability of knowledge, and the level of analysis and modelling techniques related to the design. As the level of abstraction of the design changes, and more and more detail is required to define it, the number of design variables will grow considerably. Design variables typically are associated with the type of material used and the geometric description of the

system(s) being designed. Eventually, during the *detail design* phase of the engineering design process, the designer will be required to specify (i.e. quantify) the design variables representing the system. This specification often takes the form of detailed engineering drawings that include materials information and all the necessary geometric information needed for fabrication, including manufacturing tolerances.

Decisions associated with quantifying (or selecting) the design variables are usually based upon an assessment of a set of behavioural variables, also referred to as *system states*. The behavioural variables or system states are used to describe the system's characteristics. The list of these characteristics also increases in detail as the level of abstraction of the system decreases.

The behavioural variables are used to assess the suitability of the design, and are based upon information obtained from several primary sources during the design process:

- Archived experience
- Engineering analysis (such as FMEA and FMECA)
- Modelling and simulation.

Interpolating or extrapolating from information on similar design concepts can provide the designer with sufficient confidence to make a decision based upon the success of earlier, similar designs. Often, this type of information is incorporated into heuristics (rules-of-thumb), design handbooks or design guidelines. Engineers commonly gather experiential information from empirical data or knowledge bases. The use of empirical information requires the designer to make numerous assumptions concerning the suitability of the available information and its applicability to the current situation. There are also many decisions made in the design process that are based upon individual or corporate experience that is not formally archived in a database.

This type of information is very valuable in the design of systems that are perturbations (evolutionary designs) of existing successful designs, but has severe limitations when considering the design of new or revolutionary designs. Though it may be useful information, in a way that will assist in assessing the risk associated with the entire design—which is usually not possible, it tends to compound the problem related to the concept of *uncertainty* in the engineering design process.

The second type of information available to the designer is based upon analysis, mathematical modelling and simulation. As engineering systems become more complex, and greater demands are placed upon their performance and cost, this source of information becomes even more important in the design process. However, the information provided by analysis such as FMEA and FMECA carries with it a significant level of uncertainty, and the use of such information introduces an equal level of risk to the decisions made, which will affect the integrity of the design. Quantifying uncertainty, and understanding the significant impact it has in the design process, is an important issue that requires specific consideration, especially with respect to the increasing complexity of engineering designs.

A further extension to the reliability assessment technique of FMECA is therefore considered that includes the appropriate representation of *uncertainty* and

incompleteness of information in available knowledge. The main consideration of such an approach is to provide a qualitative treatment of uncertainty based on *possibility theory* and *fuzzy sets* (Zadeh 1965). This allows for the realisation of failure effects and overall consequences (manifestations) that will be more or less *certainly* present (or absent), and failure effects and consequences that could be more or less *possibly* present (or absent) when a particular failure mode is identified. This is achieved by means of qualitative *uncertainty calculus* in causal matrices, based on Zadeh's possibility measures (Zadeh 1979), and their dual measures of certainty (or necessity).

b) Uncertainty and Incompleteness in Available Knowledge

Available knowledge in engineering design analysis (specifically in the reliability assessment techniques of FMEA and FMECA) can be considered from the point of view of *behavioural knowledge* and of *functional knowledge*. These two aspects are accordingly described:

i) In *behavioural knowledge*: expressing the *likelihood* of some or other expected consequences as a result of an identified failure mode. Information about likelihood is generally qualitative, rather than quantitative. Included is the concept of 'negative information', stating that some consequences cannot manifest, or are almost impossible as consequences of a hypothesised failure mode. Moreover, due to incompleteness of the knowledge, distinction is made between consequences that are more or less sure, and those that are only possible.
ii) In *functional knowledge*: expressing the functional activities or work that systems and equipment are designed to perform. In a similar way as in the behavioural knowledge, the propagation of system and equipment functions are also incomplete and uncertain. In order to effectively capture uncertainty, a qualitative approach is more appropriate to the available information than a quantitative one.

In the following paragraphs, an overview is given of various concepts and theory for qualitatively modelling uncertainty in engineering design.

3.3.2.4 Overview of Fuziness in Engineering Design Analysis

In the real world there exists knowledge that is vague, uncertain, ambiguous or probabilistic in nature, termed *fuzzy knowledge*. Human thinking and reasoning frequently involves fuzzy knowledge originating from inexact concepts and similar, rather than identical experiences. In complex systems, it is very difficult to answer questions on system behaviour because they generally do not have exact answers. Qualitative reasoning in engineering design analysis attempts not only to give such answers but also to describe their reality level, calculated from the uncertainty and imprecision of facts that are applicable. The analysis should also be able to cope with

unreliable and incomplete information and with different expert opinions. Many commercial *expert system* tools or shells use different approaches to handle uncertainty in knowledge or data, such as certainty factors (Shortliffe 1976) and Bayesian models (Buchanan et al. 1984), but they cannot cope with fuzzy knowledge, which constitutes a very significant part of the use of natural language in design analysis, particularly in the early phases of the engineering design process.

Several computer automated systems support some fuzzy reasoning, such as FAULT (Whalen et al. 1982), FLOPS (Buckley et al. 1987), FLISP (Sosnowski 1990) and CLIPS (Orchard 1998), though most of these are developed from high-level languages intended for a specific application.

Fuzziness and Probability

Probability and fuzziness are related but different concepts. Fuzziness is a type of *deterministic uncertainty*. It describes the event class ambiguity. Fuzziness measures the degree to which an event occurs, not whether it does occur. Probability arises from the question whether or not an event occurs, and assumes that the event class is crisply defined and that the law of non-contradiction holds. However, it would seem more appropriate to investigate the *fuzziness of probability*, rather than dismiss probability as a special case of fuzziness. In essence, whenever the outcome of an event is difficult to compute, a probabilistic approach may be used to estimate the likelihood of all possible outcomes belonging to an event class. Fuzzy probability extends the traditional notion of probability when there are outcomes that belong to *several* event classes at the same time but at different degrees. Fuzziness and probability are orthogonal concepts that characterise different aspects of the same event (Bezdek 1993).

a) Fuzzy Set Theory

Fuzziness occurs when the boundary of an element of information is not clear-cut. For example, concepts such as *high*, *low*, *medium* or even *reliable* are fuzzy. As a simple example, there is no single quantitative value that defines the term *young*. For some people, age 25 is *young* and, for others, age 35 is *young*. In fact, the concept *young* has no precise boundary. Age 1 is definitely *young* and age 100 is definitely *not young*; however, age 35 has some possibility of being *young* and usually depends on the context in which it is being considered. The representation of this kind of inexact information is based on the concept of *fuzzy set theory* (Zadeh 1965). Fuzzy sets are a generalisation of conventional set theory that was introduced as a mathematical way to represent vagueness in everyday life. Unlike classical set theory, where one deals with objects of which the membership to a set can be clearly described, in fuzzy set theory membership of an element to a set can be partial, i.e. an element belongs to a set with a certain grade (possibility) of membership.

Fuzzy interpretations of data structures, particularly during the initial stages of engineering design, are a very natural and intuitively plausible way to formulate and solve various design problems. Conventional (crisp) sets contain objects that satisfy precise properties required for membership. For example, the set of numbers H from 6 to 8 is crisp and can be defined as:

$$H = \{r \in R | 6 \leq r \leq 8\}$$

Also, H is described by its *membership* (or characteristic) *function* (MF):

m_H: $R \rightarrow \{0,1\}$ defined as:

$$\begin{aligned} m_H(r) &= \{1 \quad 6 \leq r \leq 8\} \\ &= \{0 \quad \text{otherwise}\} \end{aligned}$$

Every real number r either is or is not in H. Since m_H maps all real numbers $r \in R$ onto the two points $(0,1)$, crisp sets correspond to two-valued logic: is or is not, on or off, black or white, 1 or 0, etc. In logic, values of m_H are called *truth values* with reference to the question:

'Is r in H?' The answer is yes if, and only if $m_H(r) = 1$; otherwise, no.

Consider the set F of real numbers that are *close* to 7. Since the property 'close to 7' is fuzzy, there is not a unique membership function for F. Rather, the decision must be made, based on the potential application and properties for F, what m_H should be. Properties that might seem plausible for F include:

i) *normality*
   (i.e. $\text{MF}(7) = 1$)
ii) *monotonicity*
   (the closer r is to 7, the closer $m_H(r)$ is to 1, and conversely)
iii) *symmetry*
   (numbers equally far left and right of 7 should have equal memberships).

Given these intuitive constraints, functions that usefully represent F are m_{F1}, *which is discrete* (represented by a staircase graph), or the function m_{F1}, *which is continuous but not smooth* (represented by a triangle graph).

One can easily construct a membership (or characteristic) function (MF) for F so that every number has some positive membership in F but numbers 'far from 7', such as 100, would not be expected to be included. One of the greatest differences between crisp and fuzzy sets is that the former always have unique MFs, whereas every fuzzy set may have an infinite number of MFs. This is both a weakness and a strength, in that uniqueness is sacrificed but with a gain in flexibility, enabling fuzzy models to be adjusted for maximum utility in a given situation.

In conventional set theory, sets of real objects, such as the numbers in H, are equivalent to, and isomorphically described by, a unique membership function such as m_H. However, there is no set theory with the equivalent of 'real objects' corresponding to m_F. Fuzzy sets are always functions, from a 'universe of objects', say X,

into $[0,1]$. The fuzzy set is the function m_F that carries X into $[0,1]$. Every function $m\colon X \to [0,1]$ is a fuzzy set by definition. While this is true in a formal mathematical sense, many functions that qualify on this ground cannot be suitably interpreted as realisations of a conceptual fuzzy set. In other words, functions that map X into the unit interval may be fuzzy sets, but become fuzzy sets when, and only when, they match some intuitively plausible semantic description of imprecise properties of the objects in X (Bezdek 1993).

b) Formulation of Fuzzy Set Theory

Let X be a space of objects and x be a generic element of X. A classical set A, $A \subseteq X$, is defined as a collection of elements or objects $x \in X$, such that each element (x) can either belong to the set A, or not. By defining a membership (or characteristic) function for each element x in X, a classical set A can be represented by a set of ordered pairs $(x,0)$, $(x,1)$, which indicates $x \notin A$ or $x \in A$ respectively (Jang et al. 1997).

Unlike conventional sets, a fuzzy set expresses *the degree to which an element belongs to a set*. Hence, the membership function of a fuzzy set is allowed to have values between 0 and 1, which denote the degree of membership of an element in the given set. Obviously, the definition of a fuzzy set is a simple extension of the definition of a classical (crisp) set in which the characteristic function is permitted to have any values between 0 and 1. If the value of the membership function is restricted to either 0 or 1, then A is reduced to a classical set. For clarity, classical sets are referred to as ordinary sets, crisp sets, non-fuzzy sets, or just sets.

Usually, X is referred to as the *universe of discourse* or, simply, the *universe*, and it may consist of discrete (ordered or non-ordered) objects or it can be a continuous space. The construction of a fuzzy set depends on two requirements: the identification of a suitable *universe of discourse*, and the specification of an appropriate *membership function*. In practice, when the universe of discourse X is a continuous space, it is partitioned into several fuzzy sets with MFs covering X in a more or less uniform manner. These fuzzy sets, which usually carry names that conform to adjectives appearing in daily linguistic usage, such as ‘large’, ‘medium’ or ‘small’, are called linguistic values or linguistic labels. Thus, the universe of discourse X is often called the *linguistic variable*.

The specification of membership functions is subjective, which means that the membership functions specified for the same concept by different persons may vary considerably. This subjectivity comes from individual differences in perceiving or expressing abstract concepts, and has little to do with randomness. Therefore, the subjectivity and non-randomness of fuzzy sets is the primary difference between the study of fuzzy sets, and probability theory that deals with an objective view of random phenomena.

Fuzzy Sets and Membership Functions

If X is a collection of objects denoted generically by x, then a fuzzy set A in X is defined as a set of ordered pairs $A = \{(x, \mu A(x)) | x \in X\}$, where $\mu A(x)$ is called the *membership function* (or MF, for short) for the fuzzy set A. The MF maps each element of X to a membership grade or membership value between 0 and 1 (included).

More formally, a fuzzy set A in a universe of discourse U is characterised by the *membership function*

$$\mu_A : U \rightarrow [0,1] \tag{3.93}$$

The function associates, with each element x of U, a number $\mu_A(x)$ in the interval $[0,1]$. This represents the grade of membership of x in the fuzzy set A. For example, the fuzzy term *young* might be defined by the fuzzy set given in Table 3.12 (Orchard 1998).

Regarding Eq. (3.93), one can write:

$$\mu_{\text{young}}(25) = 1, \mu_{\text{young}}(30) = 0.8, \ldots, \mu_{\text{young}}(50) = 0$$

Grade of membership values constitute a *possibility distribution* of the term *young*. The table can be graphically represented as in Fig. 3.27.

The *possibility distribution* of a fuzzy concept like *somewhat young* or *very young* can be obtained by applying arithmetic operations to the fuzzy set of the basic fuzzy term *young*, where the *modifiers* '*somewhat*' and '*very*' are associated with specific mathematical functions.

For example, the possibility values of each age in the fuzzy set representing the fuzzy concept *somewhat young* might be calculated by taking the square root of the corresponding possibility values in the fuzzy set of *young*, as illustrated in Fig. 3.28. These modifiers are commonly referred to as *hedges*.

A *modifier* may be used to further enhance the ability to describe fuzzy concepts. Modifiers (very, slightly, etc.) used in phrases such as *very* hot or *slightly* cold change (modify) the shape of a fuzzy set in a way that suits the meaning of the word used. A typical set of predefined modifiers (Orchard 1998) that can be used to describe fuzzy concepts in fuzzy terms, fuzzy rule patterns or fuzzy facts is given in Table 3.13.

Table 3.12 Fuzzy term *young*

Age	Grade of membership
25	1.0
30	0.8
35	0.6
40	0.4
45	0.2
50	0.0

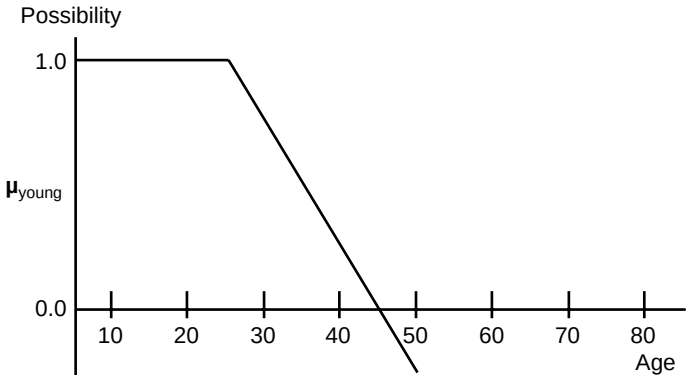


Fig. 3.27 Possibility distribution of *young*

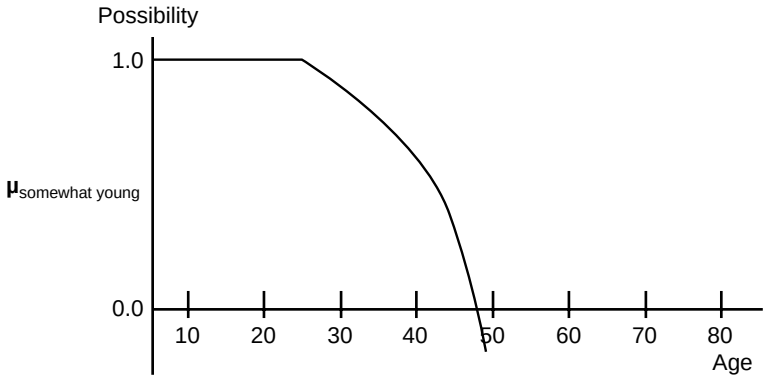


Fig. 3.28 Possibility distribution of *somewhat young*

Table 3.13 Modifiers (hedges) and linguistic expressions

Modifier name	Modifier description
Not	$1-y$
Very	$y^{**}2$
Somewhat	$y^{**}0.333$
More-or-less	$y^{**}0.5$
Extremely	$y^{**}3$
Intensify	$(y^{**}2)$ if y in $[0,0.5]$ $1-2(1-y)^{**}2$ if y in $(0.5,1]$
Plus	$y^{**}1.25$
Norm	Normalises the fuzzy set so that the maximum value of the set is scaled 1.0 ($y = y^{*}1.0/\text{max-value}$)
Slightly intensify	(norm (plus A AND not very A)) = norm ($y^{**}1.25$ AND $1-y^{**}2$)

These modifiers change the shape of a fuzzy set using mathematical operations on each point of the set. In the above table, the variable *y* represents each membership value in the fuzzy set, and *A* represents the entire fuzzy set (i.e. the term *very A* applies the *very* modifier to the entire set where the modifier description $y^{**}2$ squares each membership value). When a modifier is used in descriptive expressions, it can be used in upper or lower case (i.e. NOT or not).

c) Uncertainty

Uncertainty occurs when one is *not* absolutely sure about an element of information. The degree of uncertainty is usually represented by a *crisp* numerical value on a scale from 0 to 1, where a *certainty factor* of 1 indicates that the assessment of a particular fact is very *certain* that the fact is true, and a certainty factor of 0 indicates that the assessment is very *uncertain* that the fact is true. A fact is composed of two parts: the statement of the fact in non-fuzzy reasoning, and its certainty factor. Only facts have associated certainty factors. In general, a factual statement takes the following form:

(fact) {CF certainty factor}

The CF acts as the delimiter between the fact and the numerical certainty factor, and the brackets { } indicate an optional part of the statement. For example, (pressure high) {CF 0.8} is a fact that indicates a particular system attribute of pressure will be high with a certainty of 0.8. However, if the certainty factor is omitted, as in a non-fuzzy fact, (pressure high), then the assumption is that the pressure will be high with a certainty of 1 (or 100%). The term *high* in itself is fuzzy and relates to a fuzzy set. The fuzzy term *high* also has a certainty qualification through its certainty factor. Thus, uncertainty and fuzziness can occur simultaneously.

d) Fuzzy Inference

Expression of fuzzy knowledge is primarily through the use of *fuzzy rules*. However, there is no unique type of fuzzy knowledge, nor is there only one kind of fuzzy rule. It is pointed out that the interpretation of a fuzzy rule dictates the way the fuzzy rule should be combined in the framework of fuzzy sets and possibility theory (Dubois et al. 1994).

The various kinds of fuzzy rules that can be considered (certainty rules, gradual rules, possibility rules, etc.) have different *fuzzy inference* behaviours, and correspond to various applications. Rule evaluation depends on a number of different factors, such as whether or not fuzzy variables are found in the antecedent or consequent part of a rule, whether a rule contains multiple antecedents or consequents, or whether a fuzzy fact being asserted has the same fuzzy variable as an already existing fuzzy fact (global contribution). The representation of fuzzy knowledge through fuzzy inference needs to be briefly investigated for inclusion in engineering design analysis.

e) Simple Fuzzy Rules

Algorithms for evaluating certainty factors (CF) and simple fuzzy rules are first considered, such as the simple rule of form:

if A then C	$\mathrm{CF_r}$
A'	$\mathrm{CF_f}$
C'	$\mathrm{CF_c}$

where
A is the antecedent of the rule
A' is the matching fact in the fact database
C is the consequent of the rule
C' is the actual consequent calculated
$\mathrm{CF_r}$ is the certainty factor of the rule
$\mathrm{CF_f}$ is the certainty factor of the fact
$\mathrm{CF_c}$ is the certainty factor of the conclusion

Three types of simple rules are defined:
CRISP_;
FUZZY_CRISP; and
FUZZY_FUZZY.

If the antecedent of the rule does not contain a fuzzy object, then the type of rule is CRISP_ regardless of whether or not a consequent contains a fuzzy fact. If only the antecedent contains a fuzzy fact, then the type of rule is FUZZY_CRISP. If both antecedent and consequent contain fuzzy facts, then the type of rule is FUZZY_FUZZY.

CRISP_ simple rule If the type of rule is CRISP_, then A' must be equal to A in order for this rule to validate (or *fire* in computer algorithms). This is a non-fuzzy rule (actually, A would be a pattern, and A' would match the pattern specification but, for simplicity, patterns are not dealt with here). In this case, the conclusion C' is equal to C, and

$$\mathrm{CF_c} = \mathrm{CF_r} * \mathrm{CF_f} \,. \tag{3.94}$$

FUZZY_CRISP simple rule If the type of rule is FUZZY_CRISP, then A' must be a fuzzy fact with the same fuzzy variable as specified in A for a match. In addition, values of the fuzzy variables A and A', as represented by the fuzzy sets F_α and F'_α, do not have to be equal.

For a FUZZY_CRISP rule, the conclusion C' is equal to C, and

$$\mathrm{CF_c} = \mathrm{CF_r} * \mathrm{CF_f} * S \,. \tag{3.95}$$

S is a measure of similarity between the fuzzy sets F_α (determined by the fuzzy pattern A) and F'_α (of the matching fact A'). The measure of similarity S is based upon the measure of *possibility* P and the measure of *necessity* N. It is calculated

according to the following formula

$$S = P\left(F_\alpha | F'_\alpha\right) \quad \text{if } N\left(F_\alpha | F'_\alpha\right) > 0.5$$
$$S = \left(N\left(F_\alpha | F'_\alpha\right) + 0.5\right) * P\left(F_\alpha | F'_\alpha\right)$$

Otherwise where $\forall$ u $\in$ U:

$$P\left(F_\alpha | F'_\alpha\right) = \max\left(\min\left(\mu_{F_\alpha}(\mathrm{u})\ ,\ \mu_{F'_\alpha}(\mathrm{u})\right)\right) \tag{3.96}$$

[min is the minimum and max is the maximum, so that max $(\min(a,b))$ would represent the maximum of all the minimums between pairs a and b] (Cayrol et al. 1982), and

$$N\left(F_\alpha | F_{\alpha'}\right) = 1 - P\left(F'_\alpha | F_{\alpha'}\right) \tag{3.97}$$

F'_α is the complement of F_α described by the membership function

$$\forall (u \in U) \mu_{F'_\alpha}(u) = 1 - \mu_{F_\alpha}(u)\ . \tag{3.98}$$

Therefore, if the similarity between the fuzzy sets associated with the fuzzy pattern (A) and the matching fact (A') is high, the certainty factor of the conclusion is very close to $\mathrm{CF_r} * \mathrm{CF_f}$, since S will be close to 1. If the fuzzy sets are identical, then S will be 1 and the certainty factor of the conclusion will equal $\mathrm{CF_r} * \mathrm{CF_f}$. If the match is poor, then this is reflected in a lower certainty factor for the conclusion. Note also that if the fuzzy sets do not overlap, then the similarity measure would be zero and the certainty factor of the conclusion would be zero as well. In this case, the conclusion would not be asserted and the match considered to have failed, with the outcome that the rule is not to be considered (Orchard 1998).

FUZZY_FUZZY simple rule If the type of rule is FUZZY_FUZZY, and the fuzzy fact and antecedent fuzzy pattern match in the same manner as discussed for a FUZZY_CRISP rule, then it can be shown that the antecedent and consequent of such a rule are connected by the fuzzy relation (Zadeh 1973):

$$R = F_\alpha * F_c \tag{3.99}$$

where:

F_α = fuzzy set denoting the value of the fuzzy antecedent pattern
F_c = fuzzy set denoting the value of the fuzzy consequent

The *membership function* of the relation R is calculated according to the following formula

$$\mu R(u,v) = \min\left(\mu_{F_\alpha}(u)\ ,\ \mu_{F_\mathrm{c}}(v)\right)\ , \quad \forall (uv) \in U \times V \tag{3.100}$$

The calculation of the conclusion is based upon the compositional rule of inference, which can be described as follows (Zadeh 1975):

$$F_{\mathrm{c}}' = F_{\alpha}' \circ R \tag{3.101}$$

F_{c}' is a fuzzy set denoting the value of the fuzzy object of the consequent. The membership function of F_{c}' is calculated as follows (Chiueh 1992):

$$\mu_{F_{\mathrm{c}}'}(v) = \max_{u \in U} \left(\min \mu_{F_{\alpha'}}(u) \, , \, \mu_R(u,v) \right)$$

which may be simplified to

$$\mu_{F_{\mathrm{c}}'}(v) = \min(z, \mu_{F_{\mathrm{c}}}(v)) \tag{3.102}$$

where:

$$z = \max \left(\min \left(\mu_{F_{\alpha'}}(u) \, , \, \mu_{F_{\alpha}}(u) \right) \right)$$

The certainty factor of the conclusion is calculated according to the formula

$$\mathrm{CF_c} = \mathrm{CF_r} * \mathrm{CF_f} \tag{3.103}$$

f) Complex Fuzzy Rules

Complex fuzzy rules—multiple consequents and multiple antecedents—include multiple patterns that are treated as multiple rules with a single assertion in the consequent.

Multiple consequents The consequent part of a fuzzy rule may contain only multiple patterns, specifically (C_1, $C_2, \ldots, C_n$), which are treated as multiple rules with a single consequent. Thus, the following rule,

if Antecedents then C_1 *and* C_2 *and* ... *and* C_n

is equivalent to the following rules:

if Antecedents then C_1
if Antecedents then C_2
...
if Antecedents then C_n

Multiple Antecedents

From the above, it is clear that only the problem of multiple patterns in the antecedent with a single assertion in the consequent needs to be considered. If the consequent assertion is not a fuzzy fact, then no special treatment is needed, since the conclusion will be the crisp (non-fuzzy) fact. However, if the consequent assertion is a fuzzy fact, the fuzzy value is calculated using the following algorithm (Whalen et al. 1983).

If the logical term, *and*, is used:

if A_1 and A_2 then C	$\mathrm{CF_r}$
A_1'	$\mathrm{CF_{f1}}$
A_2'	$\mathrm{CF_{f2}}$
C'	$\mathrm{CF_c}$

A_1' and A_2' are facts (crisp or fuzzy), which match the antecedents A_1 and A_2 respectively.

In this case, the fuzzy set describing the value of the fuzzy assertion in the conclusion is calculated according to the formula

$$F_c' = F_{c1}' \cap F_{c2}' \tag{3.104}$$

where $\cap$ denotes the intersection of two fuzzy sets in which a membership function of a fuzzy set C, which is the intersection of fuzzy sets A and B, is defined by the following formula

$$\mu_C(x) = \min\left(\mu_A(x)\ ,\ \mu_B(x)\right)\ , \quad \text{for } x \in U \tag{3.105}$$

and:

F_{c1}' is the result of fuzzy inference for the fact A_1' and the simple rule:

if A_1 *then* C

F_{c2}' is the result of fuzzy inference for the fact A_2' and the simple rule:

if A_2 *then* C

g) Global Contribution

In non-fuzzy knowledge, a fact is asserted with specific values. If the fact already exists, then the approach would be as if the fact was not asserted (unless fact duplication is allowed). In such a crisp system, there is no need to reassess the facts in the system—once they exist, they exist (unless certainty factors are being used, when the certainty factors are modified to account for the new evidence). In a fuzzy system, however, refinement of a fuzzy fact may be possible. Thus, in the case where

a fuzzy fact is asserted, this fact is treated as contributing evidence towards the conclusion about the fuzzy variable (it contributes globally). If information about the fuzzy variable has already been asserted, then this new evidence (or information) about the fuzzy variable is combined with the existing information in the fuzzy fact. Thus, the concept of restrictions on fact duplication for fuzzy facts does not apply as it does for non-fuzzy facts. There are many readily identifiable methods of combining evidence. In this case, the new value of the fuzzy fact is calculated accordingly

$$F_{\mathrm{g}} = F_{\mathrm{f}} \cup F_{\mathrm{c}}' \tag{3.106}$$

where:

F_{g} is the new value of the fuzzy fact
F_{f} is the existing value of the fuzzy fact
F_{c}' is the value of the fuzzy fact to be asserted

where $\cup$ denotes the union of two fuzzy sets in which a membership function of a fuzzy set C, which is the union of fuzzy sets A and B, is defined by the following formula

$$\mu_C(x) = \max\left(\mu_A(x)\ ,\ \mu_B(x)\right) \qquad \text{for } x \in U \tag{3.107}$$

The *uncertainties* are also aggregated to form an *overall uncertainty*. Basically, two uncertainties are combined, using the following formula

$$\mathrm{CF_g} = \mathrm{maximum}\left(\mathrm{CF_f}, \mathrm{CF_c}\right) \tag{3.108}$$

where:

$\mathrm{CF_g}$ is the combined uncertainty
$\mathrm{CF_f}$ is the uncertainty of the existing fact
$\mathrm{CF_c}$ is the uncertainty of the asserted fact

3.3.2.5 Fuzzy Logic and Fuzzy Reasoning

The use of fuzzy logic and fuzzy reasoning methods are becoming more and more popular in intelligent information systems (Ryan et al. 1994; Yen et al. 1995), in knowledge formation processes within knowledge-based systems (Walden et al. 1995), in hyper-knowledge support systems (Carlsson et al. 1995a,b,c), and in active decision support systems (Brännback et al. 1997).

a) Linguistic Variables

As indicated in Sect. 3.3.2.4, the use of fuzzy sets provides a basis for the manipulation of vague and imprecise concepts. Fuzzy sets were introduced by Zadeh (1975) as a means of representing and manipulating imprecise data and, in particular, fuzzy

sets can be used to represent *linguistic variables*. A linguistic variable can be regarded either as a variable of which the value is a fuzzy number or as a variable of which the values are defined in linguistic terms, such as *failure modes*, *failure effects*, *failure consequences* and *failure causes* in FMEA and FMECA.

A *linguistic variable* is characterised by a quintuple

$$(x, T(x), U, G, M) \tag{3.109}$$

where:

x	is the name of the linguistic variable;
$T(x)$	is the term set of x, i.e. the set of names of linguistic values of x with each value being a fuzzy number defined on U;
G	is a syntactic rule for generating the names of values of x;
M	is a semantic rule for associating with each value its meaning.

Consider the example If pressure in a process design is interpreted as a linguistic variable, then its term set $T(\text{pressure})$ could be: $T = \{\text{very low, low, moderate, high, very high, more or less high, slightly high, } \ldots\}$ where each of the terms in $T(\text{pressure})$ is characterised by the fuzzy set in a universe of discourse $U = [0, 300]$ with a unit of measure that the variable *pressure* might have.

We might interpret:

low as 'a pressure below about 50 psi'
moderate as 'a pressure close to 120 psi'
high as 'a pressure close to 190 psi'
very high as 'a pressure above about 260 psi'

These terms can be characterised as *fuzzy sets* of which the membership functions are:

$$\text{low}\ (p) = \begin{cases} 1 & \text{if } p \le 50 \\ 1 - (p - 50)/70 & \text{if } 50 \le p \le 120 \\ 0 & \text{otherwise} \end{cases}$$

$$\text{moderate}\ (p) = \begin{cases} 1 - |p - 120|/140 & \text{if } 50 \le p \le 190 \\ 0 & \text{otherwise} \end{cases}$$

$$\text{high}\ (p) = \begin{cases} 1 - |p - 190|/140 & \text{if } 120 \le p \le 260 \\ 0 & \text{otherwise} \end{cases}$$

$$\text{very high}\ (p) = \begin{cases} 1 & \text{if } p \le 260 \\ 1 - (260 - p)/140 & \text{if } 190 \le p \le 260 \\ 0 & \text{otherwise} \end{cases}$$

The term set $T(\text{pressure})$ given by the above linguistic variables, $T(\text{pressure}) = \{\text{low}\ (p), \text{moderate}\ (p), \text{high}\ (p), \text{very high}\ (p)\}$, and the related fuzzy sets can be represented by the *mapping* illustrated in Fig. 3.29.

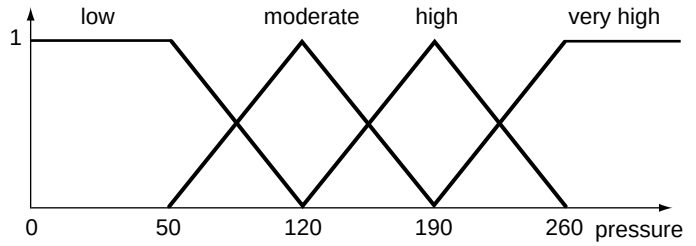


Fig. 3.29 Values of linguistic variable *pressure*

A *mapping* can be formulated as:

$$T\colon [0,1]\times[0,1]\to[0,1]$$

which is a triangular norm (t-norm for short) if it is symmetric, associative and non-decreasing in each argument, and $T(a,1)=a$, for all $a\in[0,1]$.

The mapping formulated by

$$S\colon [0,1]\times[0,1]\to[0,1]$$

is a triangular co-norm (t-conorm, for short) if it is symmetric, associative and non-decreasing in each argument, and $S(a,0)=a$, for all $a\in[0,1]$.

b) Translation Rules

Zadeh introduced a number of *translation rules* that allow for the representation of common *linguistic statements* in terms of propositions (or premises). These translation rules are expressed as (Zadeh 1979):

Main premise	x is A	x is an element of set A
Helping premise	x is B	x is an element of set B
Conclusion	x is $A\cap B$	x is an element of intersection A and B

Some of the translation rules include:

Entailment rule:

x is A	pressure is very low
$A\subset B$	very low $\subset$ low
x is B	pressure is low

Conjunction rule:

x is A	pressure is not very high
x is B	pressure is not very low
x is $A\cap B$	pressure is not very high *and* not very low

Disjunction rule:

$$\frac{x \text{ is } A \quad \text{or } x \text{ is } B}{x \text{ is } A \cup B} \qquad \frac{\text{pressure is not very high} \quad \text{or pressure is not very low}}{\text{pressure is not very high } \textit{or} \text{ not very low}}$$

Projection rule: $\dfrac{(x, y) \text{ have relation } R}{x \text{ is } \prod_X(R)}$ $\dfrac{(x, y) \text{ have relation } R}{y \text{ is } \prod_Y(R)}$

where: $\prod_X$ is a possibility measure defined on a finite propositional language and R is a particular rule-base (defined later).

Negation rule: $\dfrac{\text{not } (x \text{ is } A)}{x \text{ is } \neg A}$ $\dfrac{\text{not } (x \text{ is high})}{x \text{ is not high}}$

c) Fuzzy Logic

Prior to reviewing fuzzy logic, some consideration must first be given to *crisp logic*, especially on the concept of *implication*, in order to understand the comparable concept in fuzzy logic. Rules are a form of propositions. A *proposition* is an ordinary statement involving terms that have been defined, e.g. 'the failure rate is low'. Consequently, the following rule can be stated: 'IF the failure rate is low, THEN the equipment's reliability can be assumed to be high'.

In traditional *propositional logic*, a proposition must be meaningful to call it 'true' or 'false', whether or not we know which of these terms properly applies. *Logical reasoning* is the process of combining given propositions into other propositions, and repeating this step over and over again. Propositions can be combined in many ways, all of which are derived from several fundamental operations (Bezdek 1993):

- *conjunction* denoted $p \wedge q$ where we assert the simultaneous truth of two separate propositions p and q;
- *disjunction* denoted $p \vee q$ where we assert the truth of either or both of two separate propositions; and
- *implication* denoted $p \rightarrow q$, which takes the form of an IF–THEN rule. The IF part of an implication is called the *antecedent*, and the THEN part is called the *consequent*.
- *negation* denoted by $(\sim p)$ where a new proposition can be obtained from a given one by the clause 'it is false that ...'.
- *equivalence* denoted by $p \leftrightarrow q$, which means that p and q are *both* true or false.

In traditional propositional logic, unrelated propositions are combined into an implication, and *no cause or effect relation is assumed to exist*. This results in fundamental problems when traditional propositional logic is applied to engineering design analysis, such as in a diagnostic FMECA, where cause and effect are definite (i.e. causes and effects do occur).

In traditional propositional logic, an implication is said to be *true* if one of the following holds:

1) (antecedent is true, consequent is true),
2) (antecedent is false, consequent is false),
3) (antecedent is false, consequent is true).

The implication is said to be *false* when:

4) (antecedent is true, consequent is false).

Situation 1 is familiar from common experience. Situation 2 is also reasonable because, if we start from a false assumption, then we expect to reach a false conclusion. However, intuition is not always reliable. We may reason correctly from a false antecedent to a true consequent. Hence, a false antecedent can lead to a consequent that is either true or false, and thus both situations 2 and 3 are acceptable in traditional propositional logic. Finally, situation 4 is in accordance with intuition, for an implication is clearly false if a true antecedent leads to a false consequent.

A *logical structure* is constructed by applying the above four operations to propositions. The objective of a logical structure is to determine the truth or falsehood of all propositions that can be stated in the terminology of this structure. A *truth table* is very convenient for showing relationships between several propositions. The fundamental truth tables for *conjunction*, *disjunction*, *implication*, *equivalence* and *negation* are collected together in Table 3.14, in which symbol T means that the corresponding proposition is true, and symbol F means it is false. The fundamental axioms of traditional propositional logic are:

1) Every proposition is either true or false, but not both true and false.
2) The expressions given by defined terms are propositions.
3) Conjunction, disjunction, implication, equivalence and negation.

Using truth tables, many interpretations of the preceding translation rules can be derived.

A *tautology* is a proposition formed by combining other propositions, which is true regardless of the truth or falsehood of the forming propositions. The most important tautologies are

$$(p \to q) \leftrightarrow \sim[p \wedge (\sim q)] \leftrightarrow (\sim p) \vee q \tag{3.110}$$

These tautologies can be verified by substituting all the possible combinations for p and q and verifying how the equivalence always holds true. The importance of these tautologies is that they express the membership function for $p \to q$ in terms of membership functions of either propositions p and $\sim q$ or $\sim p$ and q, thus giving the following

$$\mu_{p\to q}(x,y) = 1 - \mu_{p\cap q}(x,y) = 1 - \min\left\{\mu_p(x), 1-\mu_q(y)\right\} \tag{3.111}$$

$$\mu_{p\to q}(x,y) = \mu_{p\cup q}(x,y) = 1 - \max\left\{1-\mu_p(x), \mu_q(y)\right\}. \tag{3.112}$$

Instead of min and max, the product and algebraic sum for intersection and union may be respectively used. The two equations can be verified by substituting 1 for true and 0 for false.

Table 3.14 Truth table applied to propositions

p	q	$p \wedge q$	$p \vee q$	$p \rightarrow q$	$p \leftrightarrow q$	$\sim p$
T	T	T	T	T	T	F
T	F	F	T	F	F	F
F	T	F	T	T	F	T
F	F	F	F	T	T	T

In traditional propositional logic, there are two very important inference rules associated with implication and proposition, specifically the inferences *modus ponens* and *modus tollens*.

Modus ponens:
Premise 1: 'x is A';
Premise 2: 'IF x is A THEN y is B';
Consequence: 'y is B'.

Modus ponens is associated with the implication 'A implies B'. In terms of propositions p and q, modus ponens is expressed as

$$[p \wedge (p \rightarrow q)] \rightarrow q \tag{3.113}$$

Modus tollens:
Premise 1: 'y is not B';
Premise 2: 'IF x is A THEN y is B';
Consequence: 'x is not A'.

In terms of propositions p and q, modus tollens is expressed as

$$[(\sim q) \wedge (p \rightarrow q)] \rightarrow (\sim p) \tag{3.114}$$

Modus ponens plays a central role in engineering applications such as control logic, largely due to its basic consideration of cause and effect.

Modus tollens has in the past not featured in engineering applications, and has only recently been applied to engineering analysis logic such as in engineering design analysis with the application of FMEA and FMECA.

Although traditional fuzzy logic borrows notions from crisp logic, it is not adequate for engineering applications of *fuzzy* control logic, because cause and effect is the cornerstone of modelling in engineering control systems, whereas in traditional propositional logic it is not. Ultimately, this has prompted redefinition of fuzzy implication operators for engineering applications of fuzzy control logic. An understanding of *why* the traditional approach fails in engineering is essential. The extension of crisp logic to fuzzy logic is made by replacing the bivalent membership functions of crisp logic with *fuzzy membership functions*.

Thus, the IF–THEN statement:
'IF x is A, THEN y is B' where $x \in X$ and $y \in Y$
has a membership function

$$\mu_{p\to q}(x,y) \in [0,1] \tag{3.115}$$

Note that $\mu_{p\to q}(x,y)$ measures the degree of truth of the implication relation between x and y. This membership function can be defined as for the crisp case. In fuzzy logic, modus ponens is extended to a *generalised modus ponens*.

Generalised modus ponens:
Premise 1: 'x is A^*';
Premise 2: 'IF x is A THEN y is B';
Consequence: 'y is B^*'.

The difference between modus ponens and generalised modus ponens is subtle, namely the fuzzy set A^* is not the same as rule antecedent fuzzy set A, and fuzzy set B^* is not necessarily the same as rule consequent B.

d) Fuzzy Implication

Classical set theory operations can be extended from ordinary set theory to fuzzy sets. All those operations that are extensions of crisp concepts reduce to their usual meaning when the fuzzy subsets have membership degrees that are drawn from the set $\{0,1\}$. Therefore, extending operations to fuzzy sets, the same symbols are used as in set theory.

For example, let A and B be fuzzy subsets of a nonempty (crisp) set X.
The intersection of A and B is defined as

$$(A\cap B)(t) = T(A(t),B(t)) = A(t)\wedge B(t) \tag{3.116}$$

where:

$\wedge$ denotes the Boolean *conjunction* operation
(i.e. $A(t)\wedge B(t) = 1$ if $A(t) = B(t) = 1$
and $A(t)\wedge B(t) = 0$ otherwise).

Conversely:

$\vee$ denotes a Boolean *disjunction* operation
(i.e. $A(t)\vee B(t) = 0$ if $A(t) = B(t) = 0$
and $A(t)\vee B(t) = 1$ otherwise).
This will be considered more closely later.

and:

T is a t-norm. If $T = \min$, then we get:
$(A\cap B)(t) = \min\{A(t),B(t)\}$ for all $t \in X$.

If a proposition is of the form 'u is A' where A is a fuzzy set—for example, 'high pressure'—and a proposition is of the form 'v is B' where B is a fuzzy set—for example, 'small volume'—, then the membership function of the *fuzzy implication* $A \rightarrow B$ is defined as

$$(A \rightarrow B)(u,v) = f(A(u), B(v)) \tag{3.117}$$

where f is a specific function relating u to v. The following is used

$$(A \rightarrow B)(u,v) = A(u) \rightarrow B(v) \tag{3.118}$$

$A(u)$ is considered the truth value of the proposition 'u is high pressure', $B(v)$ is considered the truth value of the proposition 'v is small volume'.

e) Fuzzy Reasoning

We now turn our attention to the research of Dubois and Prade about representation of the different kinds of fuzzy rules in terms of *fuzzy reasoning* on *certainty* and *possibility* qualifications, and in terms of *graduality* (Dubois et al. 1992a,b,c).

Certainty rules This first kind of *implication-based fuzzy rule* corresponds to fuzzy reasoning statements of the form 'the more x is A, the more certain y lies in B'. Interpretation of this rule gives:

'$\forall u$, if $x = u$, it is at least $\mu_A(u)$ certain that y lies in B'

The degree $1 - \mu_A(u)$ is the *possibility* that y is outside of B when $x = u$, since the more x is A, the less possible y lies outside B, and the more certain y lies in B. In this case, the *certainty* of an event corresponds to the *impossibility* of the contrary event.

The conditional possibility distribution of this rule is

$$\forall u \in U,\ \forall v \in V \quad \pi_{y|x}(v,u) \leq \max\left(1 - \mu_A(u), \mu_A(v)\right) \tag{3.119}$$

where: π is the conditional possibility distribution that y relates to x.

In the particular case where A is an ordinary subset, Eq. (3.119) yields

$$\begin{aligned} &\forall u \in A \quad \pi_{y|x}(v,u) \leq \mu_B(v) \\ &\forall u \notin A \quad \pi_{y|x}(v,u) \text{ is completely unspecified .} \end{aligned} \tag{3.120}$$

This corresponds to the implication-based modelling of a fuzzy rule with a non-fuzzy condition.

Gradual rules This second kind of *implication-based fuzzy rule* corresponds to fuzzy reasoning statements of the form 'the more x is A, the more y is B'. Statements involving 'the less' in place of 'the more' are easily obtained by changing A (or B)

into its complement $\bar{A}$ (or $\bar{B}$), due to the equivalence between 'the more x is A' and 'the less x is $\bar{A}$' (with $\mu_{\bar{A}} = 1 - \mu_A$).

More precisely, the intended meaning of a gradual rule can be understood in the following way: 'the greater the degree of membership of the value of x to the fuzzy set A and the more the value of y is considered to be in relation (in the sense of the rule) with the value of x, the greater the degree of membership the value of y should be to B', i.e.

$$\forall u \in U \quad \min\left(\mu_A(u), \pi_{y|x}(v,u)\right) \leq \mu_B(v)\,. \tag{3.121}$$

Possibility rules This kind of *conjunction-based fuzzy rule* corresponds to fuzzy reasoning statements of the form 'the more x is A, the more possible B is a range for y'. Interpretation of this rule gives:

'$\forall u$, if $x = u$, it is at least $\mu_A(u)$ possible that B is a range for y'

This yields the conditional possibility distribution $\pi_{y|x}(u)$ to represent the rule when $x = u$

$$\forall u \in U,\ \forall v \in V \quad \min\left(\mu_A(u), \mu_B(v)\right) \leq \pi_{y|x}(v,u)\,. \tag{3.122}$$

The degree of possibility of the values in B is lower bounded by $\mu_A(u)$.

3.3.2.6 Theory of Approximate Reasoning

Zadeh introduced the theory of approximate reasoning (Zadeh 1979). This theory provides a powerful framework for reasoning in the face of imprecise and uncertain information, typically such as for engineering design. Central to this theory is the representation of propositions as statements, assigning fuzzy sets as values to variables.

For example, suppose we have two interactive variables $x \in X$ and $y \in Y$ and the causal relationship between x and y is known. In other words, we know that y is a function of x, or $y = f(x)$, and then the following inferences can be made (cf. Fig. 3.30):

$$\text{“}y = f(x)\text{”}\ \&\ \text{“}x = x_1\text{”}\ \rightarrow\ \text{“}y = f(x_1)\text{”}$$

This inference rule states that if $y = f(x)$ for all $x \in X$ and we observe that $x = x_1$, then y takes the value $f(x_1)$. However, more often than not, we do not know the complete causal link f between x and y, and only certain values $f(x)$ for some particular values of x are known, that is

$$R_i\text{: If } x = x_i \text{ then } y = y_i\,, \quad \text{for } i = 1, \ldots, m \tag{3.123}$$

where R_i is a particular rule-base in which the values of x_i ($i = 1, \ldots, m$) are known. Suppose that we are given an $x \in X$ and want to find a $y \in Y$ that corresponds to x

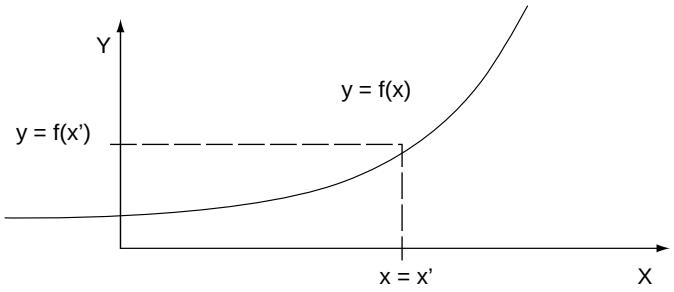


Fig. 3.30 Simple crisp inference

under the rule-base $R = \{R_i, \ldots, R_m\}$, then this problem is commonly approached through *interpolation*.

Let x and y be linguistic variables, e.g. 'x is high' and 'y is small'. Then, the basic problem of approximate reasoning is to find a membership function of the consequence C from the stated rule-base $R = \{R_i, \ldots, R_n\}$ and the fact A, where R_i is of the form

$$R_i: \text{ if } x \text{ is } A_i \text{ then } y \text{ is } C_i \tag{3.124}$$

In fuzzy logic and approximate reasoning, the most important fuzzy implication inference rule is the *generalised modus ponens* (*GMP*; Fullér 1999). As previously indicated, the classical modus ponens inference rule states:

Premise	if p then q
Fact	p
Consequence	q

This inference rule can be interpreted as:

If p is true and $p \rightarrow q$ (p implicates q) is true, then q is true.

The fuzzy implication inference $\rightarrow$ is based on the compositional rule of inference for approximate reasoning, which states (Zadeh 1973):

Premise	if x is A then y is B
Fact	x is A'
Consequence	y is B'

In addition to the phrase 'modus ponens' (where the term modus ponens $\Rightarrow$ method of argument), there are other special terms in approximate reasoning for the various features of these arguments. The 'If ... then' premise is called a *conditional*, and the two claims are similarly called the *antecedent* and the *consequent* where:

Main premise	<antecedent>
Helping premise	if <antecedent> then <consequent>
Conclusion	<consequent>

The valid connection between a premise and a conclusion is known as *deductive validity*.

From the classical modus ponens inference rule, the consequence B' is determined as a composition of the fact and the fuzzy implication operator $B' = A'^{\circ}(A \to B)$. Thus

$$\begin{aligned} &\text{For all } v \in V: \\ &B'(v) = \sup_{u \in U} \min\{A'(u), (A \to B)(u,v)\} \end{aligned} \tag{3.125}$$

where $\sup_{u \in U}$ is the fuzzy relations composition operator.

Instead of the fuzzy sup-min composition operator, the sup-T composition operator may be used, where T is a t-norm

$$\begin{aligned} &\text{For all } v \in V: \\ &B'(v) = \sup_{u \in U} T(A'(u), (A \to B)(u,v)) \end{aligned} \tag{3.126}$$

Use of the t-norm operator comes from the crisp max–min and max–prod compositions, where both min and prod are t-norms. This corresponds to the product of matrices, as the t-norm is replaced by the product, and sup is replaced by the sum. It is clear that T cannot be chosen independently of the implication operator. Suppose that A, B and A' are fuzzy numbers, then the generalised modus ponens should satisfy some rational properties that are given as (cf. Figs. 3.31a,b, 3.32a,b, 3.33a,b):

Property 1: basic property

if x is A then y is B	if pressure is high then volume is small
x is A	pressure is high
y is B	volume is small

Property 2: total indeterminance

if x is A then y is B	if pressure is high then volume is small
x is $\neg A$	pressure is not high
y is unknown	volume is unknown

where x is $\neg A$ means that x being an element of A is impossible (defined later).

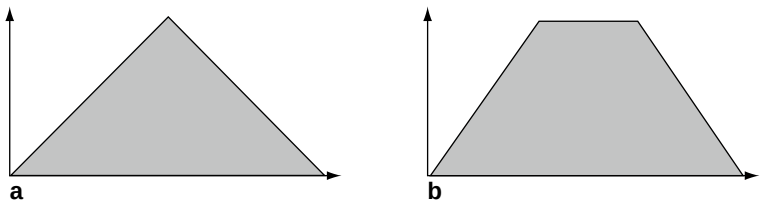


Fig. 3.31 **a** Basic property $A' = A$. **b** Basic property $B' = B$

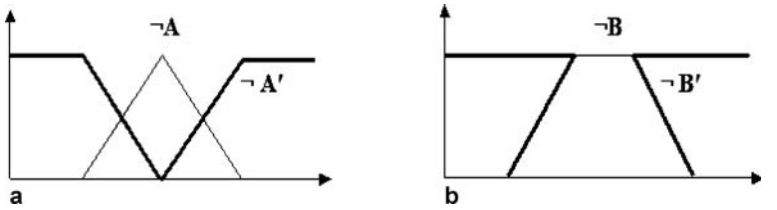


Fig. 3.32 **a**, **b** Total indeterminance

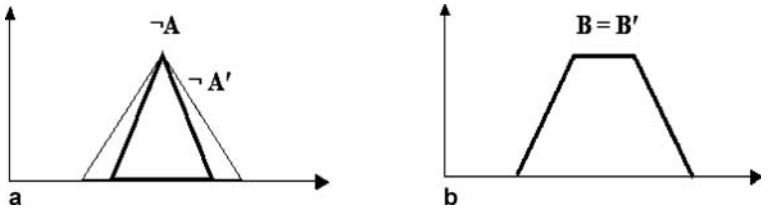


Fig. 3.33 **a**, **b** Subset property

The t-norms are represented as:

Property 3: subset

$$\frac{\text{if } x \text{ is } A \text{ then } y \text{ is } B \qquad x \text{ is } A' \subset A}{y \text{ is } B} \qquad \frac{\text{if pressure is high then volume is small} \qquad \text{pressure is very high}}{\text{volume is small}}$$

where x is $A' \subset A$ means x is an element of the subset of A' with A.

3.3.2.7 Overview of Possibility Theory

The basic concept of possibility theory, introduced by Zadeh, is to use fuzzy sets that no longer simply represent the gradual aspect of vague concepts such as 'high', but also represent *incomplete knowledge* subject to *uncertainty* (Zadeh 1979). In such a situation, the fuzzy variable 'high' represents the only information available on some parameter value (such as pressure). In possibility theory, uncertainty is described using dual *possibility* and *necessity* measures defined as follows (Dubois et al. 1988):

A possibility measure $\prod$ defined on a finite propositional language, and valued on $[0,1]$, satisfies the following axioms:

a) $\prod(\bot) = 0\,; \quad \prod(\top) = 1$
b) $\forall p, \forall q\,, \quad \prod(p \vee q) = \max[\prod(p), \prod(q)]$
c) if p is equivalent to q, then $\prod(p) = \prod(q)$

where:

$\perp$ and $\top$ denote the ever-false proposition (contradiction) and the ever-true proposition (tautology) respectively.

$\forall p$ denotes 'for all p' and $\forall q$ denotes 'for all q', and $\vee$ denotes a Boolean *disjunction* operation (i.e. $p \vee q = 0$ if $p = q = 0$ and $p \vee q = 1$ otherwise)

and, conversely, $\wedge$ denotes the Boolean *conjunction* operation (i.e. $p \wedge q = 1$ if $p = q = 1$ and $p \wedge q = 0$ otherwise)

Axiom b) means that $p \vee q$ is possible as soon as one of p or q is possible, including the case when both are so.

$\prod(p) = 1$ means that p is to be expected but not that p is sure, since $\prod(p) = 1$ is compatible with $\prod(\neg p) = 1$ as well.

On the contrary, $\prod(p) = 0$ implies $\prod(\neg p) = 1$ where $\neg p$ means that p is impossible.

a) Deviation of Possibility Theory from Fuzzy Logic

It must be emphasised that only the following proposition holds in the general case, since $p \wedge q$ is rather impossible

$$\prod(p \wedge q) \leq \min\left(\prod(p)\ ,\ \prod(q)\right) \tag{3.127}$$

(e.g. if $q = \neg p$, $p \wedge q$ is $\perp$, which is impossible) while p as well as q may remain somewhat possible under a state of incomplete information.

More generally, $\prod(p \wedge q)$ is not only a function of $\prod(p)$ and of $\prod(q)$. This departs completely from fully truth functional multiple-valued calculi, which is referred to as *fuzzy logic* (Lee 1972), specifically where the truth of *vague* propositions is a matter of degree.

In possibility theory, a *necessity measure* N is associated by duality with a *possibility measure* $\prod$, such that

$$\forall p\ ,\ N(p) = 1 - \prod(\neg p) \tag{3.128}$$

It means that p is all the more certain as $\neg p$ is impossible. Axiom b) is then equivalent to

$$\forall p\ ,\ \forall q\ ,\ N(p \wedge q) = \min(N(p), N(q)) \tag{3.129}$$

This means that for being certain about $p \wedge q$, we should be both certain of p and certain of q, and that the level of certainty of $p \wedge q$ is the *smallest* level of certainty

attached to p and to q. Note that

$$N(p) > 0 \Leftrightarrow \prod(\neg p) < 1 \Rightarrow \prod(p) = 1$$

Since:

$$\max\left(\prod(p), \prod(\neg p)\right) = \prod(p \vee \neg p) = \prod(\top) = 1$$

And:

$$N(p \vee q) \geq \max(N(p), N(q)) \tag{3.130}$$

This means we may be somewhat certain of the imprecise statement $p \vee q$ without being at all certain that p is true or that q is true.

The following conventions are adopted in *possibility theory* where the *possible* values of the pair of necessity and possibility measures, $(N, \prod)$, are represented

$$\prod(p) = \max_{\omega \in [p]} \pi(\omega) \tag{3.131}$$

where:

$\prod(p)$ is the possibility measure of proposition p
ω is a representation of available knowledge
$[p]$ is the set of interpretations that make p true, i.e. the models of p
$\pi(\omega)$ is the possibility distribution of available knowledge.

Thus, starting with the plausibility of available knowledge represented by the distribution π of possible interpretations of such available knowledge, two functions of the possibility measure $\prod$ and the necessity measure N are defined that enable us to make an assessment of the uncertainty surrounding the proposition p. Ignorance is represented by a uniform possibility distribution equal to 1.

Conversely, given certain constraints $i = 1, n$

$$N(p_i) \geq \alpha_i > 0 \quad \text{for } i = 1, n \tag{3.132}$$

where:

$N(p_i)$ is the certainty measure of a particular proposition p in the set with constraints $i = 1, n$
α_i is the possibility distribution with the least restrictive constraints.

Thus, expressing a level of certainty for a collection of propositions under certain constraints, we can compute the largest possibility distribution α_i that is the least restricted by these constraints.

It should be noted that *probabilistic reasoning* does not allow for the distinction between:

the *possibility* that p is true $(\prod(p) = 1)$ and
the *certainty* that p is true $(N(p) = 1)$,
nor between:
the *certainty* that p is false $(N(\neg p) = 1 \Leftrightarrow \prod(p) = 0)$ and
the *absence of certainty* that p is true $(N(p) = 0 \Leftrightarrow \prod(\neg p) = 1)$.

Possibility theory thus contrasts with *probability theory* in which:

$P(\neg p) = 1 - P(p)$, i.e. the probability that p is impossible is 1 minus the probability that p is possible, and therefore:
$P(\neg p) = 1 \Leftrightarrow P(p) = 0$, i.e. the probability that p is impossible is true implies that the probability of p being possible is false, and
$N(p) = 0$ does not entail $N(\neg p) = 1$.

While in possibility theory, if the certainty measure N of the *possibility* of the proposition p is false, then this does not necessarily imply that the certainty measure N of the *impossibility* of proposition p is true. In this context, the distinction between *possibility* and *certainty* is crucial for distinguishing between *contingent* and *sure* effects respectively in engineering design analyses such as FMEA and FMECA.

The incomplete states of knowledge captured by possibility theory cannot be modelled by a single, well-defined probability distribution. They rather correspond to what might be called 'higher-order uncertainty', which actually means 'ill-known probabilities' (Cayrac et al. 1995). This type of uncertainty is modelled either by second-order probabilities or by interval-valued probabilities, which is complex.

Possibility theory offers a very simple substitute to these higher-order uncertainty theories, as well as a common framework for the modelling of uncertainty and imprecision in reasoning applications such as engineering design analysis. The use of max and min operations in this case satisfies the requirement for computational simplicity, and for the qualitative nature of uncertainty that can be expressed in many real-world applications. Thus, in possibility theory the modelling of uncertainty remains qualitative (Dubois et al. 1988).

b) Rationals for the Choice of Possibility Theory in Engineering Design Analysis

The complexity arising from an integration of engineering systems and their interactions makes it impossible to gather meaningful statistical data that could allow for the use of objective probabilities in engineering design analysis. Even subjective probabilities in design analysis (for example, where all the possible failure modes in an FMECA may be ordered in a criticality ranking according to prior knowledge) are fundamentally not acceptable to process or systems engineering experts.

For example, process design engineers would not be able to compare failure modes involving different equipment, or different operational domains (thermal, electrical, mechanical, etc.) in complex systems integration. At best, a partial prior ordering of failure modes identified for each individual system may be made. In addition, the number of failure modes that are generally represented in an FMECA do not encompass all the possible failures that could arise in reality as a result of a complex integration of systems. This complexity makes any engineering design knowledge base incomplete. The only intended purpose of the FMECA in engineering design analysis would therefore be primarily as a support tool for the understanding of *design integrity*, in which failure consequences are initially ranked by decreasing compatibility with their failure modes, and then ranked according to their direct relevance to an applicable measure of severity.

3.3.2.8 Uncertainty and Incompleteness in Engineering Design Analysis

Uncertainty and incompleteness is inherent to engineering design analysis. Uncertainty, arising from the complex integration of systems, can best be expressed in qualitative terms, necessitating the results to be presented in the same qualitative measures. This causes problems in analysis based upon a probabilistic framework. The only acceptable framework for an approach to qualitative probability is that of *comparative probabilities* proposed by Fishburn (1986), but its application is not easy at the practical level because its representational requirements are exponential (Cayrac et al. 1994).

An important question is to decide what kind of possibility theory or fuzzy logic representation (in the form of fuzzy sets) is best suited for engineering design analysis. The use of conjunction-based representations is perceived as *not* suitable from the point of view of logic that is automated, because conjunction-based fuzzy rules do not fit well with the usual meaning of rules in artificial intelligence-based *expert systems*. This is important because it is eventually within an *expert system framework* that engineering design analysis such as FMEA and FMECA should be established, in order to be able to develop intelligent computer automated methodology in determining the integrity of engineering design. The concern raised earlier that qualitative reasoning algorithms may not be suitable for FMEA or FMECA is thus to a large extent not correct.

This consideration is based on the premise that the FMEA or FMECA formalism of analysis requires unique predictions of system behaviour and, although some vagueness is permissible due to *uncertainty*, it cannot be ambiguous, despite the consideration that ambiguity is an inherent feature of computational qualitative reasoning (Bull et al. 1995b).

Implication-based representations of fuzzy rules may be viewed as constraints that restrict a set of possible solutions, thus eliminating any ambiguity. A possible explanation for the concern may be that two predominate types of engineering reasoning applied in engineering design analysis—systems engineering and knowledge engineering—do not have the same background. The former is usually data-driven, and applies analytic methods where analysis models are derived from data. In general, fuzzy sets are also viewed as data, resulting in any form of reasoning methodology to be based on accumulating data. Incoherency issues are not considered because incoherence is usually unavoidable in any set of data. On the contrary, knowledge engineering is knowledge-driven, and a fuzzy rule is an element of knowledge that constrains a set of possible situations. The more fuzzy rules, the more information, and the more precise one can get. Fuzzy rules clearly stand at the crossroad of these two types of engineering applied to engineering design analysis.

In the use of FMECA for engineering design analysis, the objective is to develop a flexible representation of the *effects* and *consequences* of failure modes down to the relevant level of detail, whereby available knowledge—whether incomplete or uncertain—can be expressed. The objective thus follows qualitative analysis methodology in handling uncertainty with possibility theory and fuzzy sets in fault diagnostic applications, utilising FMECA (Cayrac et al. 1994).

An expansion of FMEA and FMECA for engineering design analysis is developed in this handbook, particularly for the application of *reliability assessment* during the *preliminary and detail design phases* of the engineering design process. The expanded methodology follows the first part of the methodology proposed by Cayrac (Cayrac et al. 1994), but not the second part proposed by Cayrac, which is a further exposition of the application of *fault diagnosis* using FMECA. A detailed description of introducing uncertainty in such a causal model is given by Dubois and Prade (Dubois et al. 1993).

3.3.2.9 Modelling Uncertainty in FMEA and FMECA

In modelling *uncertainty* with regard to *possible failure* as described by failure modes in FMEA and FMECA, consider the following: let D be the set of possible *failure modes*, or *disorders* $\{d_1,\ldots,d_i,\ldots,d_p\}$ of a given causal FMEA and FMECA analysis, and let M be a set of observable *consequences*, or *manifestations* $\{m_1,\ldots,m_j,\ldots,m_n\}$ related to these failure modes. In this model, *disorders* and *manifestations* are either present or absent. For a given disorder d, we express its (more or less) certain manifestations, gathered in the fuzzy set $M(d)+$, and those that are (more or less) impossible, gathered in the fuzzy set $M(d)-$.

Thus, the fuzzy set $M(d)+$ contains manifestations that (more or less) surely *can* be caused by the presence of a given disorder d alone. In terms of membership functions

$$\mu_{M(d)+}(m) = 1\,. \tag{3.133}$$

This means that the manifestation m exists in the fuzzy set of certain manifestations for a given disorder d. This also means that m is *always* present when d alone is present.

Conversely, the set $M(d)-$ contains manifestations that (more or less) surely *cannot* be caused by d alone. Thus

$$\mu_{M(d)-}(m) = 1\,. \tag{3.134}$$

This means that the manifestation m *does not* exist in the fuzzy set of impossible manifestations for a given disorder d. This also means that m is *never* present when d alone is present.

Complete ignorance regarding the relation between a disorder and a manifestation (we do not know whether m can be a consequence of d) is expressed by

$$\mu_{M(d)+}(m) = \mu_{M(d)-}(m) = 0\,. \tag{3.135}$$

Intermediate membership degrees allow a gradation of the uncertainty.

The fuzzy sets $M(d)+$ and $M(d)-$ are not *possibility* distributions because manifestations are clearly not mutually exclusive. Furthermore, the two membership functions $\mu_{M(d)+}(m)$ and $\mu_{M(d)-}(m)$ both express *certainty* levels that the manifestation m is present and absent respectively, when disorder d alone takes place.

a) Logical Expression of FMECA

FMECA information (without uncertainty) can be expressed as a theory T consisting of a collection of clauses:

$\neg d_i \vee m_j$ corresponds to a non-fuzzy set of *certain* manifestations $M(d_i)+$, which means *either* that the disorders $\neg d_i$ are impossible *or* that the manifestations m_j are possible in a non-fuzzy set of manifestations $M(d_i)+$,

$\neg d_i \vee \neg m_k$ corresponds to a non-fuzzy set of *impossible* manifestations $M(d_i)-$, which means *either* that the disorders $\neg d_i$ are impossible *or* that manifestations $\neg m_k$ are impossible in a non-fuzzy set of manifestations $M(d_i)-$ (i.e. manifestations that *cannot* be caused by d_i alone),

where $\vee$ denotes the Boolean disjunction operation
($\neg d_i \vee m_j = 0$ if $\neg d_i = m_j = 0$, and $\neg d_i \vee m_j = 1$ otherwise).

A *disjunction* is associated with indicative linguistic statements compounded with *either ... or*, such as $(\neg d_i \vee m_j) \Rightarrow$ either the disorders are *impossible* or the manifestations are *possible*. However, the term *disjunction* is currently more often used with reference to linguistic statements or *well-formed formulae* (*wff*) of associated form occurring in formal languages. Logicians distinguish between the abstracted *form* of such linguistic statements and their roles in arguments and proofs, and the *meanings* that must be assigned to such statements to account for those roles (Artale et al. 1998). The abstracted *form* represents the *syntactic* and *proof-theoretic* concept, and the *meanings* the *semantic* or *truth-theoretic* concept in disjunction. Disjunction is a binary truth-function, the output of which is true if at least one of the input values (disjuncts) is true, and false otherwise. Disjunction together with negation provide sufficient means to define all truth-functions—hence, the use in a logical expression of FMECA.

If the disjunctive constant $\vee$ (historically suggestive of the Latin *vel* (*or*)) is a primitive constant of the linguistic statement, there will be a clause in the inductive definition of the set of well-formed formulae (wffs).

Using α and β as variables ranging over the set of well-formed formulae, such a clause will be:

$$\text{If } \alpha \text{ is a wff and } \beta \text{ is a wff, then } \alpha \vee \beta \text{ is a wff}$$

where $\alpha \vee \beta$ is the *disjunction* of the wffs α and β, and interpreted as '[name of first wff] vel ('or') [name of second wff]'.

In presentations of classical systems in which the conditional implication $\rightarrow$ or the subset $\supset$ and the negational constant $\neg$ are taken as primitive, the disjunctive constant $\vee$ will also feature in the abbreviation of a wff:

$$\neg\alpha \rightarrow \beta \text{ (or } \neg\alpha\neg\beta) \text{ as } \alpha \vee \beta$$

Alternatively, if the conjunctive & has already been introduced as a defined constant, then $\vee$ will also feature in the abbreviation of a wff:

$$\neg(\neg\alpha \,\&\, \neg\beta) \text{ as } \alpha \vee \beta$$

In its simplest, classical semantic analysis, a disjunction is understood by reference to the conditions under which it is true, and under which it is false. Central to the definition is a *valuation*, a function that assigns a value in the set $\{1,0\}$. In general, the inductive truth definition for a linguistic statement corresponds to the definition of its well-formed formulae. Thus, for a propositional linguistic statement, it will take as its basis a clause according to which an elemental part is true or false accordingly as the valuation maps it to 1 or to 0. In systems in which $\vee$ is a primitive constant, the clause corresponding to disjunction takes $\alpha \vee \beta$ to be true if at least one of α, β is true, and takes it to be false otherwise. Where $\vee$ is introduced by the definitions given earlier, the truth condition can be computed for $\alpha \vee \beta$ from those of the conditional ($\rightarrow$ or $\supset$) or conjunction (&) and negation ($\neg$).

In slightly more general perspective, then, if the disorders interact in the manifestations they cause, d_i can be replaced by a conjunction of d_k.

This general perspective is justification of the form (Cayrac et al. 1994):

$$\neg d_{i1} \wedge \cdots \wedge \neg d_{i(k)} \vee m_j \tag{3.136}$$

where the conjunctive $\wedge$ is used in place of &. Thus, 'intermediary entities' between disorders and manifestations are allowed. In other words, in failure analysis, intermediary 'effects' feature *between* failure modes and their consequences, which is appropriate to the theory on which the FMECA is based. This logical modelling of FMECA is, however, not completely satisfactory, as $\neg d_i \vee \neg m_k$ means *either* that the disorder $\neg d_i$ is impossible *or* that the manifestations $\neg m_k$ are impossible. This could mean that d_i *disallows* m_k, which is different to the fuzzy set $\mu_{M(d)-}(m) > 0$, since the disorder $\neg d_i$ being impossible only means that d_i alone is not capable of producing m_k. This does not present a problem under a single failure mode assumption but it does complicate the issue if *simultaneous* failure modes or disorders are allowed.

In Sect. 3.3.2.1, *failure mode* was described from three points of view:

- A complete functional loss.
- A partial functional loss.
- An identifiable condition.

For reliability assessment during the engineering design process, the first two failure modes—specifically, a complete functional loss, and a partial functional loss—can be practically considered. The determination of an identifiable condition would be considered when contemplating the possible *causes* of a complete functional loss or of a partial functional loss. Thus, simultaneous failure modes or disorders in FMECA would imply *both* a complete functional loss and a partial functional loss—which is contradictory. The application of the fuzzy set $\mu_{M(d)-}(m) > 0$ is thus valid in FMECA, since the implication is valid that d_i alone is *not* capable of producing m_k.

However, in the logical expressions of FMECA, two difficulties arise

$$\neg d_i \vee m_k \text{ and } \neg d_j \vee m_k \text{ imply } \neg (d_i \wedge d_j) \vee m_k \tag{3.137}$$

Equation (3.137) implies that those clauses where *either* disorder $\neg d_i$ is impossible *or* manifestations m_k are possible in a non-fuzzy set of certain manifestations $M(d_i)+$, and where *either* disorder $\neg d_j$ is impossible *or* manifestations m_k are possible in a non-fuzzy set of certain manifestations $M(d_j)+$ imply that *either* disorder $\neg d_i$ and disorder $\neg d_j$ are impossible *or* manifestations m_k are possible in non-fuzzy sets of certain manifestations $M(d_i)+$ and $M(d_j)+$. This logical approach implicitly involves the assumption of *disorder independence* (i.e. independent failure modes), leading to manifestations of simultaneous disorders. In other words, it assumes failure modes are independent but may occur simultaneously.

This approach may be in contradiction with knowledge about joint failure modes expressing $\neg(d_i \wedge d_j) \vee \neg m_k$ where *either* disorder $\neg d_i$ *and* disorder $\neg d_j$ are impossible *or* where the relating manifestations m_k are impossible in the non-fuzzy sets of manifestations $M(d_i)-$ and $M(d_j)-$.

The second difficulty that arises in the logical expressions of FMECA is

$$\neg d_i \vee \neg m_k \text{ and } \neg d_j \vee \neg m_k \text{ imply } \neg(d_i \wedge d_j) \vee \neg m_k \tag{3.138}$$

Equation (3.138) implies that those clauses where *either* disorder $\neg d_i$ is impossible *or* manifestations $\neg m_k$ are impossible in the non-fuzzy set of $M(d_i)-$ that contains manifestations that *cannot* be caused by d_i alone, and where *either* disorder $\neg d_j$ is impossible *or* manifestations $\neg m_k$ are impossible in a non-fuzzy set $M(d_j)-$ that contains manifestations that *cannot* be caused by d_j alone imply that *either* disorder $\neg d_i$ *and* disorder $\neg d_j$ are impossible *or* manifestations $\neg m_k$ are impossible in the non-fuzzy sets $M(d_i)-$ and $M(d_j)-$, which together contain manifestations that *cannot* be caused by d_i *and* d_j alone. This is, however, in *disagreement* with the assumption

$$M-\left(\left\{d_i,d_j\right\}\right) = M-(\{d_i\}) \cap M-\left(\left\{d_j\right\}\right) \tag{3.139}$$

Equation (3.139) implies that the fuzzy set of accumulated manifestations that cannot be caused by the simultaneous disorders $\{d_i,d_j\}$ is equivalent to the intersect of the fuzzy set of manifestations that cannot be caused by the disorder d_i alone, and the fuzzy set of manifestations that cannot be caused by the disorder d_j alone (it enforces a union for $M+(\{d_i,d_j\})$.

In the logical approach, if $\neg d_i \vee \neg m_k$ and $\neg d_j \vee \neg m_k$ hold, this disallows the simultaneous assumption that d_i and d_j are present, which is then not a problem under the *single failure mode* assumption, as indicated in Sect. 3.3.2.1.

On the contrary, $m_k \in M+(d_j) \cap M-(d_i)$ does not forbid $\{d_i,d_j\}$ from being a potential explanation of m_k even if the presence (or absence) of m_k eliminates d_i (or d_j) alone.

b) Expression of Uncertainty in FMECA

In the following logical expressions of FMECA, the single failure mode assumption is made (i.e. *either* a complete functional loss *or* a partial functional loss). Uncertainty in FMECA can be expressed using *possibilistic logic* in terms of a necessity measure N. For example

$$N(\neg d_i \vee m_j) \geq \alpha_{ij} \tag{3.140}$$

where:

$N(\neg d_i \vee m_j)$ is the *certainty measure* of a particular proposition that *either* disorder $\neg d_i$ is impossible *or* manifestations m_j are possible in a non-fuzzy set of certain manifestations $M(d_i)+$, and

α_{ij} is the *possibility distribution* relating to constraint i of the disorder d_i and constraint j of manifestation m_j.

The *generalised modus ponens* of possibilistic logic (Dubois et al. 1994) is

$$\begin{aligned} &N(d_i) \geq \gamma_i \text{ and } N(\neg d_i \vee m_j) \geq \alpha_{ij} \\ &\Rightarrow N(m_j) \geq \min(\gamma_i, \alpha_{ij}) \end{aligned} \tag{3.141}$$

where:

$N(d_i)$ is the certainty measure of the proposition that the disorder d_i is certain, γ_i is the possibility distribution relating to constraint i of disorder d_i and

$N(m_j)$ is the certainty measure of the proposition that the manifestation m_j is certain, and bound by the minimum cut set of the possibility distributions γ_i and α_{ij}. In other words, the presence of the manifestation m_j is all the more certain, as the disorder d_i is certainly present, and that m_j is a certain consequence of d_i.

3.3.2.10 Development of the Qualitative FMECA

A further extension of the FMECA is considered, in which representation of *indirect* links between disorders and manifestations are also made. In addition to disorders and manifestations, intermediate entities called *events* are considered (Cayrac et al. 1994).

Referring to Sect. 3.3.2.1, these events may be viewed as *effects*, where *the effects of failure are associated with the immediate results within the component's or assembly's environment.*

Disorders (failure modes) can cause events (effects) and/or manifestations (consequences), where events themselves can cause other events and/or manifestations (i.e. failure modes can cause effects and/or consequences, where effects themselves can cause other effects and/or consequences). Events may not be directly observable.

An FMECA can therefore be defined by a theory consisting of a collection of clauses of the form

$$\neg d_i \vee m_j \ , \quad \neg d_k \vee e_1 \ , \quad \neg e_m \vee e_n \ , \quad \neg e_p \vee m_q$$

and, to express negative information,

$$\neg d_{i'} \vee \neg m_{j'} \ , \quad \neg d_{k'} \vee \neg e_{1'} \ , \quad \neg e_{m'} \vee \neg e_{'n} \ , \quad \neg e_{p'} \vee m_{q'}$$

where d represents disorders (failure modes), m represents manifestations (consequences), and e represents events (effects). All these one-condition clauses are weighted by a lower bound equal to 1 if the implication is certain. The positive and negative observations (m or $\neg m$) can also be weighted by a lower bound of a necessity degree. From the definitions above, it is possible to derive the direct relation between disorders and manifestations (failure modes and consequences), characterised by the fuzzy sets $\mu_{M(d)+}(m)$ and $\mu_{M(d)-}(m)$ as shown in the following relations (Dubois et al. 1994):

$$\begin{aligned} \mu_{M(d_i)+}(m_j) &= \alpha_{ij} \\ \mu_{M(d_i)-}(m_j) &= \gamma_{ij} \end{aligned} \tag{3.142}$$

The extended FMECA allows for an expression of uncertainty in engineering design analysis that evaluates the extent to which the identified fault modes can be *discriminated* during the *detail design phase* of the engineering design process. The various failure modes are expressed with their (more or less) *certain* effects and consequences. The categories of more or less *impossible* consequences are also expressed if necessary. After this refinement stage, if a set of failure modes cannot be *discriminated* in a satisfying way, the inclusion of the failure mode in the analysis is questioned.

The *discriminability* of two failure modes d_i and d_j is maximum when a *sure consequence* of one is an *impossible consequence* of the other. This can be extended to the fuzzy sets previously defined. The discriminability of a set of disorders D can be defined by

$$\begin{aligned} \text{Discrimin}(D) &= \min_{d_i, d_j \in D, i \neq j} \max(F) \\ \text{Where: } F &= \text{cons}(M(d_i)+, M(d_j)-) \ , \\ &\quad\ \text{cons}(M(d_i)-, M(d_j)+) \end{aligned} \tag{3.143}$$

and $\text{cons}(M(d_i)+, M(d_j)-)$ is the consistency of disorders d_i and d_j in the non-fuzzy set of *certain* manifestations $M(d_i)+$, as well as in the non-fuzzy set of *impossible* manifestations $M(d_j)-$:

and $\text{cons}(M(d_i)-, M(d_j)+)$ is the consistency of disorders d_i and d_j in the non-fuzzy set of *impossible* manifestations $M(d_i)-$, as well as in the non-fuzzy set of *certain* manifestations $M(d_j)+$.

For example, referring to the three types of failure modes:

The discriminability of the failure mode total loss of function (TLF) represented by the disorder d_1 and failure mode partial loss of function (PLF) represented by disorder d_2 is: Discrimin $(\{d_1, d_2\}) = 0$.

The discriminability of the failure mode total loss of function (TLF) represented by disorder d_1 and failure mode potential failure condition (PFC) represented by disorder d_3 is: Discrimin $(\{d_1, d_3\}) = 0.5$.

The discriminability of the failure mode partial loss of function (PLF) represented by disorder d_2 and failure mode potential failure condition (PFC) represented by disorder d_3 is: Discrimin $(\{d_2, d_3\}) = 0.5$.

a) Example of Uncertainty in the Extended FMECA

Tables 3.15 to 3.19 are extracts from an FMECA worksheet of a RAM analysis field study conducted on an environmental plant for the recovery of sulphur dioxide emissions from a non-ferrous metals smelter to produce sulphuric acid. The FMECA covers the pump assembly, pump motor, MCC and control valve components, as well as the pressure instrument loops of the reverse jet scrubber pump no. 1.

Three failure modes are normally defined in the FMECA as:

- TLF ⇒ 'total loss of function',
- PLF ⇒ 'partial loss of function',
- PFC ⇒ 'potential failure condition'.

Five consequences are normally defined in the FMECA as:

- Safety (by risk description)
- Environmental
- Production
- Process
- Maintenance.

The 'critical analysis' column of the FMECA worksheet includes items numbered 1 to 5 that indicate the following:

(1) Probability of occurrence (given as a percentage value)
(2) Estimated failure rate (the number of failures per year)
(3) Severity (expressed as a number from 0 to 10)
(4) Risk (product of 1 and 3)
(5) Criticality value (product of 2 and 4).

The semi-qualitative criticality values are ranked accordingly:

(1) High criticality ⇒ +6 onwards
(2) Medium criticality ⇒ +3 to 6 (i.e. 3.1 to 6.0)
(3) Low criticality ⇒ +0 to 3 (i.e. 0.1 to 3.0)

Table 3.15 Extract from FMECA worksheet of quantitative RAM analysis field study: RJS pump no. 1 assembly

System	Assembly	Failure description	Failure mode	Failure effect	Failure consequence	Cause of failure	Critical analysis
Reverse jet scrubber	RJS pump no. 1	Shaft leakage	TLF	Unsafe operating conditions for personnel	Injury risk	Seal elements broken or pump shaft damaged due to loss of alignment or seals not correctly fitted	(1) 50% (2) 2.50 (3) 11 (4) 5.5 (5) 13.75 High criticality
Reverse jet scrubber	RJS pump no. 1	Shaft leakage	TLF	Unsafe operating conditions for personnel	Injury risk	Seal elements broken or pump shaft damaged due to the seal bellow cracking because the rubber hardens in service	(1) 50% (2) 2.50 (3) 11 (4) 5.5 (5) 13.75 High criticality
Reverse jet scrubber	RJS pump no. 1	Restricted or no circulation	TLF	Prevents quenching of the gas and protection of the RJS structure due to reduced flow. Standby pump should start up and emergency water system may start up and supply water to weir bowl. Gas supply may be cut to plant. RJS damage unlikely	Maintenance	Loss of drive due to coupling connection failure caused by loss of alignment or loose studs	(1) 100% (2) 3.00 (3) 2 (4) 2.00 (5) 6.00 *Medium/high criticality*

Table 3.15 (continued)

System	Assembly	Failure description	Failure mode	Failure effect	Failure consequence	Cause of failure	Critical analysis
Reverse jet scrubber	RJS pump no. 1	Restricted or no circulation	TLF	Prevents quenching of the gas and protection of the RJS structure due to reduced flow. Standby pump should start up and emergency water system may start up and supply water to weir bowl. Gas supply may be cut to plant. RJS damage unlikely	Maintenance	Air intake at shaft seal area due to worn or damaged seal faces caused by solids ingress or loss of seal flushing	(1) 100% (2) 2.50 (3) 2 (4) 2.00 (5) 5.00 *Medium criticality*
Reverse jet scrubber	RJS pump no. 1	Excessive vibration	PFC	No immediate effect other than potential equipment damage	Maintenance	Bearing deterioration due to worn coupling out of alignment	(1) 100% (2) 2.00 (3) 1 (4) 1.0 (5) 2.00 *Low criticality*
Reverse jet scrubber	RJS pump no. 1	Excessive vibration	PFC	No immediate effect other than potential equipment damage	Maintenance	Bearing deterioration due to low barrel oil level or leaking seals	(1) 100% (2) 1.00 (3) 1 (4) 1.0 (5) 1.00 *Low criticality*
Reverse jet scrubber	RJS pump no. 1	Excessive vibration	PFC	No immediate effect other than potential equipment damage	Maintenance	Cavitations due to excessive flow or restricted suction condition	(1) 100% (2) 1.50 (3) 1 (4) 1.0 (5) 1.50 *Low criticality*

Table 3.16 Extract from FMECA worksheet of quantitative RAM analysis field study: motor RJS pump no. 1 component

Assembly	Component	Failure description	Failure mode	Failure effect	Failure consequence	Cause of failure	Critical analysis
RJS pump no. 1	Motor RJS pump no. 1	Motor fails to start or drive pump	TLF	Motor failure prevents quenching of the gas and the protection of the RJS structure due to reduced flow. Standby pump should start up automatically	Maintenance	Loose or corroded connections or motor terminals	(1) 100% (2) 0.50 (3) 2 (4) 2.0 (5) 1.00 *Low criticality*
RJS pump no. 1	Motor RJS pump no. 1	Motor fails to start or drive pump	TLF	Motor failure prevents quenching of the gas and the protection of the RJS structure due to reduced flow. Standby pump should start up automatically	Maintenance	Motor winding short or insulation fails	(1) 100% (2) 0.25 (3) 2 (4) 2.0 (5) 0.50 *Low criticality*
RJS pump no. 1	Motor RJS pump no. 1	Motor cannot be stopped or started locally	TLF	If required to respond in an emergency failure of motor, this could result in injury risk	Injury risk	Local stop/start switch fails	(1) 50% (2) 0.25 (3) 11 (4) 5.5 (5) 1.38 *Low criticality*
RJS pump no. 1	Motor RJS pump no. 1	Motor overheats and trips	PFC	Motor failure prevents quenching of the gas and the protection of the RJS structure due to reduced flow. Standby pump should start up automatically	Maintenance	Motor winding short or insulation fails	(1) 100% (2) 0.25 (3) 1 (4) 1.0 (5) 0.25 *Low criticality*

Table 3.16 (continued)

Assembly	Component	Failure description	Failure mode	Failure effect	Failure consequence	Cause of failure	Critical analysis
RJS pump no. 1	Motor RJS pump no. 1	Motor overheats and trips	PFC	Motor failure prevents quenching of the gas and the protection of the RJS structure due to reduced flow. Standby pump should start up automatically	Maintenance	Bearings fail due to lack of or to excessive lubrication	(1) 100% (2) 0.50 (3) 1 (4) 1.0 (5) 0.50 *Low criticality*
RJS pump no. 1	Motor RJS pump no. 1	Motor vibrates excessively	PFC	Motor failure prevents quenching of the gas and the protection of the RJS structure due to reduced flow. Standby pump should start up automatically	Maintenance	Bearings worn or damaged	(1) 100% (2) 0.50 (3) 1 (4) 1.0 (5) 0.50 *Low criticality*

Table 3.17 Extract from FMECA worksheet of quantitative RAM analysis field study: MCC RJS pump no. 1 component

Assembly	Component	Failure description	Failure mode	Failure effect	Failure consequence	Cause of failure	Critical analysis
RJS pump no. 1	MCC RJS pump no. 1	Motor fails to start upon command	TLF	Motor failure starting upon command prevents the standby pump to start up automatically	Maintenance	Electrical supply or starter failure	(1) 100% (2) 0.25 (3) 2 (4) 2.0 (5) 0.50 *Low criticality*
RJS pump no. 1	MCC RJS pump no. 1	Motor fails to start upon command	TLF	Motor failure starting upon command prevents the standby pump to start up automatically	Maintenance	High/low voltage defective fuses or circuit breakers	(1) 100% (2) 0.25 (3) 2 (4) 2.0 (5) 0.50 *Low criticality*
RJS pump no. 1	MCC RJS pump no. 1	Motor fails to start upon command	TLF	Motor failure starting upon command prevents the standby pump to start up automatically	Maintenance	Control system wiring malfunction due to hot spots	(1) 100% (2) 0.25 (3) 2 (4) 2.0 (5) 0.50 *Low criticality*

Table 3.18 Extract from FMECA worksheet of quantitative RAM analysis field study: RJS pump no. 1 control valve component

Assembly	Component	Failure description	Failure mode	Failure effect	Failure consequence	Cause of failure	Critical analysis
RJS pump no. 1	Control valve	Fails to open	TLF	Prevents discharge of acid from the pump that cleans and cools gas and protects the RJS. Flow and pressure protections would prevent damage. May result in downtime if it occurs on standby pump when needed	Production	No PLC output due to modules electronic fault or cabling	(1) 100% (2) 0.50 (3) 6 (4) 6.0 (5) 3.00 *Low/medium criticality*
RJS pump no. 1	Control valve	Fails to open	TLF	Prevents discharge of acid from the pump that cleans and cools gas and protects the RJS. Flow and pressure protections would prevent damage. May result in downtime if it occurs on standby pump when needed	Production	Solenoid valve fails, failed cylinder actuator or air receiver failure	(1) 100% (2) 0.50 (3) 6 (4) 6.0 (5) 3.00 *Low/medium criticality*

Table 3.19 Extract from FMECA worksheet of quantitative RAM analysis field study: RJS pump no. 1 instrument loop (pressure) assembly

Assembly	Component	Failure description	Failure mode	Failure effect	Failure consequence	Cause of failure	Critical analysis
RJS pump no. 1 instrument loop (pressure)	Instrument (pressure. 1)	Fails to provide accurate pressure indication	TLF	Fails to permit pressure monitoring	Maintenance	Restricted sensing port due to blockage by chemical or physical action	(1) 100% (2) 3.00 (3) 2 (4) 2.0 (5) 6.00 *Medium/high criticality*
RJS pump no. 1 instrument loop (pressure)	Instrument (pressure. 2)	Fails to detect low-pressure condition	TLF	Does not permit essential pressure monitoring and can cause damage to the pump due to lack of mechanical seal flushing	Maintenance	Pressure switch fails due to corrosion or relay or cable failure	(1) 100% (2) 0.50 (3) 2 (4) 2.0 (5) 1.00 *Low criticality*
RJS pump no. 1 instrument loop (pressure)	Instrument (pressure. 2)	Fails to provide output signal for alarm condition	TLF	Does not permit essential pressure monitoring and can cause damage to the pump due to lack of mechanical seal flushing	Maintenance	PLC alarm function or indicator fails	(1) 100% (2) 0.30 (3) 2 (4) 2.0 (5) 0.60 *Low criticality*

To introduce uncertainty in this analysis, according to the theory developed for the extended FMECA, the following approach is considered:

- Express the various failure modes, including their (more or less) certain consequences (i.e. the more or less certainty that the consequence can or cannot occur)
- Present the number of uncertainty levels in linguistic terms
- For a given failure mode, sort the occurrence of the consequences into a specific range of $(6+1)$ categories:
  - Three levels of more or less certain consequences ('completely certain', 'almost certain', 'likely')
  - Three levels of more or less impossible consequences ('completely impossible', 'almost impossible', 'unlikely')
  - One level for ignorance.

The approach is thus initiated by expressing the various failure modes, along with their (more or less) *certain* consequences. The discriminability of the failure modes

Table 3.20 Uncertainty in the FMECA of a critical control valve

Component	Failure description	Failure mode	Failure consequence	Failure cause	(1) $\mu_{M(d)+}$	(1) $\mu_{M(d)-}$	Critical analysis
Control valve	Fails to open	TLF	Production	No PLC output due to modules electronic fault or cabling	0.6	0.4	(2) 0.5 (3) 6 (4) 3.6 (or not—2.4) (5) 1.8 (or not—1.2) *Low criticality*
Control valve	Fails to open	TLF	Production	Solenoid valve fails, due to failed cylinder actuator or air receiver failure	0.6	0.4	(2) 0.5 (3) 6 (4) 3.6 (or not—2.4) (5) 1.8 (or not—1.2) *Low criticality*
Control valve	Fails to seal/close	TLF	Production	Valve disk damaged due to corrosion or wear	0.8	0.2	(2) 0.5 (3) 6 (4) 4.8 (or not—1.2) (5) 2.4 (or not—0.6) *Low criticality*
Control valve	Fails to seal/close	TLF	Production	Valve stem cylinders seized due to chemical deposition or corrosion	0.8	0.2	(2) 0.5 (3) 6 (4) 4.8 (or not—1.2) (5) 2.4 (or not—0.6) *Low criticality*

with their (more or less) *certain* consequences is checked. If this is not sufficient, then the question is explored whether some of the (more or less) certain consequences of one failure mode could not be expressed as more or less impossible for some other fault modes. The three categories of more or less impossible consequences are thus indicated whenever necessary, to allow a better discrimination. After this refinement stage, if a set of failure modes still cannot be discriminated in a satisfying way, then the observability of the consequence should be questioned.

b) Results of the Qualitative FMECA

As an example, the critical control valve considered in the FMECA chart of Table 3.18 has been itemised for inclusion in an extended FMECA chart relating to the discriminated failure mode, TLF, along with its (more or less) *certain* conse-

Table 3.21 Uncertainty in the FMECA of critical pressure instruments

Component	Failure description	Failure mode	Failure consequence	Failure cause	(1) $\mu_{M(d)+}$	(1) $\mu_{M(d)-}$	Critical analysis
Instrument (pressure. 1)	Fails to detect low-pressure condition	TLF	Maintenance	Pressure switch fails due to corrosion or relay or cable failure	0.6	0.4	(2) 0.50 (3) 2 (4) 1.2 (or not—0.8) (5) 0.6 (or not—0.4) *Low criticality*
Instrument (pressure. 1)	Fails to provide accurate pressure indication	TLF	Maintenance	Restricted sensing port due to blockage by chemical or physical action	0.8	0.2	(2) 3.00 (3) 2 (4) 1.6 (or not—0.4) (5) 4.8 (or not—1.2) *Medium criticality*
Instrument (pressure. 2)	Fails to detect low-pressure condition	TLF	Maintenance	Pressure switch fails due to corrosion or relay or cable failure	0.6	0.4	(2) 0.50 (3) 2 (4) 1.2 (or not—0.8) (5) 0.6 (or not—0.4) *Low criticality*
Instrument (pressure. 2)	Fails to provide output signal for alarm condition	TLF	Maintenance	PLC alarm function or indicator fails	0.8	0.2	(2) 3.00 (3) 2 (4) 1.6 (or not—0.4) (5) 4.8 (or not—1.2) *Medium criticality*

quences, given in Tables 3.20 and 3.21. To simplify, it is assumed that all the *events* are directly observable—that is, each *effect* is non-ambiguously associated to a *consequence*, although the same consequence can be associated to other effects (i.e. the effects, or events, are equated to their associated consequences, or manifestations). The knowledge expressed in Tables 3.20 and 3.21 describes the fuzzy relation between failure modes, effects and consequences, in terms of the fuzzy sets for the expanded FMECA, $M(d)+(m_i)$ and $M(d)-(m_i)$.

The linguistic qualitative-numeric mapping used for uncertainty representation is tabulated below (Cayrac et al. 1994).

Qualifier	Ref. code	$\mu_{M(d)+}$	$\mu_{M(d)-}$
Certain	1	1.0	0.0
Almost certain	2	0.8	0.2
Likely	3	0.6	0.4
Unlikely	4	0.4	0.6
Almost unlikely	5	0.2	0.8
Impossible	6	0.0	1.0
Unknown	7	0.0	0.0

The 'critical analysis' column of the extended FMECA chart relating to the discriminated failure mode, along with its (more or less) *certain* consequences, includes items numbered 1 to 5 that indicate the following:

(1) Possibility of occurrence of a consequence ($\mu_{M(d)+}$) or impossibility of occurrence of a consequence ($\mu_{M(d)-}$)
(2) Estimated failure rate (the number of failures per year)
(3) Severity (expressed as a number from 0 to 10)
(4) Risk (product of 1 and 3)
(5) Criticality value (product of 2 and 4).

3.3.3 Analytic Development of Reliability Evaluation in Detail Design

The most applicable methods selected for further development as tools for *reliability evaluation* in determining the integrity of engineering design in the *detail design* phase are:

i. *The proportional hazards model* (or instantaneous failure rate, indicating the probability of survival of a component);
ii. *Expansion of the exponential failure distribution* (considering component functional failures that occur at random intervals);
iii. *Expansion of the Weibull failure distribution* (to determine component criticality for wear-out failures, not random failures);
iv. *Qualitative analysis of the Weibull distribution model* (when the Weibull parameters cannot be based on obtained data).

3.3.3.1 The Proportional Hazards Model

The proportional hazards (PH) model was developed in order to estimate the effects of different covariates influencing the times to failure of a system (Cox 1972). In its original form, the model is non-parametric, i.e. no assumptions are made about the nature or shape of the underlying failure distribution. The original non-parametric formulation as well as a parametric form of the model are considered, utilising the Weibull life distribution. Special developments of the proportional hazards model are:

General log-linear, GLL—exponential
General log-linear, GLL—Weibull models.

a) Non-Parametric Model Formulation

From the PH model, the failure rate of a system is affected not only by its operating time but also by the *covariates* under which it operates. For example, a unit of equipment may have been tested under a combination of different accelerated stresses such as humidity, temperature, voltage, etc. These factors can affect the failure rate of the unit, and typically represent the type of stresses that the unit will be subject to, once installed.

The instantaneous failure rate (or hazard rate) of a unit is given by the following relationship

$$\lambda(t) = \frac{f(t)}{R(t)}\,, \tag{3.144}$$

where:
$f(t)$ = the probability density function,
$R(t)$ = the reliability function.

For the specific case where the failure rate of a particular unit is dependent not only on time but also on other covariates, Eq. (3.144) must be modified in order to be a function of time *and* of the covariates. The proportional hazards model assumes that the failure rate (hazard rate) of a unit is the product of the following factors:

- An unspecified baseline failure rate, $\lambda_o(t)$, which is a function of time only,
- A positive function $g(x, \underline{\mathbf{A}})$ that is independent of time, and that incorporates the effects of a number of covariates such as humidity, temperature, pressure, voltage, etc.

The failure rate of the unit is then given by

$$\lambda(t, \underline{\mathbf{X}}) = \lambda_o(t) \cdot g(\underline{\mathbf{X}}, \underline{\mathbf{A}})\,, \tag{3.145}$$

where:
$\underline{\mathbf{X}}$ = a row vector consisting of the covariates,
$\underline{\mathbf{X}} = (x_1, x_2, x_3, \ldots, x_m)$

$\underline{\mathbf{A}}$ = a column vector consisting of the unknown model parameters (regression parameters),
$\underline{\mathbf{A}} = (a_1, a_2, a_3, \ldots, a_m)^{\mathrm{T}}$
m = number of stress-related variates (time-independent).

It can be assumed that the form of $g(\underline{\mathbf{X}}, \underline{\mathbf{A}})$ is known and $\lambda_{\mathrm{o}}(t)$ is unspecified. Different forms of $g(\underline{\mathbf{X}}, \underline{\mathbf{A}})$ can be used but the *exponential* form is mostly used, due to its simplicity.

The *exponential* form of $g(\underline{\mathbf{X}}, \underline{\mathbf{A}})$ is given by the following expression

$$g(\underline{\mathbf{X}}, \underline{\mathbf{A}}) = \mathrm{e}^{\underline{\mathbf{A}}^{\mathrm{T}} \underline{\mathbf{X}}^{\mathrm{T}}} = \exp\left[\sum_{j=1}^{m} a_j x_j\right] , \tag{3.146}$$

where:
a_j = model parameters (regression parameters),
x_j = covariates.

The failure rate can then be written as

$$\lambda(t, \underline{\mathbf{X}}) = \lambda_{\mathrm{o}} \cdot \exp\left[\sum_{j=1}^{m} a_j x_j\right] . \tag{3.147}$$

b) Parametric Model Formulation

A parametric form of the proportional hazards model can be obtained by assuming an underlying distribution. In general, the exponential and the Weibull distributions are the easiest to use. The lognormal distribution can be utilised as well but it is not considered here. In this case, the Weibull distribution will be used to formulate the parametric proportional hazards model. The exponential distribution case can be easily obtained from the Weibull equations, by simply setting the Weibull shape parameter $\beta = 1$. In other words, it is assumed that the baseline failure rate is parametric and given by the Weibull distribution. The baseline failure rate is given by the following expression taken from Eq. (3.37):

$$\lambda_{\mathrm{o}} = \frac{\beta (t)^{\beta - 1}}{\mu^{\beta}} ,$$

where:
μ = the scale parameter,
β = the shape parameter.

Note that μ is the baseline Weibull scale parameter but not the PH scale parameter. The PH failure rate then becomes

$$\lambda(t, \underline{\mathbf{X}}) = \frac{\beta (t)^{\beta - 1}}{\mu^{\beta}} \exp\left[\sum_{j=1}^{m} a_j x_j\right] , \tag{3.148}$$

where:

a_j and x_j = regression parameters and covariates,
β and μ = the shape and scale parameters.

It is often more convenient to define an additional covariate, $x_o = 1$, in order to allow the Weibull scale parameter to be included in the vector of regression coefficients, and the proportional hazards model expressed solely by the beta (shape parameter), together with the regression parameters and covariates. The PH failure rate can then be written as

$$\lambda(t, \underline{\mathbf{X}}) = \beta(t)^{\beta-1} \exp\left[\sum_{j=0}^{m} a_j x_j\right] . \tag{3.149}$$

The PH reliability function is thus given by the expression

$$\begin{aligned} R(t, \underline{\mathbf{X}}) &= \exp\left[-\int_0^t \lambda(u)\,\mathrm{d}u\right] \\ R(t, \underline{\mathbf{X}}) &= \exp\left[-\int_0^t \lambda(u, \underline{\mathbf{X}})\,\mathrm{d}u\right] \\ R(t, \underline{\mathbf{X}}) &= \exp\left[-t^{\beta} \cdot \exp\left[\sum_{j=0}^{m} a_j x_j\right]\right] \end{aligned} \tag{3.150}$$

The probability density function (p.d.f.) can be obtained by taking the partial derivative with respect to time of the reliability function given by Eq. (3.150). The PH probability density function is given by the expression $f(t, \underline{\mathbf{X}}) = \lambda(t, \underline{\mathbf{X}}) R(t, \underline{\mathbf{X}})$. The total number of unknowns to solve in this model is $m+2$ (i.e. $\beta, \mu, a_1, a_2, a_3, \ldots, a_m$).

The *maximum likelihood estimation* method can be used to determine these parameters. Solving for the parameters that maximise the maximum likelihood estimation will yield the parameters for the PH Weibull model. For $\beta = 1$, the equation then becomes the likelihood function for the PH exponential model, which is similar to the original form of the proportional hazards model proposed by Cox (1972).

c) Maximum Likelihood Estimation (MLE) Parameter Estimation

The idea behind maximum likelihood parameter estimation is to determine the parameters that maximise the probability (likelihood) of the sample data. From a statistical point of view, the method of maximum likelihood is considered to be more robust (with some exceptions) and yields estimators with good statistical properties. In other words, MLE methods are versatile and apply to most models and to different types of data. In addition, they provide efficient methods for quantifying uncertainty through confidence bounds. Although the methodology for maximum likelihood estimation is simple, the implementation is mathematically complex. By utilising computerised models, however, the mathematical complexity of MLE is not an obstacle.

Asymptotic behaviour In many cases, estimation is performed using a set of independent, identically distributed measurements. In such cases, it is of interest to determine the behaviour of a given estimator as the set of measurements increases to infinity, referred to as *asymptotic behaviour*. Under certain conditions, the MLE exhibits several characteristics that can be interpreted to mean it is 'asymptotically optimal'. While these asymptotic properties become strictly true only in the limit of infinite sample size, in practice they are often assumed to be approximately true, especially with a large sample size. In particular, inference about the estimated parameters is often based on the asymptotic Gaussian distribution of the MLE.

As MLE can generally be applied to failure-related sample data that are available for critical components during the *detail design phase* of the engineering design process, it is necessary to examine more closely the theory that underlies maximum likelihood estimation for the quantification of *complete data*. Alternately, when no data are available, the method of *qualitative parameter estimation* becomes essential, as considered in detail later in Section 3.3.3.3.

Background theory If x is a continuous random variable with probability density function:

$$f(x;\theta_1,\theta_2,\theta_3,\ldots,\theta_k)\,,$$

where:

$\theta_1,\theta_2,\theta_3,\ldots,\theta_k$ are k unknown and constant parameters that need to be estimated through n independent observations, $x_1,x_2,x_3,\ldots,x_n$.

Then, the likelihood function is given by the following expression

$$L(x_1,x_2,x_3,\ldots,x_n)=\prod_{i=1}^{n} f(x_i;\theta_1,\theta_2,\theta_3,\ldots,\theta_k) \quad i=1,2,3,\ldots,n\,. \tag{3.151}$$

The logarithmic likelihood function is given by

$$\Lambda=\ln L=\sum_{i=1}^{n}\ln f(xi;\theta_1,\theta_2,\theta_3,\ldots,\theta_k)\,. \tag{3.152}$$

The maximum likelihood estimators (MLE) of $\theta_1,\theta_2,\theta_3,\ldots,\theta_k$ are obtained by maximising Λ. By maximising Λ, which is much easier to work with than L, the maximum likelihood estimators (MLE) of the range $\theta_1,\theta_2,\theta_3,\ldots,\theta_k$ are the simultaneous solutions of k equations where the partial derivatives of Λ are equal to zero:

$$\frac{\partial(\Lambda)}{\partial\theta_j}=0 \quad j=1,2,3,\ldots,k\,.$$

Even though it is common practice to plot the MLE solutions using *median ranks* (points are plotted according to median ranks and the line according to the MLE solutions), this method is not completely accurate. As can be seen from the equations above, the MLE method is independent of any kind of ranks or plotting methods. For this reason, the MLE solution appears many times not to track the data on a prob-

ability plot. This is perfectly acceptable, since the two methods are independent of each other.

Illustrating the MLE Method Using the Exponential Distribution:

To estimate λ, for a sample of n units (all tested to failure), the likelihood function is obtained

$$\begin{aligned} L(\lambda|t_1,t_2,t_3,\ldots,t_n) &= \prod_{i=1}^{n} f(t_i) \\ &= \prod_{i=1}^{n} \lambda\, \mathrm{e}^{-\lambda t_i} \\ &= \lambda^n \mathrm{e}^{-\lambda \Sigma t_i} \end{aligned} \tag{3.153}$$

Taking the natural log of both sides

$$\Lambda = \ln(L) = n\ln(\lambda) - \lambda \sum_{i=1}^{n} t_i$$

$$\frac{\partial(\Lambda)}{\partial\lambda} = \frac{n}{\lambda} - \sum_{i=1}^{n} t_i = 0$$

Solving for λ gives:

$$\lambda = n/\sum_{i=1}^{n} t_i\,. \tag{3.154}$$

Notes on Lambda

The value of λ is an *estimate* because, if another sample from the same population is obtained and λ re-estimated, then the new value would differ from the one previously calculated.

How close is the value of the estimate to the true value? To answer this question, one must first determine the distribution of the parameter λ. This methodology introduces another term, the *confidence level*, which allows for the specification of a *range* for the estimate with a certain confidence level. The treatment of confidence intervals is integral to reliability engineering, and to statistics in general.

Illustrating the MLE Method Using the Normal Distribution

To obtain the MLE estimates for the mean, $\mathcal{T}$, and standard deviation, σ_T, for the normal distribution, the probability density function of the normal distribution is

given by

$$F(T) = \frac{1}{\sigma_T\sqrt{2\pi}} \exp\left[-\frac{\frac{1}{2}(T-\mathcal{T})^2}{\sigma_T}\right], \tag{3.155}$$

where:

$\mathcal{T}$ = mean of the normal distribution,
σ_T = standard deviation of the normal distribution.

If $T_1, T_2, T_3, \ldots, T_n$ are *known* times to failure (and with no suspensions), then the likelihood function is given by

$$\begin{aligned} L(T_1, T_2, T_3, \ldots, T_n|\mathcal{T}, \sigma_T): \\ L &= \prod_{i=1}^{n}\left\{\frac{1}{\sigma_T\sqrt{2\pi}} \exp\left[-\frac{\frac{1}{2}(T-\mathcal{T})^2}{\sigma_T}\right]\right\} \\ L &= \frac{1}{\left(\sigma_T\sqrt{2\pi}\right)^n} \exp\left[-\frac{1}{2}\sum_{i=1}^{n}\frac{(T_i-\mathcal{T})^2}{\sigma_T}\right] \\ \Lambda &= \ln(L): \\ \ln(L) &= -\frac{n}{2}\ln(2\pi) - n\ln\sigma_T - \frac{1}{2}\sum_{i=1}^{n}\frac{(T_i-\mathcal{T})^2}{\sigma_T} \end{aligned} \tag{3.156}$$

Then, taking the partial derivatives of Λ with respect to each one of the parameters, and setting these equal to zero yields:

$$\frac{\partial(\Lambda)}{\partial\mathcal{T}} = \frac{1}{\sigma_T^2} - \sum_{i=1}^{n}(T_i-\mathcal{T}) = 0$$

and:

$$\frac{\partial(\Lambda)}{\partial\sigma_T} = \frac{n}{\sigma_T} + \frac{1}{\sigma_T^3}\sum_{i=1}^{n}(T_i-\mathcal{T})^2 = 0\,.$$

Solving these equations simultaneously yields

$$\mathcal{T} = \frac{1}{n}\sum_{i=1}^{n}T_i \tag{3.157}$$

$$\sigma_T^2 = \frac{1}{n}\sum_{i=1}^{n}(T_i-\mathcal{T})^2 \tag{3.158}$$

These solutions are valid only for data with no suspensions, i.e. all units are tested to failure. In cases in which suspensions are present, the methodology changes and the problem becomes much more complicated.

Estimator As indicated, the parameters obtained from maximising the likelihood function are estimators of the true value. It is clear that the sample size determines the accuracy of an estimator. If the sample size equals the whole population, then the

estimator *is* the true value. Estimators have properties such as non-bias and consistency (as well as properties of sufficiency and efficiency, which are not considered here).

Unbiased estimator An estimator given by the relationship $\theta = d(x_1, x_2, x_3, \ldots, x_n)$ is considered to be unbiased *if and only if* the estimator satisfies the condition $E(\theta) = \theta$ for all θ. In this case, $E(x)$ denotes the expected value of x and is defined by the following expression for continuous distributions

$$E(x) = \int_{\psi} x f(x)\,\mathrm{d}x \quad x \in \psi\,. \tag{3.159}$$

This implies that the true value is not consistently underestimated nor overestimated.

Consistent estimator An unbiased estimator that converges more closely to the true value as the sample size increases is called a *consistent* estimator. The standard deviation of the normal distribution was obtained using MLE. However, this estimator of the true standard deviation is a biased one. It can be shown that the consistent estimate of the variance and standard deviation for complete data (for the normal distribution) is given by

$$\sigma_T^2 = \frac{1}{n-1}\sum_{i=1}^{n}(T_i - \bar{T})^2\,. \tag{3.160}$$

Analysis of censored data So far, parameter estimation has been considered for complete data only. Further expansion on the maximum likelihood parameter estimation method needs to include estimating parameters with right censored data. The method is based on the same principles covered previously, but modified to take into account the fact that some of the data are censored.

MLE analysis of right censored data The maximum likelihood method is by far the most appropriate analysis method for censored data. When performing maximum likelihood analysis, the likelihood function needs to be expanded to take into account the suspended items. A great advantage of using MLE when dealing with censored data is that each suspension term is included in the likelihood function. Thus, the estimates of the parameters are obtained from consideration of the entire sample population of tested components. Using MLE properties, confidence bounds can be obtained that also account for all the suspension terms. In the case of suspensions, and where x is a continuous random variable with p.d.f. and c.d.f. of the following forms

$$f(x; \theta_1, \theta_2, \theta_3, \ldots, \theta_k)$$
$$F(x; \theta_1, \theta_2, \theta_3, \ldots, \theta_k)$$

$\theta_1, \theta_2, \theta_3, \ldots, \theta_k$ are the k unknown parameters that need to be estimated from R *failures* at $(T_1, V_{T_1}), (T_2, V_{T_2}), (T_3, V_{T_3}), \ldots, (T_R, V_{T_R})$, and from M *suspensions* at $(S_1, V_{S_1}), (S_2, V_{S_2}), (S_3, V_{S_3}), \ldots, (S_M, V_{S_M})$, where V_{T_R} is the Rth stress level corresponding to the Rth observed failure, and V_{S_M} the Mth stress level corresponding to the Mth observed suspension.

The likelihood function is then formulated, and the parameters solved by maximising

$$L\left((T_1,V_{T_1}),\ldots,(T_R,V_{T_R}),(S_1,V_{S_1}),\ldots,(S_M,V_{S_M})|\theta_1,\theta_2,\theta_3,\ldots,\theta_k\right) = \prod_{i=1}^{R} f(T_i,V_{T_i};\theta_1,\theta_2,\theta_3,\ldots,\theta_k)\prod_{j=1}^{M}\left[1-F(S_j,V_{S_j};\theta_1,\theta_2,\theta_3,\ldots,\theta_k)\right] . \quad (3.161)$$

3.3.3.2 Expansion of the Exponential Failure Distribution

Estimating failure rate As indicated previously in Section 3.2.3.2, the exponential distribution is a very commonly used distribution in reliability engineering. Due to its simplicity, it has been widely employed in *designing for reliability*. The exponential distribution describes components with a single parameter, the *constant failure rate*. The single-parameter *exponential probability density function* is given by

$$f(T) = \lambda\, \mathrm{e}^{-\lambda T} = (1/\mathrm{MTBF})\, \mathrm{e}^{-T/\mathrm{MTBF}} . \quad (3.162)$$

This distribution requires the estimation of only one parameter, λ, for its application in designing for reliability, where:

λ = constant failure rate,
$\lambda > 0$,
$\lambda = 1/\mathrm{MTBF}$,
MTBF = mean time between failures, or to a failure,
MTBF > 0,
T = operating time, life or age, in hours, cycles, etc.
$T \geq 0$.

There are several methods for estimating λ in the single-parameter exponential failure distribution. In designing for reliability, however, it is important to first understand some of its statistical properties.

a) Characteristics of the One-Parameter Exponential Distribution

The statistical characteristics of the one-parameter exponential distribution are better understood by examining its parameter, λ, and the effect that this parameter has on the exponential probability density function as well as the reliability function.

Effects of λ on the probability density function:

- The *scale parameter* is $1/\lambda = m$. The only parameter it has is the failure rate, λ.
- As λ is decreased in value, the distribution is stretched to the right.
- This distribution has no *shape parameter* because it has only one shape, i.e. the exponential.
- The distribution starts at $T = 0$ where $f(T = 0) = \lambda$ and decreases exponentially as T increases (Fig. 3.34), and is convex as $T \to \infty$, $f(T) \to 0$.

- This probability density function (p.d.f.) can be thought of as a special case of the Weibull probability density function with $\beta = 1$.

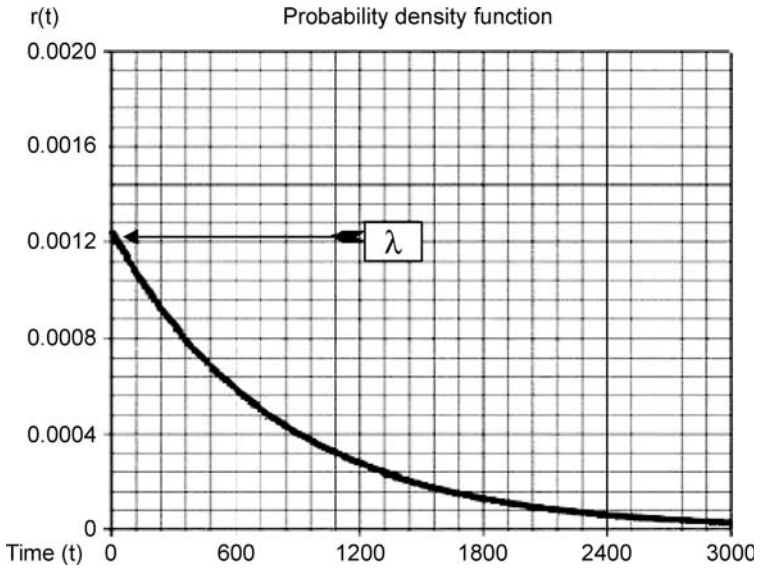


Fig. 3.34 Effects of λ on the probability density function

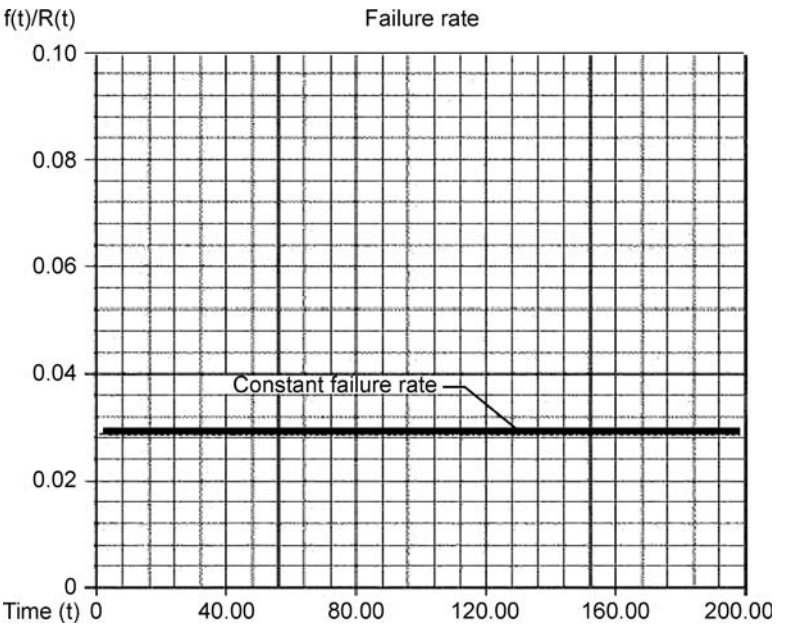


Fig. 3.35 Effects of λ on the reliability function

Effects of λ on the reliability function:

- The failure rate of the function is represented by the parameter λ.
- The failure rate of the reliability function is constant (Fig. 3.35).
- The one-parameter exponential reliability function starts at the value of 1 at $T = 0$.
- As $T \to \infty$, $R(T) \to 0$.

b) Estimating the Parameter of the Exponential Distribution

The parameter of the exponential distribution can be estimated graphically by probability plotting or analytically by either *least squares* or *maximum likelihood*.

Probability plotting The graphical method of estimating the parameter of the exponential distribution is by probability plotting, illustrated in the following example.

Estimating the parameter of the exponential distribution with probability plotting Assume six identical units have pilot reliability test results at the same application and operation stress levels. All of these units appear to have failed after operating for the following testing periods, measured in hours: 96, 257, 498, 763, 1,051 and 1,744. Steps for estimating the parameter of the exponential probability density function, using probability plotting, are as follows (Table 3.22).

The times to failure are sorted from small to large values, and median rank percentages calculated. Median rank positions are used instead of other ranking methods because median ranks are at a specific confidence level (50%). Exponential probability plots use scalar data arranged in rank order for the *x*-axis of the probability plot. The *y*-axis plot is found from a statistical technique, Benard's median rank position (Abernethy 1992).

Determining the X and Y positions of the plot points The points plotted represent times-to-failure data in reliability analysis. For example, the times to failure in Table 3.22 would be used as the *x* values or time values. Determining what the appropriate *y* plot position, or the *unreliability* values should be is a little more complex. To determine the *y* plot positions, a value indicating the corresponding

Table 3.22 Median rank table for failure test results

Time to failure (h)	Failure order number	Median rank (%)
96	1	10.91
257	2	26.44
498	3	42.14
763	4	57.86
1,051	5	73.56
1,744	6	89.10

unreliability for that failure must first be determined. In other words, the cumulative percent failed must be obtained for each time to failure. In the example, the cumulative percent failed by 96 h is 17%, by 257 h 34% and so forth. This is a simple method illustrating the concept. The problem with this method is that the 100% point is not defined on most probability plots. Thus, an alternative and more robust approach must be used, such as the method of obtaining the *median rank* for each failure.

Method of median ranks Median ranks are used to obtain an estimate of the unreliability, $U(T_j)$, for each failure. It is the value that the true probability of failure, $Q(T_j)$, should have at the jth failure out of a sample of N components, at a 50% confidence level. This essentially means that this is a *best estimate* for the unreliability: half of the time the true value will be greater than the 50% confidence estimate, while the other half of the time the true value will be less than the estimate. The estimate is then based on a solution of the binomial distribution.

The rank can be found for any percentage point, P, greater than zero and less than one, by solving the cumulative binomial distribution for Z. This represents the rank, or unreliability estimate, for the jth failure in the following equation for the cumulative binomial distribution

$$P = \sum_{k=j}^{N} (N_k) Z^k (1-Z)^{N-k} , \tag{3.163}$$

where:

N = the sample size,
j = the order number.

The median rank is obtained by solving for Z at $P = 0.50$ in

$$0.50 = \sum_{k=j}^{N} (N_k) Z^k (1-Z)^{N-k} . \tag{3.164}$$

For example, if $N = 6$ and we have six failures, then the median rank equation would be solved six times, once for each failure with $j = 1, 2, 3, 4, 5$ and 6, for the value of Z. This result can then be used as the *unreliability estimate* for each failure, or the y plotting position. The solution of Eq. (3.164) for Z requires the use of numerical methods. A quick though less accurate approximation of the median ranks is given by the following expression. This approximation of the median ranks is known as Benard's approximation (Abernethy 1992):

$$MR = \frac{j - 0.3}{N + 0.4} . \tag{3.165}$$

For the six failures in Table 3.22, the following values are equated (Table 3.23):

Table 3.23 Median rank table for Bernard's approximation

Failure order number	Bernard's approximation ($\times 10^{-2}$)	Binomial equation	Error margin
Failure 1	$MR_1 = 0.7/6.4 = 10.94$	10.91	+0.275%
Failure 2	$MR_2 = 1.7/6.4 = 26.56$	26.44	+0.454%
Failure 3	$MR_3 = 2.7/6.4 = 42.19$	42.14	+0.120%
Failure 4	$MR_4 = 3.7/6.4 = 57.81$	57.86	−0.086%
Failure 5	$MR_5 = 4.7/6.4 = 73.44$	73.56	−0.163%

Kaplan–Meier estimator The Kaplan–Meier estimator is used as an alternative to the median ranks method for calculating the estimates of the unreliability for probability plotting purposes

$$F(t_i) = 1 - \prod_{j=1}^{i} \frac{n_j - r_j}{n_j} \,, \tag{3.166}$$

where:

i $= 1,2,3,\ldots,m$,
m = total number of data points,
n = total number of units.

and:

$$n_i = \sum_{j=0}^{i-1} S_j - \sum_{j=0}^{i-1} R_j \,,$$

where:

i $= 1,2,3,\ldots,m$,
R_j = number of failures in the jth data group,
S_j = number of surviving units in the jth data group.

The exponential probability graph is based on a log-linear scale, as illustrated in Fig. 3.36. The best possible straight line is drawn that goes through the $t = 0$ and $R(t) = 100\%$ point, and through the plotted points on the x-axis and their corresponding rank values on the y-axis. A horizontal line is drawn at the ordinate point $Q(t) = 63.2\%$ or at the point $R(t) = 36.8\%$, until this line intersects the fitted straight line. A vertical line is then drawn through this intersection until it crosses the abscissa. The value at the abscissa is the estimate of the mean.

For this example, MTBF = 833 h, which means that $\lambda = 1/\text{MTBF} = 0.0012$. This is always at 63.2%, since $Q(T) = 1 - \mathrm{e}^{-1} = 63.2\%$.

The reliability value for any mission or operational time t can be obtained. For example, the reliability for an operational duration of 1,200 h can now be obtained. To obtain the value from the plot, a vertical line is drawn from the abscissa, at $t = 1{,}200$ h, to the fitted line. A horizontal line from this intersection to the ordinate is drawn and $R(t)$ obtained. This value can also be obtained analytically from the exponential reliability function. In this case, $R(t) = 98.15\%$ where $R(t) = 1 - U$ and $U = 1.85\%$ at $t = 1{,}200$.

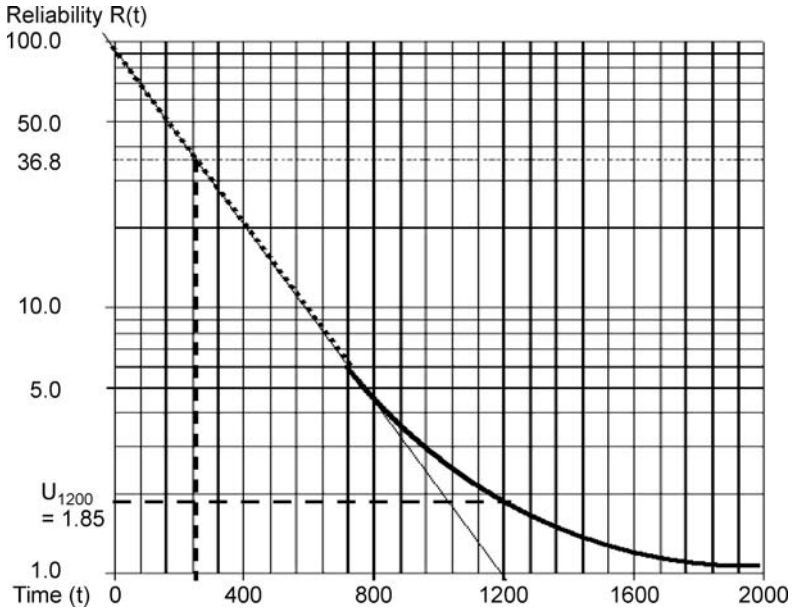


Fig. 3.36 Example exponential probability graph

c) Determining the Maximum Likelihood Estimation Parameter

The parameter of the exponential distribution can also be estimated using the *maximum likelihood estimation (MLE)* method. This function is log-likelihood and composed of two summation portions

$$\Lambda = \ln(L) = \sum_{i=1}^{F} N_i \ln\left[\lambda\, \mathrm{e}^{-\lambda T_i}\right] - \sum_{i=1}^{S} \check{N}_i \lambda \check{T}_i \,, \tag{3.167}$$

where:

F is the number of groups of times-to-failure data points.
N_i is the number of times to failure in the ith time-to-failure data group.
λ is the failure rate parameter (unknown a priori, only one to be found).
T_i is the time of the ith group of time-to-failure data.
S is the number of groups of suspension data points.
$\check{N}_i$ is the number of suspensions in the ith group of data points.
$\check{T}_i$ is the time of the ith suspension data group.

The solution will be found by solving for a parameter λ, so that

$$\frac{\partial(\Lambda)}{\partial\lambda} = 0 \quad \text{and} \quad \frac{\partial(\Lambda)}{\partial\lambda} = \sum_{i=1}^{F} N_i \left[\frac{1}{\lambda} - T_i\right] - \sum_{i=1}^{S} \check{N}_i \check{T}_i \,, \tag{3.168}$$

where also:

F is the number of groups of times-to-failure data points.
N_i is the number of times to failure in the ith time-to-failure data group.
λ is the failure rate parameter (unknown a priori, only one to be found).
T_i is the time of the ith group of time-to-failure data.
S is the number of groups of suspension data points.
$\breve{N}_i$ is the number of suspensions in the ith group of data points.
$\breve{T}_i$ is the time of the ith suspension data group.

3.3.3.3 Expansion of the Weibull Distribution Model

a) Characteristics of the Two-Parameter Weibull Distribution

The characteristics of the two-parameter Weibull distribution can be exemplified by examining the two parameters β and μ, and the effect they have on the Weibull probability density function, reliability function and failure rate function. Changing the value of β, the shape parameter or slope of the Weibull distribution changes the shape of the probability density function (p.d.f.), as shown in Tables 3.15 to 3.19. In addition, when the cumulative distribution function (c.d.f.) is plotted, as shown in Tables 3.20 and 3.21, a change in β results in a change in the slope of the distribution.

Effects of β on the Weibull p.d.f. The parameter β is dimensionless, with the following effects on the Weibull p.d.f.

- For $0 < \beta < 1$, the failure rate decreases with time and:

$$\text{As } T \to 0\,, \quad f(T) \to \infty\,.$$
$$\text{As } T \to \infty\,, \quad f(T) \to 0\,.$$

  $f(T)$ decreases monotonically and is convex as T increases.
  The mode $\mathring{u}$ is non-existent.
- For $\beta = 1$, it becomes the exponential distribution, as a special case, with:

$$f(T) = 1/\mu\, \mathrm{e}^{-T/\mu} \quad \text{for} \quad \mu > 0\,,\ T \geq 0$$
$$1/\mu = \lambda \quad \text{the chance, useful life, or failure rate.}$$

- For $\beta > 1$, $f(T)$ assumes wear-out type shapes, i.e. the failure rate increases with time:

$$f(T) = 0 \text{ at } T = 0\,.$$

  $f(T)$ increases as $T \to \mathring{u}$ (mode) and decreases thereafter.
- For $\beta = 2$, the Weibull p.d.f. becomes the *Rayleigh distribution*.
- For $\beta < 2.6$, the Weibull p.d.f. is positively skewed.
- For $2.6 < \beta < 3.7$, its coefficient of skewness approaches zero (no tail), and approximates the normal p.d.f.

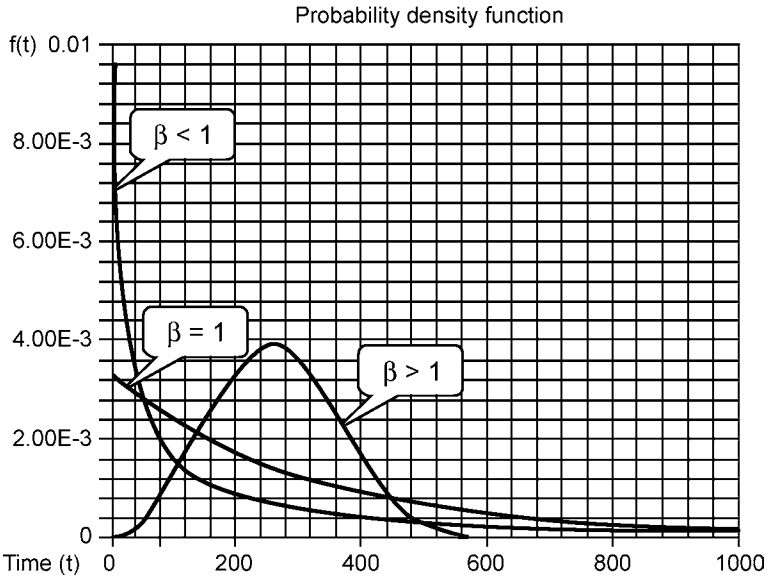


Fig. 3.37 Weibull p.d.f. with $0 < \beta < 1$, $\beta = 1$, $\beta > 1$ and a fixed μ (ReliaSoft Corp.)

- For $\beta > 3.7$, the Weibull p.d.f. is negatively skewed.

From Fig. 3.37:

- For $0 < \beta < 1$: $T \to 0$, $f(T) \to \infty$. $T \to \infty$, $f(T) \to 0$.
- For $\beta = 1$: $f(T) = 1/\mu\, \mathrm{e}^{-T/\mu}$. $T \to \infty$, $f(T) \to 0$.
- For $\beta > 1$: $f(T) = 0$ at $T = 0$. $T \to \mathring{u}$, $f(T) > 0$.

Effects of β on the Weibull reliability function and the c.d.f. Considering first the Weibull *unreliability function* (Fig. 3.38), or cumulative distribution function, $F(t)$, the following effects of β are observed:

- For $0 < \beta < 1$ and constant μ, $F(T)$ is linear with minimum slope and values of $F(T)$ ranging from 5 to below 90.00.
- For $\beta = 1$ and constant μ, $F(T)$ is linear with a steeper slope and values of $F(T)$ ranging from less than 1 to above 90.00.
- For $\beta > 1$ and constant μ, $F(T)$ is linear with maximum slope and values of $F(T)$ ranging from well below 1 to well above 99.90.

Considering the Weibull *reliability function* (Fig. 3.39), or one minus the cumulative distribution function, $1 - F(t)$, the following effects of β are observed:

- For $0 < \beta < 1$ and constant μ, $R(T)$ is convex, and decreases sharply and monotonically.
- For $\beta = 1$ and constant μ, $R(T)$ is convex, and decreases monotonically but less sharply.

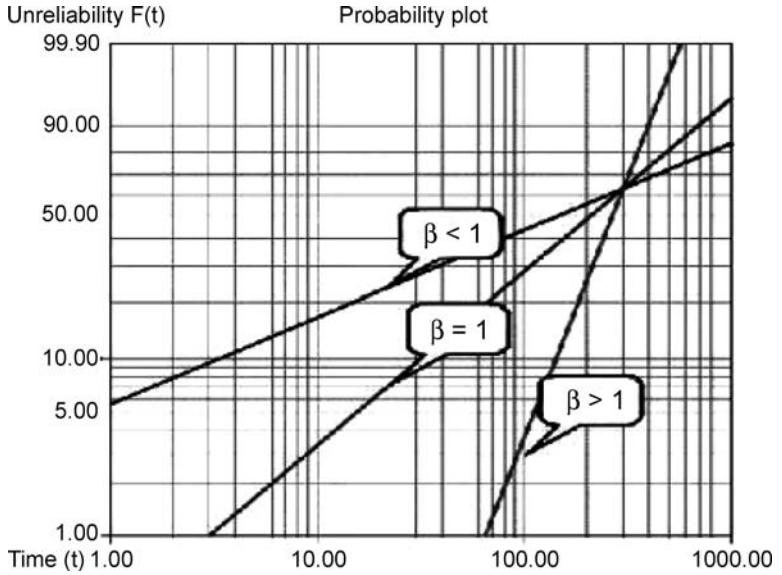


Fig. 3.38 Weibull c.d.f. or unreliability vs. time (ReliaSoft Corp.)

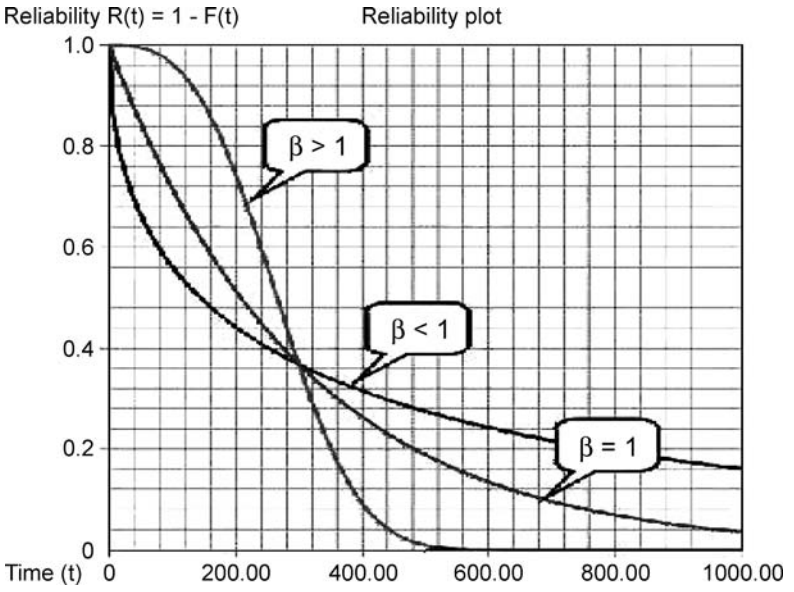


Fig. 3.39 Weibull 1–c.d.f. or reliability vs. time (ReliaSoft Corp.)

- For $\beta > 1$ and constant μ, $R(T)$ decreases as T increases but less sharply than before and, as wear-out sets in, it decreases sharply and goes through an inflection point.

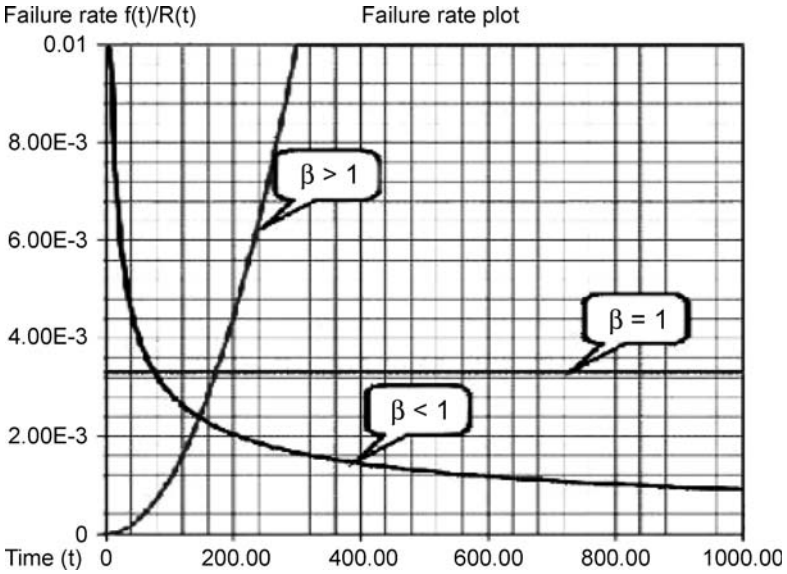


Fig. 3.40 Weibull failure rate vs. time (ReliaSoft Corp.)

Effects of β on the Weibull failure rate function The Weibull failure rate for $0 < \beta < 1$ is unbounded at $T = 0$. The failure rate $\lambda(T)$ decreases thereafter monotonically and is convex, approaching the value of zero as $T \to 0$ or $\lambda(\infty) = 0$. This behaviour makes it suitable for representing the failure rates of components that exhibit early-type failures, for which the failure rate *decreases* with age (Fig. 3.40).

When such behaviour is encountered in pilot tests, the following conclusions may be drawn:

- Burn-in testing and/or environmental stress screening are not well implemented.
- There are problems in the process line, affecting the expected life of the component.
- Inadequate quality control of component manufacture is bringing about early failure.

Effects of β on the Weibull failure rate function and derived failure characteristics The effects of β on the hazard or failure rate function of the Weibull distribution result in several observations and conclusions about the *characteristics of failure*:

- When $\beta = 1$, the hazard rate $\lambda(T)$ yields a *constant* value of $1/\mu$ where: $\lambda(T) = \lambda = 1/\mu$.
  This parameter becomes suitable for representing the hazard or failure rate of chance-type or random failures, as well as the useful life period of the component.

- When $\beta > 1$, the hazard rate $\lambda(T)$ *increases* as T increases, and becomes suitable for representing the failure rate of components with wear-out type failures.
- For $1 < \beta < 2$, the $\lambda(T)$ curve is concave. Consequently, the failure rate *increases* at a *decreasing* rate as T increases.
- For $\beta = 2$, the $\lambda(T)$ curve represents the Rayleigh distribution where: $\lambda(T) = 2/\mu(T/\mu)$.
  There emerges a straight-line relationship between $\lambda(T)$ and T, starting with a failure rate value of $\lambda(T) = 0$ at $T = 0$, and increasing thereafter with a slope of $2/\mu^2$. Thus, the failure rate *increases at a constant rate* as T increases.
- When $\beta > 2$, the $\lambda(T)$ curve is convex, with its slope increasing as T increases. Consequently, the failure rate *increases at an increasing rate* as T increases, indicating component wear-out.

The scale parameter μ A change in the Weibull *scale parameter* μ has the same effect on the distribution (Fig. 3.41) as a change of the abscissa scale:

- If μ is *increased* while β is kept the same, the distribution gets stretched out to the right and its height decreases, while maintaining its shape and location.
- If μ is *decreased* while β is kept the same, the distribution gets pushed in towards the left (i.e. towards 0) and its height increases.

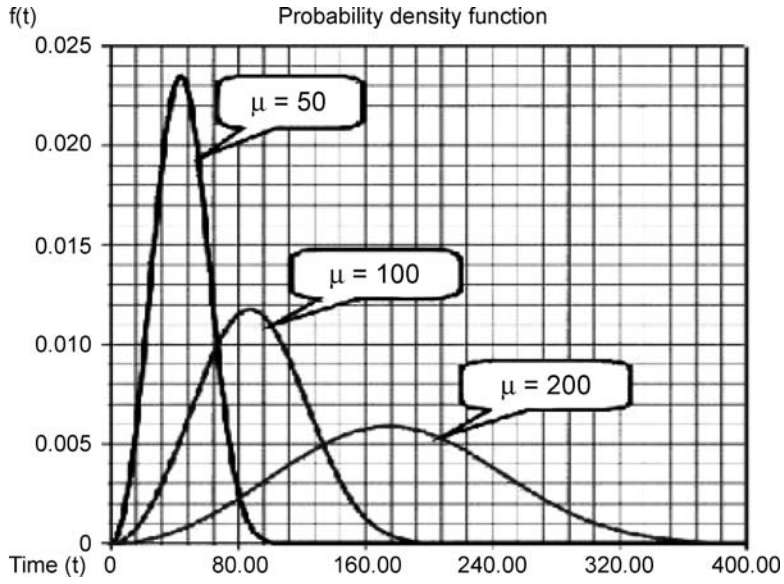


Fig. 3.41 Weibull p.d.f. with $\mu = 50$, $\mu = 100$, $\mu = 200$ (ReliaSoft Corp.)

b) The Three-Parameter Weibull Model

The mathematical model for reliability of the Weibull distribution has so far been determined from a *two-parameter* Weibull distribution formula, where the *two parameters* are β and μ. The mathematical model for reliability of the Weibull distribution can also be determined from a *three-parameter* Weibull distribution formula, where the *three parameters* are:

β = shape parameter or *failure pattern*
μ = scale parameter or *characteristic life*
γ = location, position or *minimum life parameter*.

This reliability model is given as

$$R(t) = \mathrm{e}^{-[(t-\gamma)/\mu]^{\beta}}. \tag{3.169}$$

The *three-parameter* Weibull distribution has wide applicability. The mathematical model for the *cumulative probability*, or the cumulative distribution function (c.d.f.) of the *three-parameter* Weibull distribution is

$$F(t) = 1 - \mathrm{e}^{-[(t-\gamma)/\mu]^{\beta}}, \tag{3.170}$$

where:

$F(t)$ = cumulative probability of failure,
γ = location or position parameter,
μ = scale parameter,
β = shape parameter.

The location, position, or minimum life parameter γ This parameter can be thought of as a guarantee period within which no failures occur, and a guaranteed *minimum life* could exist. This means that no appreciable or noticeable degradation or wear is evident before γ hours of operation. However, when a component is subject to failure immediately after being placed in service, no guarantee or failure-free period is apparent; then, $\gamma = 0$.

The scale or characteristic life parameter μ This parameter is a constant and, by definition, is the mean operating period or, in terms of system unreliability, the operating period during which at least 63% of the system's equipment is expected to fail. This '*unreliability*' value of 63%, which is obtained from the previous formula $Q = 1 - R = 100 - 37\%$, can readily be determined from the *reliability* model by substituting specific values for $\gamma = 0$, and $t = \mu$ in the case of the Weibull graph being a straight line, and the period t being equal to the characteristic life or scale parameter μ respectively.

The shape or failure pattern parameter β As its name implies, β determines the contour of the Weibull p.d.f. By finding the value of β for a given set of data, the particular phase of an equipment's *characteristic life* may be determined:

- When $\beta < 1$, the equipment is in a *wear-in* or *infant mortality* phase of its characteristic life, with a resulting *decreasing rate of failure*.
- When $\beta = 1$, the equipment is in the *steady operational period* or *service life* phase of its characteristic life, with a resulting *constant rate of failure*.
- When $\beta > 1$, the equipment begins to fail due to aging and/or degradation through use, and is in a *wear-out* phase of its characteristic life, with a resulting *increasing rate of failure*.

Since the *probability of survival* $p(s)$, or the reliability for the Weibull distribution, is the unity complement of the *probability of failure* $p(f)$, or failure distribution $F(t)$, the following mathematical model for reliability will plot a straight line on logarithmic scales

$$R(t) = p(s) = \mathrm{e}^{-[(t-\gamma)/\mu]^{\beta}} \,. \tag{3.171}$$

To facilitate calculations for the Weibull parameters, a *Weibull graph* has been developed. The principal advantage of this method of the Weibull analysis of failure is that it gives a complete picture of the type of distribution that is represented by the failure data and, furthermore, relatively few failures are needed to be able to make a satisfactory evaluation of the characteristics of component failure.

Figure 3.42 shows the basic features of the Weibull graph.

c) Procedure to Calculate the Weibull Parameters β, μ and γ

The procedure to calculate the Weibull parameters using the Weibull graph illustrated in Fig. 3.42 is given as follows:

- The percentage failure is plotted on the *y*-axis against the age at failure on the *x*-axis ($q - q$).

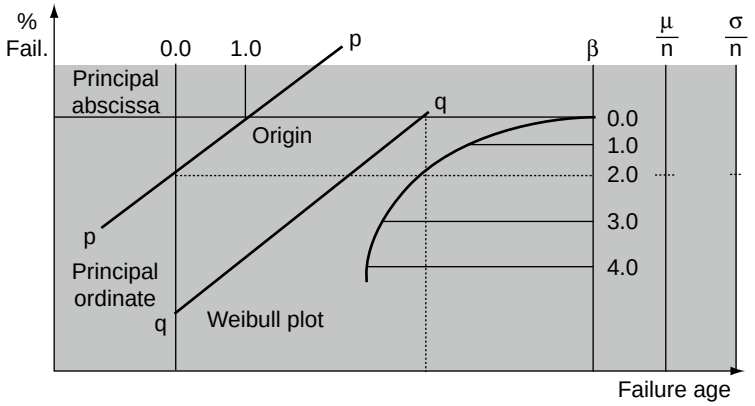


Fig. 3.42 Plot of the Weibull density function, $F(t)$, for different values of β

- If the plot is linear, then $\gamma = 0$. If the plot is non-linear, then $\gamma \neq 0$, and the procedure to make it linear by calculation is to add a constant value to the parameter γ in the event the plot is convex relative to the origin on the Weibull graph, or to subtract a constant value from the parameter γ in the event the plot is concave. A *best fit* straight line through the original plot would suffice.
- A line (pp) is drawn through the origin of the chart, parallel to the calculated linear Weibull plot (qq), or estimated straight line fit.
- The line pp is extended until it intersects *the principal ordinate*, (point i in Fig. 3.37). The value for β is then determined from the β-scale at a point horizontally opposite the line pp intersection with the principal ordinate.
- The linear Weibull plot (qq), or the graphically estimated straight line fit, is extended until it intersects the *principal abscissa*. The value for μ is then found at the bottom of the graph, vertically opposite the linear principal abscissa intersection.

d) Procedure to Derive the Mean Time Between Failures (MTBF)

Once the Weibull parameters have been determined, the *mean time between failures (MTBF)* may be evaluated. There are two other scales parallel to the β-scale on the Weibull graph:

$$\mu/n \quad \text{and} \quad \sigma/n\,,$$

where:

μ = characteristic life,
σ = standard deviation,
n = number of data points.

The value on the μ/n scale, adjacent to the previously determined value of β, is determined. This value is, in effect, the mean time between failures (MTBF), as a ratio to the number of data points, or the percentage failures that were plotted on the y-axis against the age at failure.

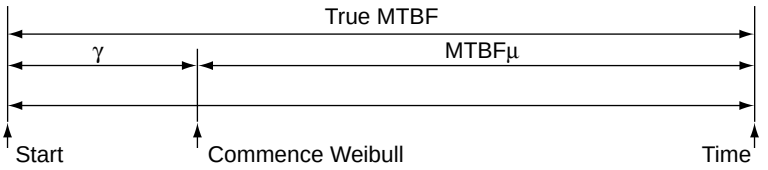
$$\text{Thus, MTBF} = \text{scale value of } \mu/n\,.$$

It is important to note that this mean value is referenced from the *beginning* of the Weibull distribution and should therefore be *added* to the *minimum life parameter* γ to obtain the *true* MTBF, as shown below in Fig. 3.43.

e) Procedure to Obtain the Standard Deviation σ

The standard deviation is the value on the σ/n scale, adjacent to the determined value of β.

$$\sigma = n \times \text{scale value of } \sigma/n\,.$$



True MTBF = from Start *to* Commence Weibull *to* Time
True MTBF = $\gamma + \mu$

Fig. 3.43 Minimum life parameter and true MTBF

The *standard deviation* value of the Weibull distribution is used in the conventional manner and can be applied to obtain a general idea of the *shape* of the distribution.

Summary of Quantitative Analysis of the Weibull Distribution Model

In the *two-parameter* Weibull, the parameters β and μ, where β is the shape parameter or *failure pattern*, and μ is the scale parameter or *characteristic life*, have an effect on the probability density function, reliability function and failure rate function (cf. Fig. 3.44).

The effect of β on the Weibull p.d.f. is that when $\beta > 1$, the probability density function, $f(T)$, assumes a *wear-out* type shape, i.e. the failure rate increases with time.

The effect of β on the Weibull reliability function, or one minus the cumulative distribution function c.d.f., $1 - F(t)$, is that when $\beta > 1$ and μ is constant, $R(T)$ decreases as T increases until wear-out sets in, when it decreases sharply and goes through an inflection point.

The effect of β on the Weibull hazard or failure rate function is that when $\beta > 1$, the hazard rate $\lambda(T)$ *increases* as T increases, and becomes suitable for representing the failure rate of components with wear-out type failures.

A change in the Weibull *scale parameter* μ has the effect that when μ, the *characteristic life*, is *increased* while β, the *failure pattern*, is constant, the distribution $f(T)$ is spread out with a greater variance about the mean and, when μ is *decreased* while β is constant, the distribution is peaked.

With the inclusion of γ, the location or *minimum life parameter* in a *three-parameter* Weibull distribution, no appreciable or noticeable degradation or wear is evident before γ hours of operation.

3.3.3.4 Qualitative Analysis of the Weibull Distribution Model

It was stated earlier that the principal advantage of Weibull analysis is that it gives a complete picture of the type of distribution that is represented by the failure data, and that relatively few failures are needed to be able to make a satisfactory assess-

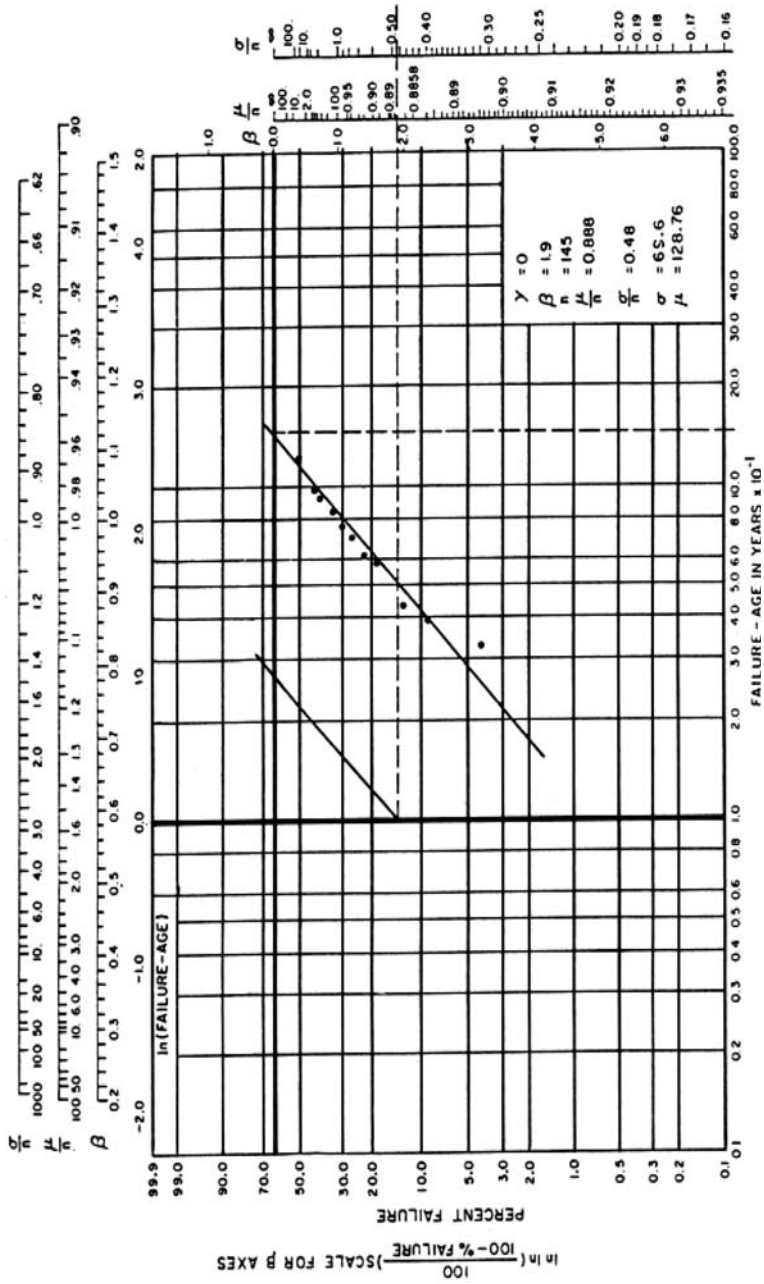


Fig. 3.44 Revised Weibull chart

ment of the characteristics of failure. A major problem arises, though, when the measures and/or estimates of the Weibull parameters *cannot be based on obtained data*, and engineering design analysis cannot be *quantitative*. Credible and statistically acceptable *qualitative* methodologies to determine the integrity of engineering design in the case where data are not available or not meaningful are included, amongst others, in the concept of *information integration technology (IIT)*.

IIT is a combination of techniques, methods and tools for collecting, organising, analysing and utilising diverse information to guide optimal decision-making. The method know as performance and reliability evaluation with diverse information combination and tracking (PREDICT) is a highly successful example (Booker et al. 2000) of IIT that has been applied in automotive system design and development, and in nuclear weapons storage. Specifically, IIT is a formal, multidisciplinary approach to evaluating the performance and reliability of engineering processes *when data are sparse or non-existent*. This is particularly useful when complex integrations of systems and their interactions make it difficult and even impossible to gather meaningful statistical data that could allow for a quantitative estimation of the performance parameters of probability distributions, such as the Weibull distribution.

The objective is to evaluate equipment reliability early in the detail design phase, by making effective use of all available information: expert knowledge, historical information, experience with similar processes, and computer models. Much of this information, especially expert knowledge, is *not* formally included in performance or reliability calculations of engineering designs, because it is often implicit, undocumented or not quantitative. The intention is to provide accurate reliability estimates for equipment while they are still in the engineering design stage. As equipment may undergo changes during the development or construction stage, or conditions change, or new information becomes available, these reliability estimates must be updated accordingly, providing a lifetime record of performance of the equipment.

a) Expert Judgment as Data

Expert judgment is the expression of informed opinion, based on knowledge and experience, made by experts in responding to technical problems (Ortiz et al. 1991). Experts are individuals who have specialist background in the subject area and are recognised by their peers as being qualified to address specific technical problems. Expert judgment is used in fields such as medicine, economics, engineering, safety/risk assessment, knowledge acquisition, the decision sciences, and in environmental studies (Booker et al. 2000).

Because expert judgment is often used implicitly, it is not always acknowledged as expert judgment, and is thus preferably obtained explicitly through the use of *formal elicitation*. Formal use of expert judgment is at the heart of the engineering design process, and appears in all its phases. For years, methods have been researched on how to structure elicitations so that analysis of this information can be performed statistically (Meyer and Booker 1991). Expertise gathered in an ad hoc manner is not recommended (Booker et al. 2000).

Examples of expert judgment include:

- the probability of an occurrence of an event,
- a prediction of the performance of some product or process,
- decision about what statistical methods to use,
- decision about what variables enter into statistical analysis,
- decision about which datasets are relevant for use,
- the assumptions used in selecting a model,
- decision concerning which probability distributions are appropriate,
- description of information sources for any of the above responses.

Expert judgment can be expressed *quantitatively* in the form of probabilities, ratings, estimates, weighting factors, distribution parameters or physical quantities (e.g. costs, length, weight). Alternatively, expert judgment can be expressed *qualitatively* in the form of textual descriptions, linguistic variables and natural language statements of extent or quantities (e.g. minimum life or characteristic life, burn-in, useful life or wear-out failure patterns).

Quantitative expert judgment can be considered to be data. Qualitative expert judgment, however, must be *quantified* in order for it also to be considered as data. Nevertheless, even if expert judgment is qualitative, it can be given the same considerations as for data made available from tests or observations, particularly with the following (Booker et al. 2000):

- Expert judgment is considered affected by how it is gathered. *Elicitation methods* take advantage of the body of knowledge on human cognition and motivation, and include procedures for countering effects arising from the phrasing of questions, response modes, and extraneous influences from both the elicitor and the expert (Meyer and Booker 1991).
- The methodology of experimental design (i.e. randomised treatment) is similarly applied in expert judgment, particularly with respect to *incompleteness* of information.
- Expert judgment has *uncertainty*, which can be characterised and subsequently analysed. Many experts are accustomed to giving uncertainty estimates in the form of simple ranges of values. In eliciting uncertainties, however, the natural tendency is to underestimate it.
- Expert judgment can be subject to several *conditioning factors*. These factors include the information to be considered, the phrasing of questions (Payne 1951), the methods of solving the problem (Booker and Meyer 1988), as well as the experts' assumptions (Ascher 1978). A formal structured approach to elicitation allows a better control over conditioning factors.
- Expert judgment can be combined with other quantitative data through *Bayesian updating*, whereby an expert's estimate can be used as a prior distribution for initial reliability calculation. The expert's reliability estimates are updated when test data become available, using Bayesian methods (Kerscher et al. 1998).
- Expert judgment can be accumulated in *knowledge systems* with respect to technical applications (e.g. problem solving). For example, the knowledge system can address questions such as 'what is x under circumstance y?', 'what is the

failure probability?', 'what is the expected effect of the failure?', 'what is the expected consequence?', 'what is the estimated risk?' or 'what is the criticality of the consequence?'.

b) Uncertainty, Probability Theory and Fuzzy Logic Reviewed

A major portion of engineering design analysis focuses on propagating *uncertainty* through the use of distribution functions of one type or another, particularly the Weibull distribution in the case of reliability evaluation. Uncertainties enter into the analysis in a number of different ways. For instance, all data and information have uncertainties. Even when no data are available, and estimates are elicited from experts, uncertainty values usually in the form of ranges are also elicited. In addition, mathematical and/or simulation models have uncertainties regarding their input–output relationships, as well as uncertainties in the choice of models and in defining model parameters.

Different measures and units are often involved in specifying the performances of the various systems being designed. To map these performances into common units, conversion factors are often required. These conversions can also have uncertainties and require representation in distribution functions (Booker et al. 2000).

Probability theory provides a coherent means for determining uncertainties. There are other interpretations of probability besides conventional distributions, such as the relative frequency theory and the subjective theory, as well as the Bayes theorem. Because of the flexibility of interpretation of the subjective theory (Bement et al. 2000a), it is perhaps the best approach to a *qualitative* evaluation of system performance and reliability, through the combination of diverse information.

For example, it is usually the case that some aspect of information relating to a specific design's system performance and/or its design reliability is known, which is utilised in engineering design analysis before observations can be made. Subjective interpretation of such information also allows for the consideration of one-of-a-kind failure events, and to interpret these quantities as a *minimal* failure rate.

Because reliability is a common performance metric and is defined as a probability that the system performs to specifications, probability theory is necessary in reliability evaluation. However, in using expert judgment due to data being unavailable, not all experts may think in terms of probability. The best approach is to use alternatives such as *possibility theory*, *fuzzy logic* and *fuzzy sets* (Zadeh 1965) where experts think in terms of rules, such as if–then rules, for characterising a certain type of *ambiguity uncertainty*.

For example, experts usually have knowledge about the system, expressed in statements such as 'if the temperature is too hot, the component's expected life will rapidly diminish'. While this statement contains no numbers for analysis or for probability distributions, it does contain valuable information, and the use of *membership functions* is a convenient way to capture and quantify that information (Laviolette 1995; Smith et al. 1998).

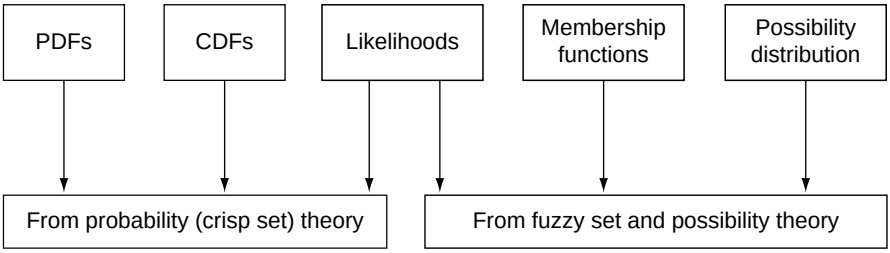


Fig. 3.45 Theories for representing uncertainty distributions (Booker et al. 2000)

However, reverting this information back into a probabilistic framework requires a bridging mechanism for the membership functions. Such a bridging can be accomplished using the Bayes theorem, whereby the membership functions may be interpreted as likelihoods (Bement et al. 2000b). This bridging is illustrated in Fig. 3.45, which depicts various methods used for formulating uncertainty (Booker et al. 2000).

c) Application of Fuzzy Logic and Fuzzy Sets in Reliability Evaluation

Fuzzy logic or, alternately, fuzzy set theory provides a basis for mathematical modelling and language in which to express quite sophisticated algorithms in a precise manner. For instance, fuzzy set theory is used to develop *expert system models*, which are fairly complex computer systems that model decision-making processes by a system of logical statements. Consequently, fuzzy set theory needs to be reviewed with respect to expert judgment in terms of *possibilities*, rather than *probabilities*, with the following definition (Bezdek 1993).

Fuzzy sets and membership functions reviewed Let X be a space of objects (e.g. estimated parameter values), and x be a generic element of X. A classical set A, $A \subseteq X$ is defined as a collection of elements or objects $x \in X$, such that each element x can either belong to or not be part of the set A. By defining a *characteristic* or *membership function* for each element x in X, a classical set A can be represented by a set of ordered pairs $(x,0)$ or $(x,1)$, which indicate $x \notin A$ or $x \in A$ respectively. Unlike conventional sets, a fuzzy set expresses the degree to which an element belongs to a set. Hence, the *membership function* of a fuzzy set is allowed to have values between 0 and 1, which denote the degree of membership of an element in the given set.

If X is a collection of objects denoted generically by x, then a *fuzzy set* A in X is defined as a set of ordered pairs where

$$A = \{(x, \mu_A(x)) | x \in X\} \tag{3.172}$$

in which $\mu_A(x)$ is called the *membership function* (or MF, for short) for the fuzzy set *A*.

The MF maps each element of *X* to a membership grade (or membership value) between 0 and 1 (included). Obviously, the definition of a fuzzy set is a simple extension of the definition of a classical (crisp) set in which the characteristic function is permitted to have any values between 0 and 1. If the value of the membership function is restricted to either 0 or 1, then *A* is reduced to a classical set. For clarity, references to classical sets consider ordinary sets, crisp sets, non-fuzzy sets, or just sets. Usually, *X* is referred to as the *universe of discourse* or, simply, the *universe*, and it may consist of discrete (ordered or non-ordered) objects or it can be a continuous space. However, a crucial aspect of fuzzy set theory, especially with respect to IIT, is *understanding how membership functions are obtained*.

The usefulness of fuzzy logic and mathematics based on fuzzy sets in reliability evaluation depends critically on the capability to construct appropriate membership functions for various concepts in various given contexts (Klir and Yuan 1995). Membership functions are therefore the fundamental connection between, on the one hand, empirical data and, on the other hand, fuzzy set models, thereby allowing for a bridging mechanism for reverting expert judgment on these membership functions back into a probabilistic framework, such as in the case of the definition of reliability.

Formally, the membership function μx is a function over some domain, or property space *X*, mapping to the unit interval $[0,1]$. The crucial aspect of fuzzy set theory is taken up in the following question: what does the membership function actually measure? It is an index of the membership of a defined set, which measures the degree to which object *A* with property *x* is a member of that set.

The usual definition of a classical set uses properties of objects to determine strict membership or non-membership. The main difference between classical set theory and fuzzy set theory is that the latter accommodates *partial* set membership. This makes fuzzy set theory very useful for modelling situations of *vagueness*, that is, *non-probabilistic uncertainty*. For instance, there is a fundamental ambiguity about the term 'failure characteristic' representing the parameter β of the Weibull probability distribution. It is difficult to put many items unambiguously into or out of the set of equipment currently in the *burn-in* or *infant mortality* phase, or in the *service life* phase, or in the *wear-out* phase of their *characteristic life*. Such cases are difficult to classify and, of course, depend heavily on the definition of 'failure'; in turn, this depends on the item's functional application. It is not so much a matter of whether the item could possibly be in a well-defined set but rather that the set itself does not have firm boundaries.

Unfortunately, there has been substantial confusion in the literature about the measurement level of a membership function. The general consensus is that a membership function is a ratio scale with two endpoints. However, in a continuous order-dense domain—that is, one in which there is always a value possible between any two given values, with no 'gaps' in the domain—the membership function may be considered as being not much different from a mathematical interval (Norwich and Turksen 1983). The membership function, unlike a probability measure, does not

fulfil the concatenation requirement that underlies any ratio scale (Roberts 1979). The simplest way to understand this is to consider the following concepts: it is meaningful to add the probability of the union of two mutually exclusive events, A and B, because a probability measure is a ratio scale

$$P(A) + P(B) = P(A \text{ and } B) \; . \tag{3.173}$$

It is *not*, however, meaningful to add the membership values of two objects or values in a fuzzy set.

For instance, the sum $\mu_A + \mu_B$ may be arithmetically *possibl*e but it is certainly not *interpretable* in terms of fuzzy sets. There does not seem to be any other concatenation operator in general that would be meaningful (Norwich and Turksen 1983). For example, if one were to add together two failure probability values in a series configuration, it makes sense to say that the probability of failure of the combined system is the sum of the two probabilities. However, if one were to take two failure probability *parameters* that are elements of fuzzy sets (such as the *failure characteristic* parameter β of the Weibull probability distribution), and attempt to sensibly add these together, there is no natural way to combine the two—unlike the failure probability.

By far the most common method for assigning membership is based on direct, subjective judgments by one or more experts. This is the method recommended for IIT. In this method, an expert rates values (such as the Weibull parameters) on a membership scale, assigning membership values directly and with no intervening transformations. For conceptually simple sets such as 'expected life', this method achieves the objective quite well, and should *not* be neglected as a means of obtaining membership values. However, the method has many shortcomings. Experts are often better with simpler estimates—e.g. paired comparisons or generating ratings on several more concrete indicators—than they are at providing values for one membership function of a relatively complex set.

Membership functions and probability measures One of the most controversial issues in uncertainty modelling and the information sciences is the relationship between probability theory and fuzzy sets. The main points are as follows (Dubois and Prade 1993a):

- Fuzzy set theory is a consistent body of mathematical tools.
- Although fuzzy sets and probability measures are distinct, there are several bridges relating these, including random sets and belief functions, and likelihood functions.
- Possibility theory stands at the crossroads between fuzzy sets and probability theory.
- Mathematical algorithms that behave like fuzzy sets exist in probability theory, in that they may produce random partial sets. This does not mean that fuzziness is reducible to randomness.
- There are ways of approaching fuzzy sets and possibility theory that are not conducive to probability theory.

Some interpretations of fuzzy sets are in agreement with probability calculus, others are not. However, despite misunderstandings between fuzzy sets and probabilities, it is just as essential to consider probabilistic interpretations of membership functions (which may help in membership function assessment) as it is to consider non-probabilistic interpretations of fuzzy sets. Some risk for confusion may be present, though, in the way various definitions are understood. From the original definition (Zadeh 1965), a fuzzy set F on a universe U is defined by a membership function:

$\mu_F: U \to [0,1]$ and $\mu_F(u)$ is the grade of membership of *element* u in F (for simplicity, let U be restricted to a finite universe).

In contrast, a probability measure P is a mapping $2^U \to [0,1]$ that assigns a number $P(A)$ to each *subset* of U, and satisfies the axioms

$$P(U) = 1;\ P(\emptyset) = 0 \tag{3.174}$$

$$P(A \cup B) = P(A) + P(B) \ \text{ if } \ A \cap B = \emptyset\,. \tag{3.175}$$

$P(A)$ is the probability that an ill-known single-valued variable x ranging on U coincides with the fixed well-known set A. Typical misunderstanding is to confuse the probability $P(A)$ with a membership grade. When $\mu_F(u)$ is considered, the element u is fixed and known, and the set is ill defined whereas, with the probability $P(A)$, the set A is well defined while the value of the underlying variable x, to which P is attached, is unknown. Such a set-theoretic calculus for probability distributions has been developed under the name of *Lebesgue logic* (Bennett et al. 1992).

Possibility theory and fuzzy sets reviewed Related to fuzzy sets is the development of the theory of possibility (Zadeh 1978), and its expansion (Dubois and Prade 1988). Possibility theory appears as a more direct contender to probability theory than do fuzzy sets, because it also proposes a set-function that quantifies the *uncertainty* of events (Dubois and Prade 1993a).

Consider a possibility measure on a finite set U as a mapping from 2^U to $[0,1]$ such that

$$\Pi(\emptyset) = 0 \tag{3.176}$$

$$\Pi(A \cup B) = \max(\Pi(A), \Pi(B))\,. \tag{3.177}$$

The condition $\Pi(U) = 1$ is to be added for normal possibility measures. These are completely characterised by the following possibility distribution π: $U \to [0,1]$ (such that $\pi(u) = 1$ for some $u \in U$, in the normal case), since $\Pi(A) = \max\{\pi(u), u \in A\}$.

In the infinite case, the equivalence between π and Π requires that Eq. (3.177) be extended to an infinite family of subsets. Zadeh (1978) views the possibility distribution π as being determined by the membership function μ_F of a fuzzy set F. This does not mean, however, that the two concepts of a fuzzy set and of a possibility distribution are equivalent (Dubois and Prade 1993a).

Zadeh's equation, given as $\pi_x(u) = \mu_F(u)$, is similar to equating the *likelihood function* to a conditional probability where $\pi_x(u)$ represents the relationship

$\pi(x = u|F)$, since it estimates the possibility that variable x is equal to the element u, with incomplete state of knowledge 'x is F'. Furthermore, $\mu_F(u)$ estimates the *degree* of compatibility of the precise information $x = u$ with the statement 'x is F'.

Possibility theory and probability theory may be viewed as complementary theories of uncertainty that model different kinds of states of knowledge. However, possibility theory further has the ability to model ignorance in a non-biased way, while probability theory, in its Bayesian approach, cannot account for ignorance. This can be explained with the definition of Bayes' theorem, which incorporates the concept of *conditional probability*.

In this case, conditional probability cannot be used directly in cases where ignorance prevails, for example:

'of the i components belonging to system F, j definitely have a high failure rate'.

Almost all the values for these variables are unknown. However, what might be known, if only informally, is how many components might fail out of a set F if a value for the *characteristic life parameter* μ of the system were available. As indicated previously, this parameter is by definition *the mean operating period in which the likelihood of component failure is 63%* or, conversely, it is *the operating period during which at least 63% of the system's components are expected to fail*.

Thus:

$$P(\text{component failure } f|\mu) \approx 63\% \,.$$

In this case, the Weibull *characteristic life parameter* μ must not be confused with the possibility distribution μ, and it would be safer to consider the probability in the following format:

$$P(\text{component failure } f|\text{characteristic life } c) \approx 63\% \,.$$

Bayes' theorem of probability states that if the likelihood of component failure and the number of components in the system are known, then the conditional probability of the characteristic life of the system (i.e. MTBF) may be evaluated, given an estimated number of component failures. Thus

$$P(c|f) = \frac{P(c)P(f|c)}{P(f)} \tag{3.178}$$

or:

$$\frac{|c \cap f|}{|f|} = \frac{|c|}{F} \cdot \frac{|f \cap c|}{|c|} \cdot \frac{F}{|f|} \,, \tag{3.179}$$

where:

$$|c \cap f| = |f \cap c|.$$

The point of Bayes' theorem is that the probabilities on the right side of the equation are easily available by comparison to the conditional probability on the left side. However, if the estimated number of component failures is *not* known (ignorance of

the probability of failure), then the conditional probability of the characteristic life of the system (MTBF) *cannot* be evaluated. Thus, probability theory in its Bayesian approach *cannot* account for ignorance.

On the contrary, possibility measures are decomposable (however, with respect to union only), and

$$N(A) = 1 - \Pi(\tilde{A}) \,, \tag{3.180}$$

where:

The certainty of A is 1—the impossibility of A,
$\tilde{A}$ is the complement (impossibility) of A, and $N(A)$ is a degree of certainty.

This is compositional with respect to intersection only, for example

$$N(A \cap B) = \min(N(A), N(B)) \,. \tag{3.181}$$

When one is totally ignorant about event A, we have

$$\Pi(A) = \Pi(\tilde{A}) = 1 \text{ and } N(A) = N(\tilde{A}) = 0 \,, \tag{3.182}$$

while

$$\Pi(A \cap \tilde{A}) = 0 \text{ and } N(A \cup \tilde{A}) = 1 \,. \tag{3.183}$$

This ability to model ignorance in a non-biased way is a typical asset of possibility theory.

The likelihood function Engineering design analysis is rarely involved with directly observable quantities. The concepts used for design analysis are, by and large, set at a fairly high level of abstraction and related to abstract design concepts. The observable world impinges on these concepts only indirectly. Requiring design engineers to rate conceptual objects on membership in a highly abstract set may be very difficult, and thus time and resources would be better spent using expert judgment to rate conceptual objects on more concrete scales, subsequently combined into a single index by an aggregation procedure (Klir and Yuan 1995).

Furthermore, judgment bias or *inconsistency* can creep in when ratings need to be estimated for conceptually complicated sets—which abound in engineering design analysis. It is much more difficult to defend a membership rating that comes solely from expert judgment when there is little to support the procedure other than the expert's status as an expert. It is therefore better to have a formal procedure in place that is transparent, such as IIT. In addition, it is essential that expert judgment relates to empirical evidence (Booker et al. 2000).

It is necessary to establish a relatively strong metric basis for membership functions for a number of reasons, the most important being the need to revert information that contains no numbers for analysis or for probability distributions, and that was captured and quantified by the use of membership functions, back into a probabilistic framework for further analysis. As indicated before, such a bridging can be accomplished using the Bayes theorem whereby the membership functions may be interpreted as likelihoods (Bement et al. 2000b).

The objective is to interpret the *membership function* of a fuzzy set as a *likelihood function*. This idea is not new in fuzzy set theory, and has been the basis of experimental design methods for constructing membership functions (Loginov 1966).

The likelihood function is a fundamental concept in statistical inference. It indicates how likely a particular set of values will contain an unknown estimated value. For instance, suppose an unknown random variable u that has values in the set U is to be estimated. Suppose also that the distribution of u depends on an unknown parameter $'\mathbf{F}'$, with values in the parameter space F. Let $P(u;'\mathbf{F}')$ be the probability distribution of the variable u, where $'\mathbf{F}'$ is the parameter vector of the distribution.

If x_o is the estimate of variable u, an outcome of expert judgment, then the *likelihood function* L is given by the following relationship

$$L('\mathbf{F}'|x_o) = P(x_o|'\mathbf{F}') \; . \tag{3.184}$$

In general, both u and x_o are vector valued. In other words, the estimate x_o is substituted instead of the random variable u into the expression for probability of the random variable, and the new expression is considered to be a function of the parameter vector $'\mathbf{F}'$.

The likelihood function may vary due to various estimates from the same expert judgment. Thus, in considering the probability density function of u at x_o denoted by $f(u|'\mathbf{F}')$, the likelihood function L is obtained by reversing the roles of $'\mathbf{F}'$ and u—that is, $'\mathbf{F}'$ is viewed as the variable and u as the estimate (which is precisely the point of view in estimation)

$$L('\mathbf{F}'|u) = f(u|'\mathbf{F}') \quad \text{for } '\mathbf{F}' \text{ in } F \text{ and } u \text{ in } U. \tag{3.185}$$

The likelihood function itself is not a probability (nor density) function because its argument is the parameter $'\mathbf{F}'$ of the distribution, not the random variable (vector) u. For example, the sum (or integral) of the likelihood function over all possible values of F should not be equal to 1. Even if the set of all possible values of F is discrete, the likelihood function still may be continuous (as the set of parameters F is continuous). In the method of *maximum likelihood*, a value u of the parameter $'\mathbf{F}'$ is sought that will maximise $L('\mathbf{F}'|u)$ for each u in U: $\max_{u\in F} L('\mathbf{F}'|u)$. The method determines the parameter values that would most likely produce the values estimated by expert judgment.

In an IIT context, consider a group of experts, wherein each expert is asked to judge whether the variable u, where $u \in U$, can be part of a fuzzy concept F or not. In this case, the likelihood function $L('\mathbf{F}'|u)$ is obtained from the probability distribution $P(u;'\mathbf{F}')$, and basically represents the proportion of experts that answered *yes* to the question. The function $'\mathbf{F}'$ is then the corresponding non-fuzzy parameter vector of the distribution (Dubois and Prade 1993a).

The *membership function* $\mu_F(u)$ of the fuzzy set F is the *likelihood function* $L('\mathbf{F}'|u)$

$$\mu_F(u) = L('\mathbf{F}'|u) \quad \forall u \in U \; . \tag{3.186}$$

This relationship will lead to a cross-fertilisation of fuzzy set and likelihood theories, provided it does not rely on a dogmatic Bayesian approach. The premise of Eq. (3.186) is to view the likelihood in terms of a conditional uncertainty measure—in this case, a probability. Other uncertainty measures may also be used, for example, the *possibility measure* Π, i.e.

$$\mu_F(u) = \Pi({}'\mathbf{F}'|u) \quad \forall u \in U \; . \tag{3.187}$$

This expresses the equality of the membership function describing the fuzzy class F viewed as a likelihood function with the possibility that an element u is classified in F. This can be justified starting with a possibilistic counterpart of the Bayes theorem (Dubois and Prade 1990)

$$\min\left(\pi(u|{}'\mathbf{F}'), \Pi({}'\mathbf{F}')\right) = \min\left(\Pi({}'\mathbf{F}'|u), \Pi(u)\right) \; . \tag{3.188}$$

This is assuming that no a priori (from cause to effect) information is available, i.e. $\pi(u) = 1 \; \forall u$, which leads to the following relationship

$$\pi(u|{}'\mathbf{F}') = \Pi({}'\mathbf{F}'|u) \; , \tag{3.189}$$

where:

π is the *conditional possibility distribution* that u relates to ${}'\mathbf{F}'$.

Fuzzy judgment in statistical inference Direct relationships between likelihood functions and possibility distributions have been pointed out in the literature (Thomas 1979), inclusive of interpretations of the likelihood function as a possibility distribution in the law of total probabilities (Natvig 1983).

The likelihood function is treated as a possibility distribution in classical statistics for so-called *maximum likelihood ratio tests*. Thus, if some hypothesis of the form $u \in F$ is to be tested against the opposite hypothesis $u \notin F$ on the basis of estimates of ${}'\mathbf{F}'$, and knowledge of the elementary likelihood function $L({}'\mathbf{F}'|u)$, $u \in U$, then the maximum likelihood ratio is the comparison between $\max_{u \in F} L({}'\mathbf{F}'|u)$ and $\max_{u \notin F} L({}'\mathbf{F}'|u)$, whereby the conditional possibility distribution is $\pi(u|{}'\mathbf{F}') = L({}'\mathbf{F}'|u)$ (Barnett 1973; Dubois et al. 1993a).

If, instead of the parameter vector ${}'\mathbf{F}'$, empirical values for expert judgment J are used, then

$$\pi(u|J) = L(J|u) \; . \tag{3.190}$$

The Bayesian updating procedure in which expert judgment can be combined with further information can be reinterpreted in terms of *fuzzy judgment*, whereby an expert's estimate can be used as a prior distribution for initial reliability until further expert judgment is available. Then

$$P(u|J) = \frac{L(J|u) \cdot P(u)}{P(J)} \; . \tag{3.191}$$

As an example, the probability function can represent the probability of failure of a component in an assembly set F, where the component under scrutiny is classed as 'critical'.

Thus, if p represents the base of the probability of failure of some component in an assembly set F, and the component under scrutiny is classed 'critical', where 'critical' is defined by the *membership function* μ_{critical}, then the a posteriori (from effect to cause) probability is

$$p(u|\text{critical}) = \frac{\mu_{\text{critical}}(u) \cdot p(u)}{P(\text{critical})} , \tag{3.192}$$

where $\mu_{\text{critical}}(u)$ is interpreted as the likelihood function, and the probability of a fuzzy event is given as (Zadeh 1968; Dubois et al. 1990)

$$P(\text{critical}) = \int_0^1 \mu_{\text{critical}}(u)\, \mathrm{d}P(u) . \tag{3.193}$$

d) Application of Fuzzy Judgment in Reliability Evaluation

The following methodology considers the combination of all available information to produce parameter estimates for application in Weibull reliability evaluation (Booker et al. 2000). Following the procedure flowchart in Fig. 3.46, the resulting

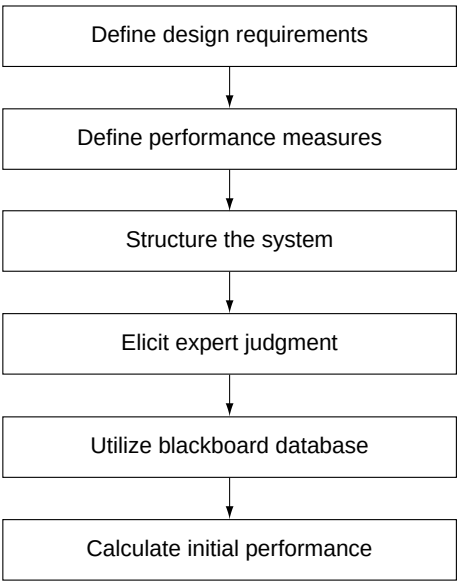


Fig. 3.46 Methodology of combining available information

fuzzy judgment information is in the form of an uncertainty distribution for the reliability of some engineering system design. This is defined at particular time periods for specific requirements, such as system warranty.

The random variable for the reliability is given as $R(t)$, where t is the period in an appropriate time measure (hours, days, months, etc.), and the uncertainty distribution function is $f(R;t,\theta)$, where θ is the set of Weibull parameters, i.e.

λ = *failure rate*,
β = shape parameter or *failure pattern*,
μ = scale parameter or *characteristic life*,
γ = location, or *minimum life parameter*.

For simplicity, consider the sources of information for estimating $R(t)$ and $f(R;t,\theta)$ originating from expert judgment, and from information arising from similar systems.

Structuring the system for system-level reliability Structuring the system is done according to the methodology of systems breakdown structuring (SBS) whereby an in-series system consisting of four levels is considered, namely:

- Level 1: process level
- Level 2: system level
- Level 3: assembly level
- Level 4: component level.

In reality, failure causes are also identified at the parts level, below the component level, but this extension is not considered here. Reliability estimates for the higher levels may come from two sources: information from the level itself, as well as from integrated estimates arising from the lower levels. The reliability for each level of the in-series system is defined as the product of the reliabilities within that level. The system-level reliability is the product R_S of all the lower-level reliabilities.

The system-level reliability, R_S, is computed as

$$R(t,\theta) = \prod_{j=1}^{n_S} R_S(t,\theta_j) \quad \text{for } n_S \text{ levels}\,. \tag{3.194}$$

$R_S(t,\theta_j)$ is a reliability model in the form of a probability distribution such as a three-parameter Weibull reliability function with

$$R_S(t,\beta_j,\mu_j,\gamma_j) = \mathrm{e}^{-[(t-\gamma)/\mu]^{\beta}}\,. \tag{3.195}$$

This reliability model must be appropriate and mathematically correct for the system being designed, and applicable for reliability evaluation during the detail design phase of the engineering design process.

It should be noted that estimates for λ, the failure rate or hazard function for each component, are also obtained from estimates of the three Weibull parameters γ, μ and β.

The γ *location parameter*, or *minimum life*, represents the period within which no failures occur at the onset of a component's life cycle. For practical reasons, it is convenient to leave the γ location parameter out of the initial estimation. This simplification, which amounts to an assumption that $\gamma = 0$, is frequently necessary in order to better estimate the β and μ Weibull parameters.

The β shape parameter, or *failure pattern*, normally fits the *early functional failure* ($\beta < 1$) and *useful life* ($\beta = 1$) characteristics of the system, from an implicit understanding of the design's reliability distribution, through the corresponding hazard curve's 'bathtub' shape.

The μ *scale parameter*, or *characteristic life*, is an estimate of the MTBF or the required operating period prior to failure. Usually, test data are absent for the conceptual and schematic design phases of a system. Information sources at this point of reliability evaluation in the system's detail design phase still reside mainly within the collective knowledge of the design experts. However, other information sources might include data from previous studies, test data from similar processes or equipment, and simulation or physical (industrial) model outputs.

The two-parameter Weibull cumulative distribution function is applied to *all three of the phases* of the *hazard rate curve* or equipment 'life characteristic curve', and the equation for the Weibull *probability density function* is the following (from Eq. 3.51):

$$f(t) = \frac{\beta \cdot t^{(\beta-1)}}{\mu^{\beta}} \cdot \mathrm{e}^{-t/\mu} \,, \tag{3.196}$$

where:

t = the *operating time* to determine *reliability* $R(t)$,
β = the Weibull distribution *shape parameter*,
μ = the Weibull distribution *scale parameter*.

As indicated previously, integrating out the Weibull probability density function gives the Weibull *cumulative distribution function* $F(t)$

$$F(t) = \int_0^1 f(t|\beta\mu)\,\mathrm{d}t = 1 - \mathrm{e}^{-t/\mu^{\beta}} \,. \tag{3.197}$$

The *reliability* for the Weibull probability density function is then

$$R(t) = 1 - F(t) = \mathrm{e}^{-t/\mu^{\beta}} \,, \tag{3.198}$$

where the Weibull *hazard rate function*, $\lambda(t)$ or *failure rate*, is derived from the ratio between the Weibull *probability density function*, and the Weibull *reliability function*

$$\lambda(t) = \frac{f(t)}{R(t)} = \frac{\beta(t)^{\beta-1}}{\mu^{\beta}} \,, \tag{3.199}$$

where μ is the component *characteristic life* and β the *failure pattern*.

e) Elicitation and Analysis of Expert Judgment

A formal elicitation is necessary to understand what expertise exists and how it can be related to the reliability estimation, i.e. how to estimate the Weibull parameters β and μ (Meyer et al. 2000). In this case, it is assumed that design experts are accustomed to working in project teams, and reaching a team consensus is their usual way of working. It is not uncommon, however, that not all teams think about performance using the same terms. Performance could be defined in terms of failures in *incidences per time period*, which convert to *failure rates* for equipment, or it could be defined in terms of failures in *parts per time period*, which translate to *reliabilities* for systems. Best estimates of such quantities are elicited from design experts, together with ranges of values. In this case, the most common method for assigning membership is based on direct, subjective judgments by one or more experts, as indicated above in Subsection c) Application of Fuzzy Logic and Fuzzy Sets in Reliability Evaluation.

In this method, a design expert rates values on a membership scale, assigning membership values with no intervening transformations. Typical fuzzy estimates for a membership function on a membership scale are interpreted as: most likely (median), maximum (worst), and minimum (best) estimates. The fundamental task is to convert these fuzzy estimates into the parameters of the Weibull distribution for each item of equipment of the design. Considering the uncertainty distribution function $f(R;t,\theta)$ (Booker et al. 2000), where θ is the set of Weibull parameters that include $\beta =$ *failure pattern*, $\mu =$ *characteristic life*, $\gamma =$ *minimum life parameter* and where $\gamma \neq 0$, an initial distribution for $\lambda =$ *failure rate* can be determined.

Failure rates are often asymmetric distributions such as the lognormal or gamma. Because of the variety of distribution shapes, the best choice for the failure rate parameter, λ, is the *gamma distribution* $f_n(t)$

$$f_n(t) = \frac{\lambda^n \cdot t^{(n-1)}}{(n-1)!} \cdot \mathrm{e}^{-\lambda t} \,, \tag{3.200}$$

where n is the *number of components* for which λ is the same.

This model is chosen because it includes cases in which more than one failure occurs.

Where more than one failure occurs, the reliability of the system can be judged not by the time for a single failure to occur but by the time for n failures to occur, where $n > 1$. The gamma probability density function thus gives an estimate of the time to the nth failure. This probability density function is usually termed the *gamma–n* distribution because the denominator of the probability density function is a gamma function.

Choosing the gamma distribution for the failure rate parameter λ is also appropriate with respect to the *characteristic life parameter* μ. As indicated previously, this parameter is by definition the mean operating period in which the likelihood of component failure is 63% or, in terms of system unreliability, it is the operating period during which at least 63% of the system's components are expected to fail.

Uncertainty distributions are also developed for the design's *reliabilities*, $R_S(t,\beta_j,\mu_j,\gamma_j)$, based on estimates of the Weibull parameters β_j, μ_j and γ_j, where $\gamma_j = 0$. The best choice for the distribution of reliabilities that are translated from the three estimates of best, most likely, and worst case values of the two Weibull parameters β_j, μ_j is the *beta distribution* $f_\beta(R|a,b)$, because of the beta's appropriate (0 to 1) range and its wide variety of possible shapes

$$f_\beta(R|a,b) = \frac{(a+b+1)!R^b}{a!b!}(1-R)^b\,, \tag{3.201}$$

where:

$f_\beta(R|a,b)$ = continuous distribution over the range $(0,1)$
R = reliabilities translated from the three estimates of best, most likely, and worst case values, and $0 < R < 1$
a = the number of survivals out of n
b = the number of failures out of n (i.e. $n-a$).

A general consensus concerning the γ parameter is that it should correspond to the typical minimum life of similar equipment, for which *warranty* is available. Maximum likelihood estimates for γ from Weibull fits of this warranty data provide a starting estimate that can be adjusted or confirmed for the equipment. Warranty data are usually available only at the system or sub-system/assembly levels, making it necessary to confirm a final decision about a γ value for all equipment at all system levels.

The best and worst case values of the Weibull parameters β_j and μ_j are defined to represent the maximum and minimum possible values. However, these values are usually weighted to account for the tendency of experts to underestimate uncertainty. Another difficulty arises when fitting three estimates, i.e. minimum (best), most likely (median), and maximum (worst), to the two-parameter Weibull distribution. One of the three estimates might not match, and the distribution may not fit exactly through all three estimates (Meyer and Booker 1991).

As part of the elicitation, experts are also required to specify all known or potential *failure modes* and *failure causes* (mechanisms) in engineering design analysis (FMECA) for reliability assessments of each item of equipment during the schematic design phase. The contribution of each failure mode is also specified. Although failure modes normally include failures in the components as such—e.g. a valve wearing out—they can also include faults arising during the manufacture of components, or the improper assembly/installation of multiple components in integrated systems. These manufacturing and assembly/installation processes are compilations of complex steps and issues during the construction/installation phase of engineering design project management, which must also be considered by expert judgment.

Figure 3.47 gives the baselines of an engineering design project, indicating the interface between the detail design phase and the construction/installation phase. Some of these issues relate to how quality control and inspections integrate with

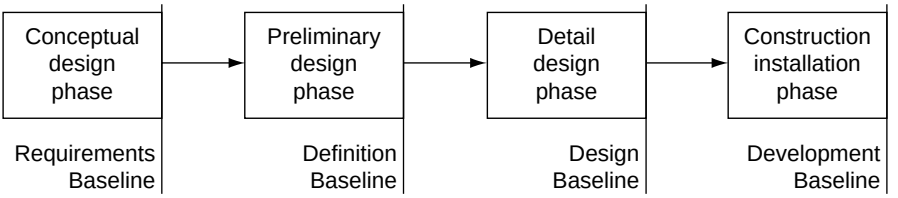


Fig. 3.47 Baselines of an engineering design project

the design process to achieve the overall integrity of engineering design. Reliability evaluation of these processes depends upon the percent or proportion of items that fail quality control and test procedures during the equipment commissioning phase. This aspect of engineering design integrity is considered later.

f) Initial Reliability Calculation Using Monte Carlo Simulation

Once the parameters and uncertainty distributions are specified for the design, the initial reliability, $R_S(t, \beta_j, \mu_j, \gamma_j)$, is calculated by using *Monte Carlo simulation*. As this model is time dependent, predictions at specified times are possible. Most of the expert estimates are thus given in terms of time t. For certain equipment, calendar time is important for warranty reasons, although in many cases operating hours is important as a lifetime indicator. The change from calendar time to operating time exemplifies the need for an appropriate conversion factor. Such factors usually have uncertainties attached, so the *conversion* also requires an uncertainty distribution. This distribution is developed using maximum likelihood techniques that are applied to typical operating time–calendar time relationship data. This uncertainty distribution also becomes part of the Monte Carlo simulation. The initial reliability calculation is concluded with system, assembly and component distributions calculated at these various time periods. Once expert estimates are interpreted in terms of *fuzzy judgment*, and prior distributions for an initial reliability are calculated, *Bayesian updating procedure* is then applied in which expert judgment is combined with other information, when it becomes available.

When the term *simulation* is used, it generally refers to any analytical method meant to imitate a real-life system, especially when other analyses are mathematically complex or difficult to reproduce. Without the aid of simulation, a mathematical model usually reveals only a single outcome, generally the most likely or average scenario, whereas with simulation the effect of varying inputs on outputs of the modelled system are analysed.

Monte Carlo (MC) simulations use random numbers and mathematical and statistical models to simulate real-world systems. Assumptions are made about how the model behaves, based either on samples of available data or on expert estimates, to gain an understanding of how the corresponding real-world system behaves.

MC simulation calculates multiple scenarios of the model by repeatedly sampling values from probability distributions for the uncertain variables, and using these values for the model. MC simulations can consist of as many trials (or scenarios) as required—hundreds or even thousands. During a single trial, a value from the defined possibilities (the range and shape of the distribution) is randomly selected for each uncertain variable, and the results recalculated. Most real-world systems are too complex for analytical evaluations.

Models must be studied with many simulation runs or iterations to estimate real-world conditions. Monte Carlo (MC) models are computer intensive and require many iterations to obtain a central tendency, and many more iterations to get confidence limit bounds. MC models help solve complicated deterministic problems (i.e. containing no random components) as well as complex probabilistic or stochastic problems (i.e. containing random components). Deterministic systems usually have one answer and perform the same way each time. Probabilistic systems have a range of answers with some central tendency.

MC models using probabilistic numbers will never give the exact same results. When simulations are rerun, the same answers are never achieved because of the random numbers that are used for the simulation. Rather, the central tendency of the numbers is determined, and the scatter in the data identified. Each MC run produces only estimates of real-world results, based on the validity of the model. If the model is not a valid description of the real-world system, then no amount of numbers will give the right answer. MC models must therefore have credibility checks to verify the real-world system. If the model is not valid, no amount of simulations will improve the expert estimates or any derived conclusions.

MC simulation randomly generates values for uncertain variables, over and over, to simulate the model. For each uncertain variable (one that has a range of possible values), the values are defined with a probability distribution. The type of distribution selected is based on the conditions surrounding that variable. These distribution types may include the normal, triangular, uniform, lognormal, Bernoulli, binomial and Poisson distributions. Bayesian inference from mixed distributions can feasibly be performed with Monte Carlo simulation.

In most of the examples, MC simulation models use the Weibull equation (as well as the special condition case where $\beta = 1$ for the exponential distribution). The Weibull equation used for such MC simulations has been solved for time constraint t, with the following relationship between the Weibull cumulative distribution function (c.d.f.), $F(t)$, t and β

$$t = \mu \cdot \ln[1/(1-F(t))]^{1/\beta} \ . \tag{3.202}$$

Random numbers between 0 and 1 are used in the MC simulation to fit the Weibull cumulative distribution function $F(t)$.

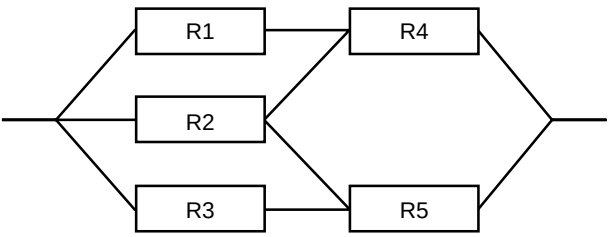
In complex systems, redundancy exists to prevent overall system failure, which is usually the case with most engineering process designs. For system success, some equipment (sub-systems, assemblies and/or components) of the system must be

successful simultaneously. The criteria for system success is based upon the system's configuration and the various combinations of equipment functionality and output, which is to be included in the simulation logic statement. The reliability of such complex systems is not easy to determine. Consequently, a relatively convoluted method of calculating the system's reliability is resorted to, through *Boolian truth tables.*

The size of these tables is usually large, consisting of 2^n rows of data, where n is the number of equipment in the system configuration. The reason the Boolian truth table is used is to calculate the theoretical reliability for the system based on the individual reliability values that are used for each item of equipment. On the first pass through the Boolian truth table, decisions are made in each row of the table about the combinations of successes or failures of the equipment. The second pass through the table calculates the contribution of each combination to the overall system reliability. The sum of all individual probabilities of success will yield the calculated system reliability. Boolian truth tables allow for the calculation of theoretical system reliabilities, which can then be used for Monte Carlo simulation. The simulation can be tested against the theoretical value, to measure how accurately the simulation came to reaching the correct answer.

As an example, consider the following MC simulation model of a complex system, together with the relative Boolian truth table, and Monte Carlo simulation results (Barringer 1993, 1994, 1995):

Given: reliability values for each block
Find: system reliability
Method: Monte Carlo simulation with Boolian truth tables:



Change, R-values	R_1 0.1	R_2 0.3	R_3 0.1	R_4 0.2	R_5 0.2	System ?
Cumulative successes	93	292	99	190	193	*131*
Cumulative failures	920	721	914	823	820	*882*
Total iterations	1013	1013	1013	1013	1013	*1013*
Simulated reliability	0.0918	0.2883	0.0977	0.1876	0.1905	*0.1293*
Theoretical reliability	0.1000	0.3000	0.1000	0.2000	0.2000	*0.1357*
% error	−8.19%	−3.92%	−2.27%	−6.22%	−4.74%	*−4.72%*

Boolean truth table Entry	R_1	R_2	R_3	R_4	R_5	Success or failure	Prob. of success
1	0	0	0	0	0	F	–
2	0	0	0	0	1	F	–
3	0	0	0	1	0	F	–
4	0	0	0	1	1	F	–
5	0	0	1	0	0	F	–
6	0	0	1	0	1	S	0.01008
7	0	0	1	1	0	F	–
8	0	0	1	1	1	S	0.00252
9	0	1	0	0	0	F	–
10	0	1	0	0	1	S	0.03888
11	0	1	0	1	0	S	0.03888
12	0	1	0	1	1	S	0.00972
13	0	1	1	0	0	F	–
14	0	1	1	0	1	S	0.00432
15	0	1	1	1	0	S	0.00432
16	0	1	1	1	1	S	0.00108
17	1	0	0	0	0	F	–
18	1	0	0	0	1	F	–
19	1	0	0	1	0	S	0.01008
20	1	0	0	1	1	S	0.00252
etc.							

g) Bayesian Updating Procedure in Reliability Evaluation

The elements of a *Bayesian reliability evaluation* are similar to those for a discrete process, considered in Eq. (3.179) above, i.e.:

$$P(c|f) = \frac{P(c) \cdot P(f|c)}{P(f)} \, .$$

However, the structure differs because the failure rate, λ, is well as the reliability, R_S, are continuous-valued. In this case, the Bayesian reliability evaluation is given by the formulae

$$P(\lambda_i|\beta_i, \mu_i, \gamma_i) = \frac{P(\lambda_i) \cdot P(\beta_i, \mu_i, \gamma_i|\lambda_i)}{P(\beta_i, \mu_i, \gamma_i)} \, , \tag{3.203}$$

where:

$$P(R_S|\beta_i, \mu_i, \gamma_i) = \frac{P(R_S) \cdot P(\beta_i, \mu_i, \gamma_i|R_S)}{P(\beta_i, \mu_i, \gamma_i)} \tag{3.204}$$

and:

$$P(\lambda_i|t) = \frac{\lambda^j \cdot t^{(j-1)}}{(j-1)!} \cdot \mathrm{e}^{-\lambda t}$$

$$P(R_\mathrm{S}|a,b) = \frac{(a+b+1)!}{a!b!} R_\mathrm{S}^b (1-R_\mathrm{S})^b$$

j = *number of components* with the same λ,
t = *operating time* for determining λ and R_S,
a = the number of survivals out of j,
b = the number of failures out of j (i.e. $j-a$).

For *both* the failure rate λ and reliability R_S, the probability $P(\beta_j, \mu_j, \gamma_j)$ may be either continuous or discrete, whereas the probabilities of $P(\lambda_j)$ for failure and of $P(R_\mathrm{S})$ for reliability are always continuous. Therefore, the prior and posterior distributions are always continuous, whereas the marginal distribution, $P(\beta_j, \mu_j, \gamma_j)$, may be either continuous or discrete.

Thus, in the case of expert judgment, new estimate values in the form of a likelihood function are incorporated into a Bayesian reliability model in a conventional way, representing updated information in the form of a posterior (a posteriori) probability distribution that depends upon a prior (a priori) probability distribution that, in turn, is subject to the estimated values of the Weibull parameters. Because the prior distribution and that for the new estimated values represented by a likelihood function are *conjugate* to one another (refer to Eq. 3.179), the mixing of these two distributions, by way of Bayes' theorem, ultimately results in a posterior distribution of the same form as the prior.

h) Updating Expert Judgment

The initial prediction of reliabilities made during the conceptual design phase may be quite poor with large uncertainties. Upon review, experts can decide which parts or processes to change, where to plan for tests, what prototypes to build, what vendors to use, or the type of *what–if* questions to ask in order to improve the design's reliability and reduce uncertainty. Before any usually expensive actions are taken (e.g. building prototypes), *what–if* cases are calculated to predict the effects on estimated reliability of such proposed changes or tests. These cases can involve changes in the structure, structural model, experts' estimates, and the terms of the reliability model as well as effects of proposed test data results. Further breakdown of systems into component failure modes may be required to properly map these changes and to modify proposed test data in the reliability model (Booker et al. 2000). Because designs are under progressive development or undergoing configuration change during the engineering design process, new information continually becomes available at various stages of the process. Design changes may include adding, replacing or eliminating processes and/or components in the light of new engineering judgment.

Incorporating these changes and new information into the existing reliability estimates is referred to as the *updating process*.

New information and data from different sources or of different types (e.g. tests, engineering judgment) are merged by combining uncertainty distribution functions of the old and new sources. This merging usually takes the form of a weighting scheme (Booker et al. 2000), $(w_1 f_1 + w_2 f_2)$, where w_1 and w_2 are weights and f_1 and f_2 are functions of parameters, random variables, probability distributions, or reliabilities, etc.

Experts often provide the weights, and sensitivity analyses are performed to demonstrate the effects of their choices. Alternatively, the Bayes theorem can be used as a particular weighting scheme, providing weights for the prior and the likelihood through application of the theorem. Bayesian combination is, in effect, Bayesian updating. If the prior and likelihood distributions overlap, then Bayesian combination will produce a posterior distribution with a smaller variance than if the two were combined via other methods, such as a linear combination of random variables. This is a significant advantage of using the Bayes theorem.

Because test data at the early stages of engineering design are lacking, initial reliability estimates, $R_0(t, \lambda, \beta)$, are developed from expert judgment, and form the prior distribution for the system (as indicated in Fig. 3.40 above). As the engineering design develops, data and information may become available for certain processes (e.g. systems, assemblies, components), and this would be used to form likelihood distributions for Bayesian updating. All of the distribution information in the items at the various levels must be combined upwards through the system hierarchy levels, to produce final estimates of the reliability and its uncertainty at various levels along the way, until reaching the top process or system level. As more data and information become available and are incorporated into the reliability calculation through Bayesian updating, they will tend to dominate the effects of the experts' estimates developed through expert judgment. In other words, $R_i(t, \lambda, \beta)$ formulated from $i = 1, 2, 3, \ldots, n$ test results will look less and less like $R_0(t, \lambda, \beta)$ derived from initial expert estimates.

Three different combination methods are used to form the following (updated) expert reliability estimate of $R_1(t, \lambda, \beta)$:

- For each prior distribution that is combined with data or likelihood distribution, the Bayes theorem is used for a posterior distribution.
- *Posterior distributions* within a given level are combined according to the model configuration (e.g. multiplication of reliabilities for systems/sub-systems/equipment in series) to form the prior distribution of the next higher level (Fig. 3.40).
- *Prior distributions* at a given level are combined within the same systems/sub-systems/equipment to form the combined prior (for that level), which is then merged with the data (for that system/sub-system/equipment). This approach is continued up the levels until a process-level posterior distribution is developed.

For general updating, test data and other new information can be added to the existing reliability calculation at any level and/or for any process, system or equipment. These data/information may be applicable only to a single failure mode at

equipment level. When new data or information become available at a higher level (e.g. sub-system) for a reliability calculation at step i, it is necessary to *back propagate* the effects of this new information to the lower levels (e.g. assembly or component). The reason is that at some future step, $i+j$, updating may be required at the lower level, and its effect propagated up the systems hierarchy. It is also possible to back propagate by apportioning either the reliability or its parameters to the lower hierarchy levels according to their contributions (criticality) at the higher systems level. The statistical analysis involved with this back propagation is difficult, requiring techniques such as *fault-tree analysis (FTA)* (Martz and Almond 1997).

While it can be shown that, for well-behaved functions, certain solutions are possible, they may not be unique. Therefore, constraints are placed on the types of solutions desired by the experts. For example, it may be required that, regardless of the apportioning used to propagate downwards, forward propagating maintain original results at the higher systems level. General updating is an extremely useful decision tool for asking *what–if* questions and for planning resources, such as pilot test facilities, to determine if the reliability requirements can be met before actually manufacturing and/or constructing the engineered installation. For example, the reliability uncertainty distributions obtained through simulation are empirical with no particular distribution form but, due to their asymmetric nature and because their range is from 0 to 1, they often appear to fit well to *beta distributions*. Thus, consider a beta distribution of the following form, for $0 = x = 1$, $a > 0$, $b > 0$

$$\text{Beta}(x|a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{(a-1)}(1-x)^{(b-1)} \,. \tag{3.205}$$

The beta distribution has important applications in Bayesian statistics, where probabilities are sometimes looked upon as random variables, and there is therefore a need for a relatively flexible probability density (i.e. the distribution can take on a great variety of shapes), which assumes non-zero values in the interval from 0 to 1. Beta distributions are used in reliability evaluation as estimates of a component's reliability with a continuous distribution over the range 0 to 1.

Characteristics of the Beta Distribution

The mean or expected value The mean, $E(x)$, of the two-parameter beta probability density function p.d.f. is given by

$$E(x) = \frac{a}{(a+b)} \,. \tag{3.206}$$

The mean $a/(a+b)$ depends on the ratio a/b. If this ratio is constant but the values for both a and b are increased, then the variance decreases and the p.d.f. tends to the unit normal distribution.

The median The beta distribution (as with all continuous distributions) has measures of location termed percentage points, X_{p}. The best known of these percentage

points is the *median*, X_{50}, the value of which there is as much chance that a random variable will be above as below it.

For a successes in n trials, the lower confidence limit u, at confidence level s, is expressed as a percentage point on a beta distribution. The median $\bar{u}$ of the two-parameter beta p.d.f. is given by

$$\bar{u} = 1 - F(u_{50}|a,b)\,. \tag{3.207}$$

The mode The mode or value with maximum probability, $\mathring{u}$, of the two-parameter beta p.d.f. is given by

$$\mathring{u} = \begin{cases} \dfrac{a-1}{(a+b-2)} & \text{for } a>1, b>1 \\ 0 \text{ and } 1 & \text{for } a<1, b<1 \\ 0 & \text{for } a<1, b\geq 1 \text{ and for } a=1, b>1 \\ 1 & \text{for } a\geq 1, b<1 \text{ and for } a>1, b=1 \end{cases} \tag{3.208}$$

$\mathring{u}$ does not exist for $a=b=1$.

If $a<1$, $b<1$, there is a minimum value or *antimode*.

The variance Moments about the mean describe the *shape* of beta p.d.f. The variance v is the second moment about the mean, and is indicative of the *spread* or *dispersion* of the distribution. The variance v of the two-parameter beta p.d.f. is given by

$$v = \frac{ab}{(a+b)^2(a+b+1)}\,. \tag{3.209}$$

The standard deviation The standard deviation σ_{T} of the two-parameter beta p.d.f. is the positive square root of the variance, v^2, which indicates the *closeness* one can expect the value of a random variable to be to the mean of the distribution, and is given by

$$\sigma_{\mathrm{T}} = \sqrt{ab/(a+b)^2(a+b+1)}\,. \tag{3.210}$$

Three-parameter beta distribution function The probability density function, p.d.f., of the three-parameter beta distribution function is given by

$$f(Y) = 1/c \cdot \mathrm{Beta}(x|a,b) \cdot (Y/c)^{a-1} \cdot (1-Y/c)^{b-1}\,, \tag{3.211}$$

for $0 \leq Y \leq c$ and $0<a$, $0<b$, $0<c$.

From this general three-parameter beta p.d.f., the standard two-parameter beta p.d.f. can be derived with the transform $x = Y/c$.

In the case where a beta distribution is fitted to a reliability uncertainty distribution, $R_i(t,\lambda,\beta)$, resulting in certain values for parameters a and b, the experts would want to determine what would be the result if they had the components manufactured under the assumption that most would not fail. Taking advantage of the beta distribution as a conjugate *prior* for the binomial data, the combined component

reliability distribution $R_j(t,\lambda,\beta)$ would also be a beta distribution. For instance, the beta expected value (mean), variance and mode, together with the fifth percentile for R_j can be determined from a reliability uncertainty distribution, $R_j(t,\lambda,\beta)$.

As an example, a beta distribution represents a reliability uncertainty distribution, $R_1(t,\lambda,\beta)$, with values for parameters $a = 8$ and $b = 2$. The beta expected value (mean), variance and mode, together with the fifth percentile value for R_1 are:

$R_1(t,\lambda,\beta)$ number of successes $a = 8$ and number of failures $b = 2$:
Distribution mean: 0.80
Distribution variance: 0.0145
Distribution mode: 0.875
Beta coefficient (E-value): 0.5709

Expert decision to have the components manufactured under the assumption that most will not fail depends upon the new component reliability distribution. The new reliability distribution would also be a beta distribution $R_2(t,\lambda,\beta)$ with modified values for the parameters being the following: $a = 8+$ number of successful prototypes and $b = 2+$ number unsuccessful. Assume that for five and ten manufactured components, the expectation is that one and two will fail respectfully:

For five components:
$R_2(t,\lambda,\beta)$ $a = 8+5$ and $b = 2+1$:
Distribution mean: 0.8125
Distribution variance: 0.0089
Distribution mode: 0.8571
Beta coefficient (E-value): 0.6366

For ten components:
$R_3(t,\lambda,\beta)$ $a = 8+10$ and $b = 2+2$:
Distribution mean: 0.8182
Distribution variance: 0.0065
Distribution mode: 0.85
Beta coefficient (E-value): 0.6708

The expected value improves slightly (from 0.8125 to 0.8182) but, more importantly, the 5th percentile E-value improves from 0.57 to 0.67, which is an incentive to invest in the components.

The general updating cycle can continue throughout the engineering design process. Figure 3.48 depicts tracking of the reliability evaluation throughout a system's design, indicating the three percentiles (5th, median or 50th, and 95th) of the reliability uncertainty distribution at various points in time (Booker et al. 2000).

The individual data points begin with the experts' initial reliability characterisation $R_0(t,\lambda,\beta)$ for the system and continue with the events associated with the general updates, $R_i(t,\lambda,\beta)$, as well as the *what–if* cases and incorporation of test results. As previously noted, asking *what–if* questions and evaluating the effects on reliability provides valuable information for engineering design integrity, and for modifying designs based on prototype tests before costly decisions are made.

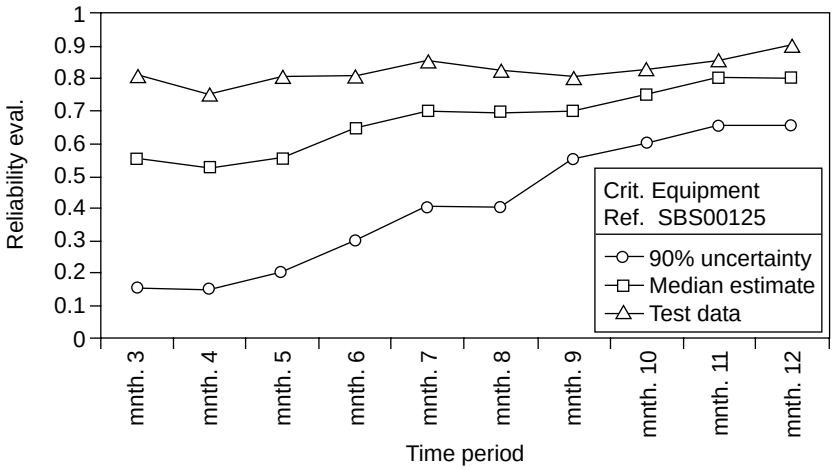


Fig. 3.48 Tracking reliability uncertainty (Booker et al. 2000)

Graphs such as Fig. 3.48 are constructed for all the hierarchical levels of *critical* systems to monitor the effects of updating for individual processes. Graphs are constructed for these levels at the desired prediction time values (i.e. monthly, 3-monthly, 6-monthly and annual) to determine if reliability requirements are met at these time points during the engineering design process as well as the manufacturing/construction/ramp-up life cycle of the process systems. These graphs capture the results of the experts' efforts to improve reliability and to reduce uncertainty. The power of the approach is that the *roadmap* developed leads to higher reliability and reduced uncertainty, and the ability to characterise all of the efforts to achieve improvement.

i) Example of the Application of Fuzzy Judgment in Reliability Evaluation

Consider an assembly set with series components that can influence the reliability of the assembly. The components are subject to various failures (in this case, the potential failure condition of *wear*), potentially *degrading* the assembly's reliability. For different component reliabilities, the assembly reliability will be variable. Figure 3.49 shows membership functions for three component condition sets, {A = no wear, B = moderate wear, C = severe wear}, which are derived from minimum (best), most likely (median) and maximum (worst) estimates.

Figure 3.50 shows membership functions for performance-level sets, corresponding to responses {a = acceptable, b = marginal, c = poor}.

Three *if–then* rules define the condition/performance relationship:

- If condition is A, then performance is a.
- If condition is B, then performance is b.
- If condition is C, then performance is c.

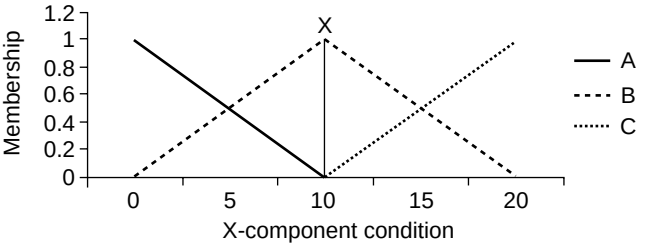


Fig. 3.49 Component condition sets for membership functions

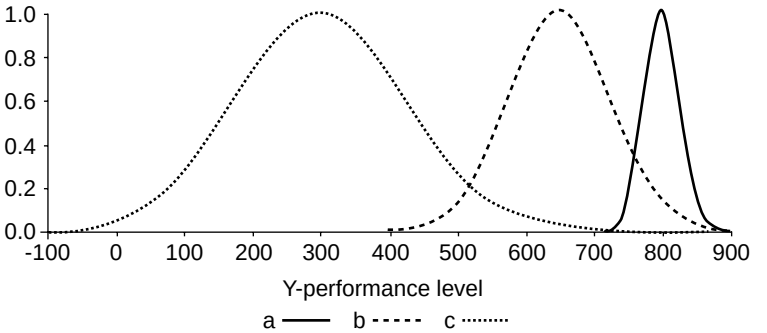


Fig. 3.50 Performance-level sets for membership functions

Referring to Fig. 3.49, if the component condition is $x = 4.0$, then x has membership of 0.6 in A and 0.4 in B. Using the rules, the defined component condition membership values are mapped to performance-level weights. Following fuzzy system methods, the membership functions for performance-level sets a and b are combined, based on the weights 0.6 and 0.4. This combined membership function can be used to form the basis of an uncertainty distribution for characterising performance for a given condition level. An equivalent probabilistic approach involving mixtures of distributions can be developed with the construction of the membership functions (Laviolette et al. 1995). In addition, linear combinations of random variables provide an alternative combination method when mixtures produce multi-modality results—which can be undesirable, from a physical interpretation standpoint (Smith et al. 1998).

Departing from standard fuzzy systems methods, the combined performance membership function can be normalised so that it integrates to 1.0. The resulting function, $f(y|x)$, is the uncertainty distribution for performance, y, corresponding to the situation where component condition is equal to x. The cumulative distribution function can now be developed, of the uncertainty distribution, $F(y|x)$. If performance must exceed some threshold, T, in order for the system to meet certain design criteria, then the reliability of the system for the situation where component condition is equal to x can be expressed as $R(x) = 1 - F(T|x)$. A specific threshold of T corresponds to a specific reliability of $R(4.0)$ (Booker et al. 1999).

In the event that the uncertainty in wear, x, is characterised by some distribution, $G(x)$, the results of repeatedly sampling x from $G(x)$ and calculating $F(y|x)$ produce an 'envelope' of cumulative distribution functions. This 'envelope' represents the uncertainty in the degradation probability that is due to uncertainty in the level of wear. The approximate distribution of $R(x)$ can be obtained from such a numerical simulation.

3.4 Application Modelling of Reliability and Performance in Engineering Design

In Sect. 1.1, the five main objectives that need to be accomplished in pursuit of the goal of the research in this handbook are:

- the development of appropriate theory on the integrity of engineering design for use in mathematical and computer models;
- determination of the validity of the developed theory by evaluating several case studies of engineering designs that have been recently constructed, that are in the process of being constructed, or that have yet to be constructed;
- application of mathematical and computer modelling in engineering design verification;
- determination of the feasibility of a practical application of intelligent computer automated methodology in engineering design reviews through the development of the appropriate industrial, simulation and mathematical models.

The following models have been developed, each for a specific purpose and with specific expected results, in part achieving these objectives:

- *RAMS analysis model* to validate the developed theory on the determination of the integrity of engineering design.
- *Process equipment models (PEMs)*, for application in dynamic systems simulation modelling to initially determine mass-flow balances for preliminary engineering designs of large integrated process systems, and to evaluate and verify process design integrity of complex integrations of systems.
- *Artificial intelligence-based (AIB) model*, in which relatively new *artificial intelligence (AI)* modelling techniques, such as inclusion of *knowledge-based expert systems* within a *blackboard model*, have been applied in the development of intelligent computer automated methodology for determining the integrity of engineering design.

The first model, the *RAMS analysis* model, will now be looked at in detail in this section of Chap. 3.

The *RAMS analysis* model was applied to an engineered installation, an environmental plant, for the recovery of sulphur dioxide emissions from a metal smelter to produce sulphuric acid. This model is considered in detail with specific reference

to the inclusion of the theory on reliability as well as performance prediction, assessment and evaluation, during the conceptual, schematic and detail design phases respectively.

Eighteen months after the plant was commissioned and placed into operation, failure data were obtained from the plant's distributed control system (DCS) operation and trip logs, and analysed with a view to matching the RAMS theory, specifically of systems and equipment criticality and reliability, with real-time operational data. The matching of theory with real-time data is studied in detail, with specific conclusions.

The *RAMS analysis* computer model (ICS 2000) provides a 'first-step' approach to the development of an artificial intelligence-based (AIB) model with knowledge-based expert systems within a blackboard model, for automated continual design reviews throughout the engineering design process. Whereas the RAMS analysis model is basically implemented and used by a single engineer for systems analysis, or at most a group of engineers linked via a local area network focused on general plant analysis, the AIB blackboard model is implemented by multi-disciplinary groups of design engineers who input specific design data and schematics into their relevant knowledge-based expert systems. Each designed system or related equipment is evaluated for integrity by remotely located design groups communicating either via a corporate intranet or via the internet. The measures of integrity are based on the theory for predicting, assessing and evaluating reliability, availability, maintainability and safety requirements for complex integrations of engineering systems.

Consequently, the feasibility of practical application of the AIB blackboard model in the design of large engineered installations has been based on the successful application of the RAMS analysis computer model in several engineering design projects, specifically in large 'super projects' in the metals smelting and processing industries. Furthermore, where only the conceptual and preliminary design phases were considered with the RAMS analysis model, all the engineering design phases are considered in the AIB blackboard model, to include a complete range of methodologies for determining the integrity of engineering design. Implementation of the RAMS analysis model was considered sufficient in reaching a meaningful conclusion as to the practical application of the AIB blackboard model.

3.4.1 The RAMS Analysis Application Model

The *RAMS analysis* model was used not only for plant analysis to determine the integrity of engineering design but also for design reviews as verification and evaluation of the commissioning of designed systems for installation and operation. The RAMS analysis application model was initially developed for analysis of the integrity of engineering design in an environmental plant for the recovery of sulphur dioxide emissions from a metal smelter to produce sulphuric acid.

In any complex process plant, there are literally thousands of different systems, sub-systems, assemblies and components, which are all subject to failure and, therefore, require specific attention with respect to the integrity of their design, design configuration as well as integration. To determine a logical starting point for any RAMS analysis, a hierarchical approach is first adopted, followed by identification of those items that are considered to be cost or process critical.

Cost critical items are the relatively few systems items of which the engineering costs (development, operational, maintenance and logistical support) make up a significant portion of the total costs of the engineered installation. *Process critical items* are those systems items that are the primary contributors to the continuation of the mainstream production process.

Determination of cost and process criticality should begin at the higher hierarchical levels of a *systems breakdown structure (SBS)*, such as the plant/facility level, since the total plant is normally broken down into logical operations/areas relating to the production process. Thus, rather than simply starting a RAMS analysis at one end of the plant and progressing through to the other end, focus is concentrated on specific areas based on their cost and process criticality. The Pareto principle is followed, which implies that 20% of the plant's areas contribute to 80% of the total engineering cost. When determining process criticality, the fundamental mainstream processes should first be identified based on the process flow and status changes of the process. All operations/areas in which the process significantly changes, and which are critical to the overall process flow, must be included. The different critical processes are then compared to those operations/areas identified as cost critical, to identify the sections or buildings (in the case of facilities) that are process critical but may not be considered as cost critical.

With such an approach, the RAMS analysis can proceed in a top-down progressive clarification of the plant's systems and equipment, already with an understanding of which items will have the highest criticality in terms of cost and process losses due to possible failure. As a result, the RAMS analysis deliverables can be summarised as follows:

RAMS activities	Deliverables
First-round costing	Estimate initial maintenance costs
Process definition	Develop operating procedures
	Develop plant shutdown and start-up procedures
Pre-commission	Initial equipment lists
Equipment register	Equipment technical specifications
	Manufacturer/supplier data
Plant definition	Equipment systems hierarchy structures
	Equipment inventory and systems coding
	Consolidated equipment technical specifications and group coding
FMEA	Failure modes, causes and effects matrices
	Failure diagnostics trouble-shooting charts

RAMS activities	Deliverables
Identification of certified and critical equipment (FMECA)	Critical equipment lists Plant safety requirements Process reliability evaluation Risk management directives
Spares requirements planning (SRP)	BOM and catalogue numbering Spares lists and critical spares Suppliers, supply lead times and supply costs
Maintenance standard work instructions (SWI)	Relevant statutory requirements Safe work practices Required safety gear
Design updates and/or reviews	Equipment modification review Interdisciplinary participation
Plant procedures	Statutory safety procedures
Maintenance procedures	Maintenance tasks per discipline/equipment Maintenance procedures sheets and coding for work orders cross referencing
Plant shutdown procedures	Plant shutdown tasks per discipline and per equipment
Manning requirements	Maintenance task times Maintenance trade crew requirements
Maintenance budgeting	Manning/spares costs against estimated maintenance tasks

The *RAMS analysis* application model is object-oriented client/server database technology initially developed in Microsoft's Visual Basic and Access. The model consists of a front-end user interface structured in OOP with drill-down data input and/or access to a normalised hierarchical database. The database consists of several keyword-linked data tables relating to major development tasks of the RAMS analysis, such as equipment, process, systems, functions, conditions tasks, procedures, costs, criticality, strategy, SWI (instructions) and logistics. These data tables relate to specific analysis tasks of the RAMS model. The keywords linking each data table reflect a structured six-tier systems breakdown structure (SBS), starting at the highest systems level of plant/facility, down to the lowest systems level of component/item. The SBS data table keywords are: plant, operation, section, system, assembly, component.

Database analysis tools, and database structuring in an SBS, enables the user to review visual data references to specific record dynasets in each of the data tables, as illustrated in Fig. 3.51.

Database structuring in an SBS, and the normalising of each dynaset of hierarchical structured records with a unique identifier (EQUIPID), allows for the establishment of a *normalised hierarchical database*. These dynasets include specific analysis activities such as:

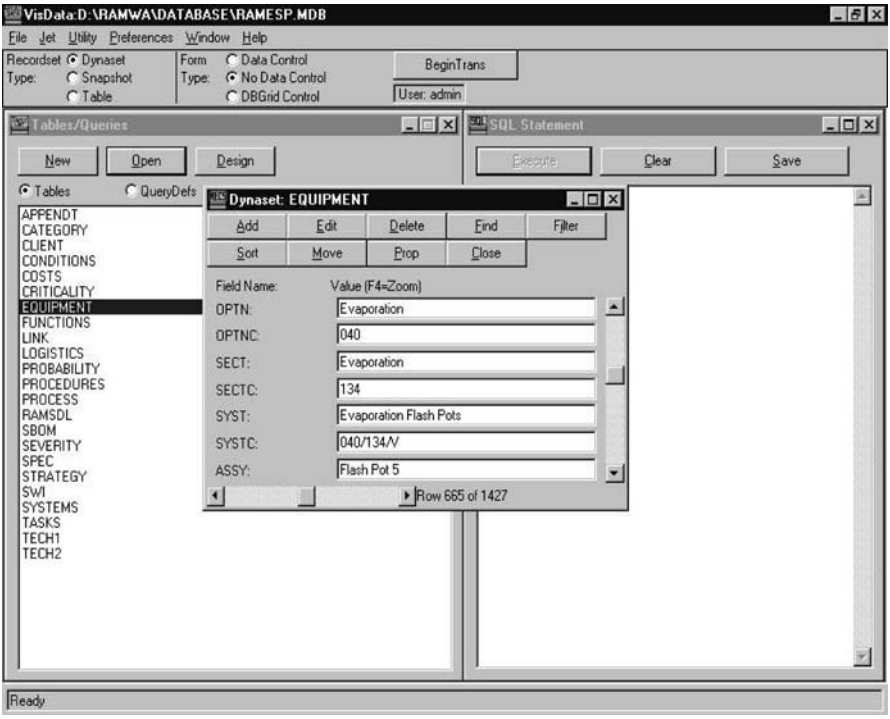


Fig. 3.51 Database structuring of SBS into dynasets

- PFD (process flow diagrams),
- P&ID (pipe and instrument diagrams),
- technical specifications,
- process specifications,
- operating specifications,
- function specifications,
- failure characteristics/conditions,
- fault diagnostics,
- equipment criticality and performance measures,
- operating procedures,
- maintenance procedures,
- process cost models,
- operating/maintenance strategies,
- safety inspection strategies,
- standard work instructions,
- spares requirements.

In designing hierarchical relational database tables, database normalisation minimises duplication of information and, in so doing, safeguards the database against certain types of logical or structural problems, specifically data anomalies. For

example, when multiple instances of information pertaining to a single item of equipment in a dynaset of hierarchical structured records occur in a data table, the possibility exists that these instances will not be kept consistent when the data within the table are updated, leading to a loss of data integrity. A table that is sufficiently normalised is less vulnerable to problems of this kind, because its structure reflects the basic assumptions for when multiple instances of the same information should be represented by a single instance only. Higher degrees of normalisation involve more tables and create the need for a larger number of joins or unique identifiers (such as EQUIPID), which reduces performance. Accordingly, more highly normalised tables are used in database applications involving many transactions (typically of the dynasets of analysis activities listed above), while less normalised tables tend to be used in database applications that do not need to map complex relationships between data entities and data attributes.

The initial systems hierarchical structure, or systems breakdown structure (SBS), illustrated in the RAMS analysis model in Fig. 3.52 is an overview *location listing* of the plant into the following systems hierarchy:

Systems hierarchy	Description
Plant/facility	Environmental plant
Operation/area	Effluent treatment
Section/building	Effluent neutralisation

The initial systems structure of an engineered installation must inevitably begin at the higher hierarchical levels of the systems breakdown structure, which constitutes a ‘top-down’ approach. However, such an SBS will have already been developed at the engineering design stage and, consequently, a ‘bottom-up’ approach can also be considered, especially for plant analysis of components and their failure effects on assemblies and systems.

The initial front-end structuring of the plant begins with the identification of operation/area, and section/building groups in a systems breakdown structure. As illustrated in Fig. 3.53, this structuring further provides visibility of process systems and their constituent assemblies and components in the RAMS analysis model spreadsheets, process flows and treeviews. Relevant information can be hierarchically viewed from system level, down to sub-system, assembly, sub-assembly and component levels. The various levels of the systems breakdown structure are normally determined by a framework of criteria that is established to logically group similar components into sub-assemblies or assemblies, which are then logically grouped into sub-systems or systems. This logical grouping of the constituent items of each level of an SBS is done by identifying the actual physical design configuration of the various items of one level of the SBS into items of a higher level of systems hierarchy, and by defining common operational and physical functions of the items at each level.

The systems hierarchical structure or systems breakdown structure (SBS) is a complete *equipment listing* of the plant into the following hierarchy with related example descriptions:

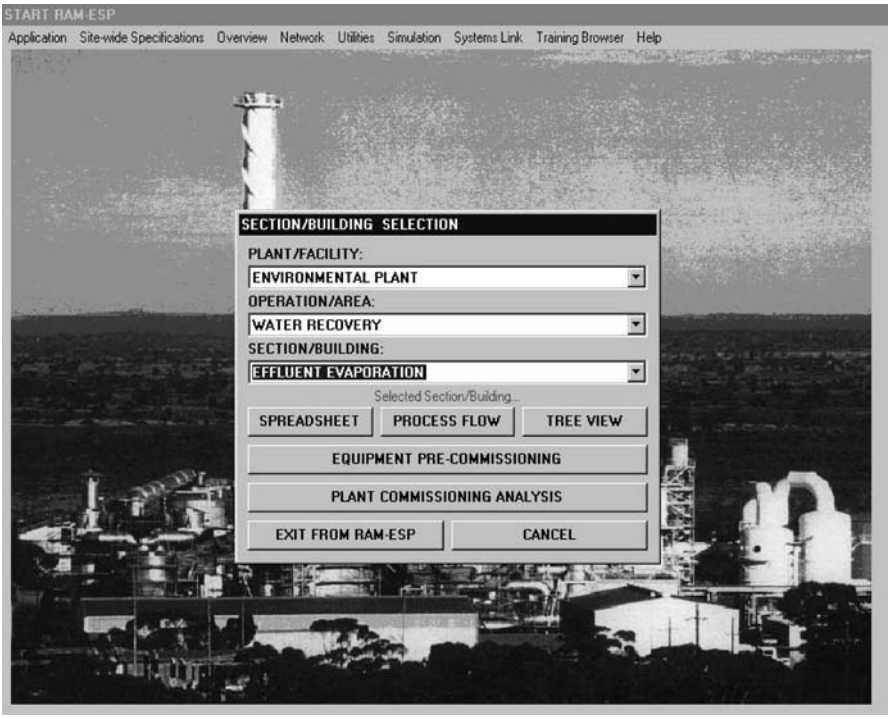


Fig. 3.52 Initial structuring of plant/operation/section

Systems hierarchy	Description
Plant/facility	Environmental plant
Operation/area	Effluent treatment
Section/building	Effluent neutralisation
System/process	*Evaporator feed tank*
Assembly/unit	*Feed pump no.1*
Component/item	*Motor–feed pump no.1*

Figure 3.54 illustrates a global grid list (or spreadsheet) of a specific system's SBS in establishing a complete equipment listing of that system.

The purpose for describing the systems in more detail is to ensure a common understanding of exactly where the boundaries of the system are, and which are the major sub-systems, assemblies and components encompassed by the system. The boundaries to other systems and the interface components that form these boundaries must also be clearly specified. This is usually done according to the most appropriate of the following criteria that are then described for the system:

- Systems boundary according to major function.
- Systems boundary according to material flow.

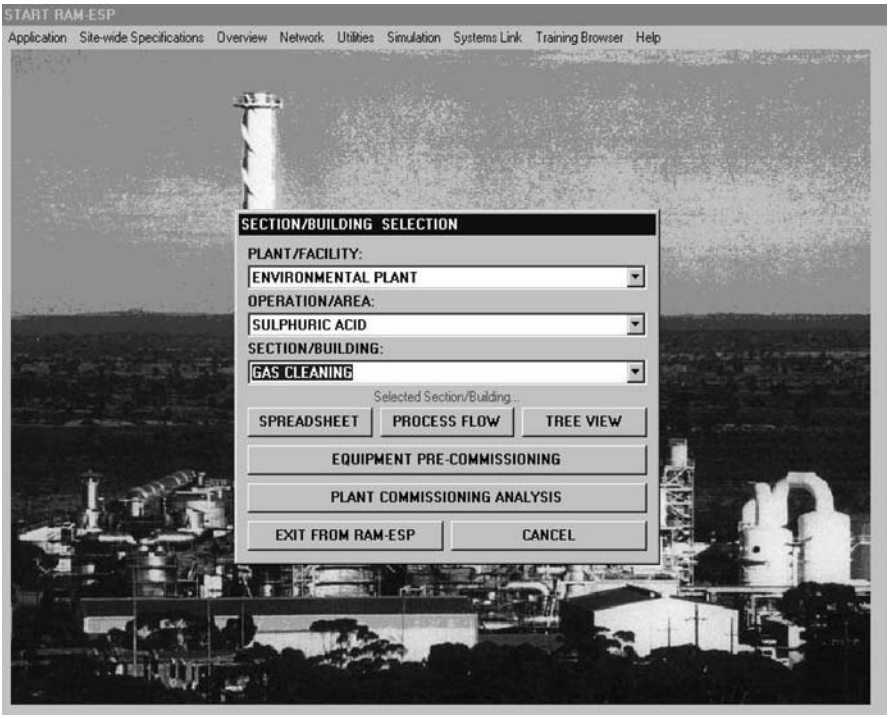
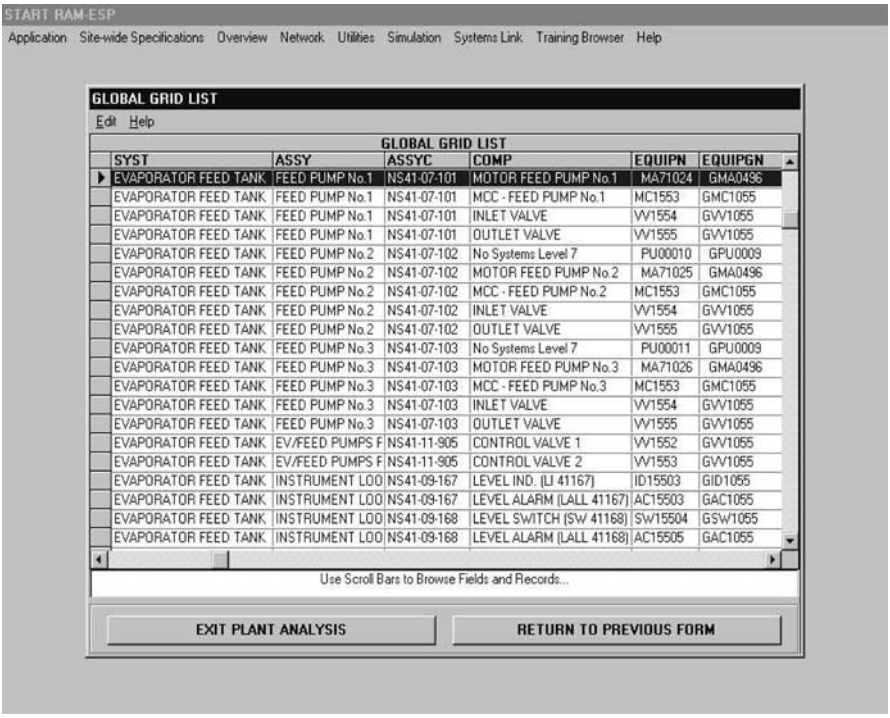


Fig. 3.53 Front-end selection of plant/operation/section: RAMS analysis model spreadsheet, process flow, and treeview

- Systems boundary according to process flow.
- Systems boundary according to mechanical action.
- Systems boundary according to state changes.
- Systems boundary according to input, throughput or output.

Interconnecting components such as cabling and piping between the boundaries of two systems should be regarded as part of the system from which the process flow emanates and enters the other system's boundary. The interface components, which are those components on the systems boundary, also need to be clearly specified since it is these components that frequently experience functional failures. Also, systems such as a hydraulic system, for instance, may not contain all the components that operate hydraulically. For example, a hydraulic lube oil pump should rather be placed under the lubrication sub-system. Where each assembly or a component is placed in the SBS should be based on the criteria selected for boundary determination. Normally for process plant, the criteria would typically be that of inputs and outputs, so that the outputs of each assembly and component contribute directly to the outputs of the system.



SYST	ASSY	ASSYC	COMP	EQUIPN	EQUIPGN
EVAPORATOR FEED TANK	FEED PUMP No.1	NS41-07-101	MOTOR FEED PUMP No.1	MA71024	GMA0496
EVAPORATOR FEED TANK	FEED PUMP No.1	NS41-07-101	MCC - FEED PUMP No.1	MC1553	GMC1055
EVAPORATOR FEED TANK	FEED PUMP No.1	NS41-07-101	INLET VALVE	VV1554	GVV1055
EVAPORATOR FEED TANK	FEED PUMP No.1	NS41-07-101	OUTLET VALVE	VV1555	GVV1055
EVAPORATOR FEED TANK	FEED PUMP No.2	NS41-07-102	No Systems Level 7	PU00010	GPU0009
EVAPORATOR FEED TANK	FEED PUMP No.2	NS41-07-102	MOTOR FEED PUMP No.2	MA71025	GMA0496
EVAPORATOR FEED TANK	FEED PUMP No.2	NS41-07-102	MCC - FEED PUMP No.2	MC1553	GMC1055
EVAPORATOR FEED TANK	FEED PUMP No.2	NS41-07-102	INLET VALVE	VV1554	GVV1055
EVAPORATOR FEED TANK	FEED PUMP No.2	NS41-07-102	OUTLET VALVE	VV1555	GVV1055
EVAPORATOR FEED TANK	FEED PUMP No.3	NS41-07-103	No Systems Level 7	PU00011	GPU0009
EVAPORATOR FEED TANK	FEED PUMP No.3	NS41-07-103	MOTOR FEED PUMP No.3	MA71026	GMA0496
EVAPORATOR FEED TANK	FEED PUMP No.3	NS41-07-103	MCC - FEED PUMP No.3	MC1553	GMC1055
EVAPORATOR FEED TANK	FEED PUMP No.3	NS41-07-103	INLET VALVE	VV1554	GVV1055
EVAPORATOR FEED TANK	FEED PUMP No.3	NS41-07-103	OUTLET VALVE	VV1555	GVV1055
EVAPORATOR FEED TANK	EV/FEED PUMPS F	NS41-11-905	CONTROL VALVE 1	VV1552	GVV1055
EVAPORATOR FEED TANK	EV/FEED PUMPS F	NS41-11-905	CONTROL VALVE 2	VV1553	GVV1055
EVAPORATOR FEED TANK	INSTRUMENT LOO	NS41-09-167	LEVEL IND. (LI 41167)	ID15503	GID1055
EVAPORATOR FEED TANK	INSTRUMENT LOO	NS41-09-167	LEVEL ALARM (LALL 41167)	AC15503	GAC1055
EVAPORATOR FEED TANK	INSTRUMENT LOO	NS41-09-168	LEVEL SWITCH (SW 41168)	SW15504	GSW1055
EVAPORATOR FEED TANK	INSTRUMENT LOO	NS41-09-168	LEVEL ALARM (LALL 41168)	AC15505	GAC1055

Fig. 3.54 Global grid list (spreadsheet) of systems breakdown structuring

The selected system is then described using the following steps:

- Determine the relevant process flow and inputs and outputs, and develop a process flow block diagram, specifically for process plant.
- List the major sub-systems and assemblies in the system, based on the appropriate criteria that will also be used for boundary determination.
- Identify the boundaries to other systems and specify the boundary interface components.
- Write an overview narrative that briefly describes the contents, criteria and boundaries of the systems under description.

A complete *equipment listing* of a plant includes the following activities at each systems hierarchical level:

Equipment listing at *system level* provides the ability to:

- identify groups of maintenance tasks for maintenance procedures,
- identify groups of maintenance tasks for maintenance budgets,
- identify critical systems for plant criticality,
- identify critical systems for maintenance priorities,
- identify critical systems for plant shutdown strategies.

Equipment listing at *assembly level* provides the ability to:

- identify location of pipelines,
- identify location of pumps,
- give codes to pumps, lube assemblies, etc.,
- identify critical assemblies for maintenance strategies.

Equipment listing at *component level* provides the ability to:

- identify relevant technical data of common equipment groups,
- identify relevant technical data to establish bill of materials groups,
- identify and link bill of spares,
- identify critical components for spares purchase,
- identify location of instrumentation,
- identify location of valves,
- give codes to classified/critical manual valves,
- identify required maintenance tasks,
- establish necessary standard work instructions,
- establish necessary safe work practices,
- give codes to valves for operation safety procedures,
- give codes to MCC panels, gearboxes, etc.

A *process flow diagram (PFD)*, as the name implies, graphically depicts the process flow and can be used to show the conversion of inputs into outputs, which subsequently form inputs into the next system. A process flow diagram essentially depicts the relationship of the different systems and sub-systems to each other, based on material or status changes that can be determined by studying the conversion of inputs to outputs at the different levels in each of the systems and sub-systems. One reason for drawing process flow diagrams is to determine the nature of the process flow in order to be able to logically determine systems relationships and the different hierarchical levels within the systems.

Most process engineering schematic designs start off with simple process flow diagrams, as that illustrated in Fig. 3.55, from which material flow and state changes in the process can then be identified. This is done by studying the changes from inputs to outputs of the different systems and determining the systems' boundaries as well as the interface components on these boundaries. A side benefit is a complete description of the system.

The treeview option enables users to view selected components in their cascaded systems hierarchical treeview structure, relating the equipment and their codes to the following systems hierarchy structure:

- parts,
- components,
- assemblies,
- systems,
- sections,
- operations,

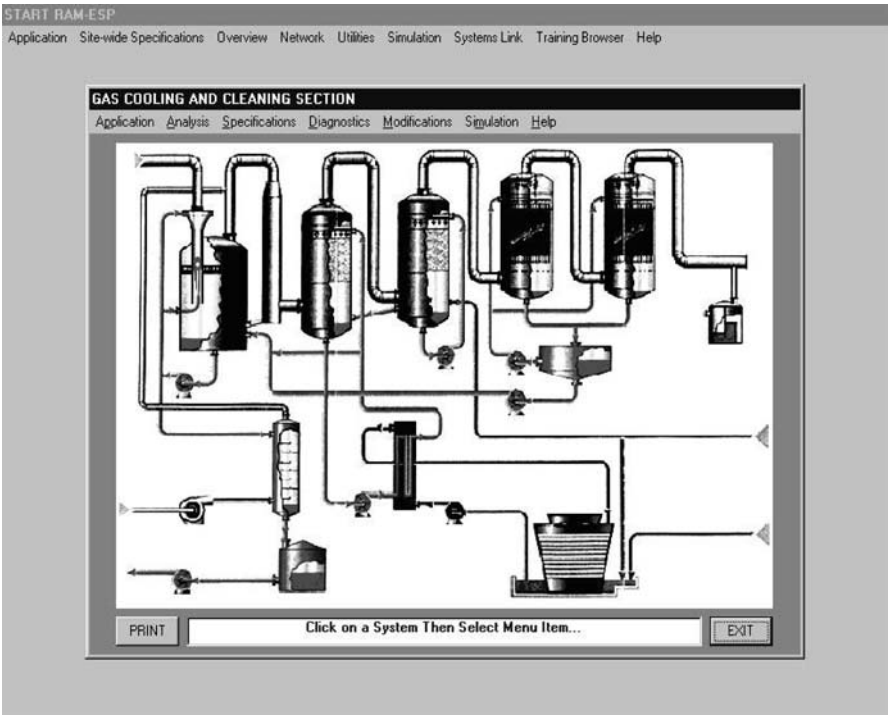


Fig. 3.55 Graphics of selected section PFD

- plant,
- site.

Figure 3.56 illustrates a typical treeview in the RAMS plant analysis model with expanded SBS (cascaded systems structure) for each system.

The *RAMS analysis list* is a sequential options list of the major development activities and specifically detailed specifications of a system selected from the section process flow diagram (PFD). By clicking on the PFD, a selection box appears for analysis.

The options listed in the selection box in Fig. 3.57 include the following analysis activities:

- Overview
- Analysis
- Specifications
- Diagnostics
- Modifications
- Simulation
- Decision logic
- Planning
- SWIs
- Procedures
- BOMs
- Technical data
- Grid list
- PIDs
- Reports
- Treeviews

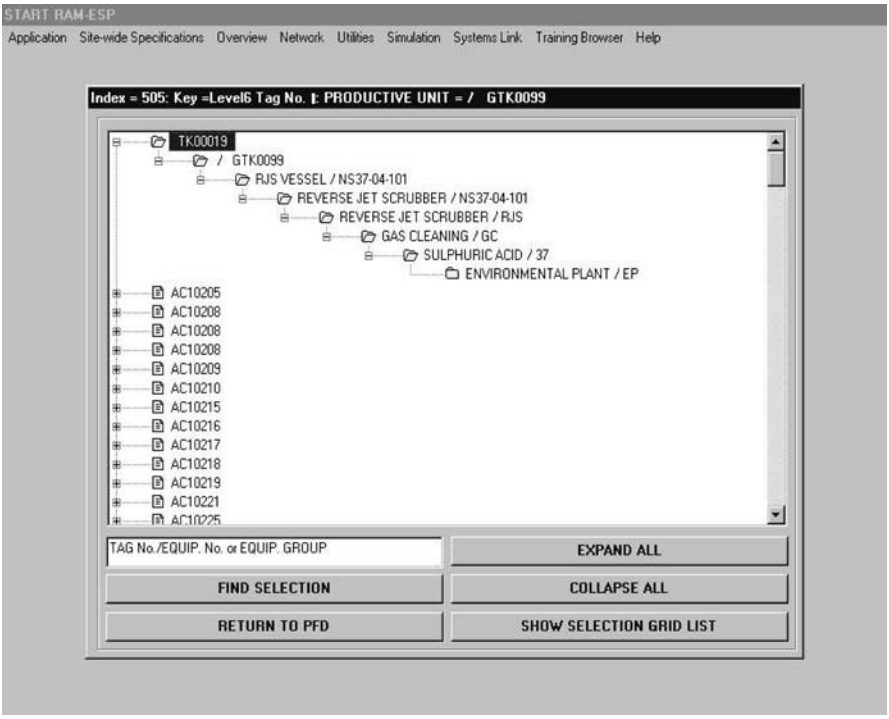


Fig. 3.56 Graphics of selected section treeview (cascaded systems structure)

The first category in the RAMS analysis list is an *overview* of specifically detailed *technical specifications* relating to the equipment's SBS, specifications, function and requirements, including the following:

- Equipment specifications
- Systems specifications
- Process specifications
- Function specifications
- Detailed tasks
- Detailed procedures
- Logistic requirements
- Standard work instructions.

Figure 3.58 illustrates the use of the overview option and equipment specification information displayed in the *equipment* tab, such as equipment description, equipment number, equipment reference and the related position in the SBS data table.

The technical data worksheet illustrated in Fig. 3.59 is established for each item of equipment that is considered during the design process to determine and/or modify specific equipment technical criteria such as:

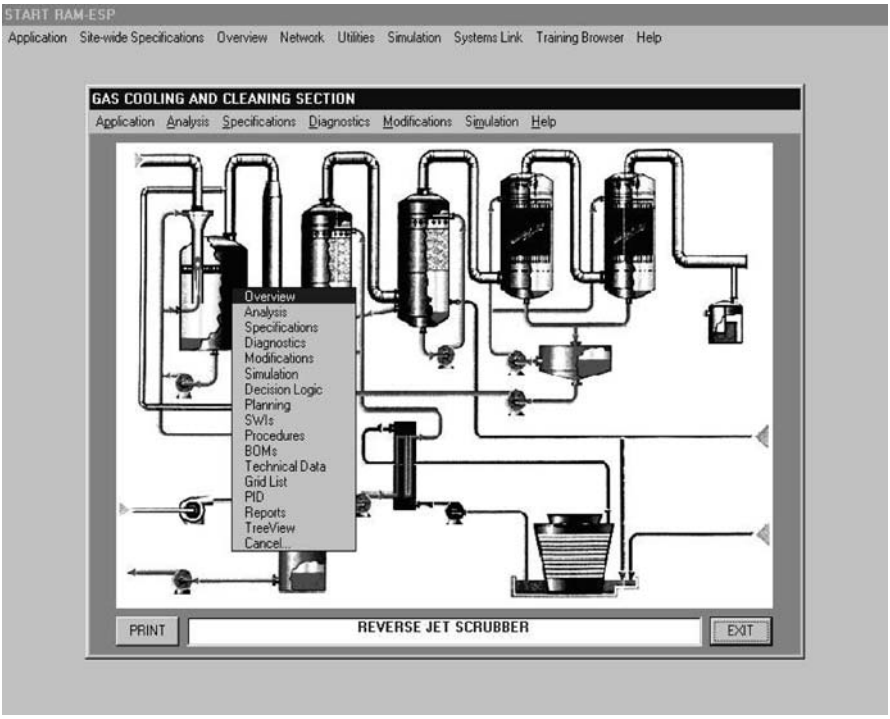


Fig. 3.57 Development list options for selected PFD system

- equipment physical data such as type, make, size, mass, volume, number of parts;
- equipment rating data such as performance, capacity, power (rating and factor), efficiency and output;
- equipment measure data such as rotation, speed, acceleration, governing, frequency and flow in volume and/or rate;
- equipment operating data such as pressures, temperatures, current (electrical), potential (voltage) and torque (starting and operational);
- equipment property data such as the type of enclosure, insulation, cooling, lubrication, and physical protection.

The *technical specification document* illustrated in Fig. 3.60 automatically formats the technical attributes relevant to each type of equipment that is selected in the design process. The document is structured into three sectors, namely:

- technical data obtained from the technical data worksheet, relevant to the equipment's physical and rating data, as well as performance measures and performance operating, and property attributes that are considered during the design process,
- technical specifications obtained from an assessment and evaluation of the required process and/or system design specifications,

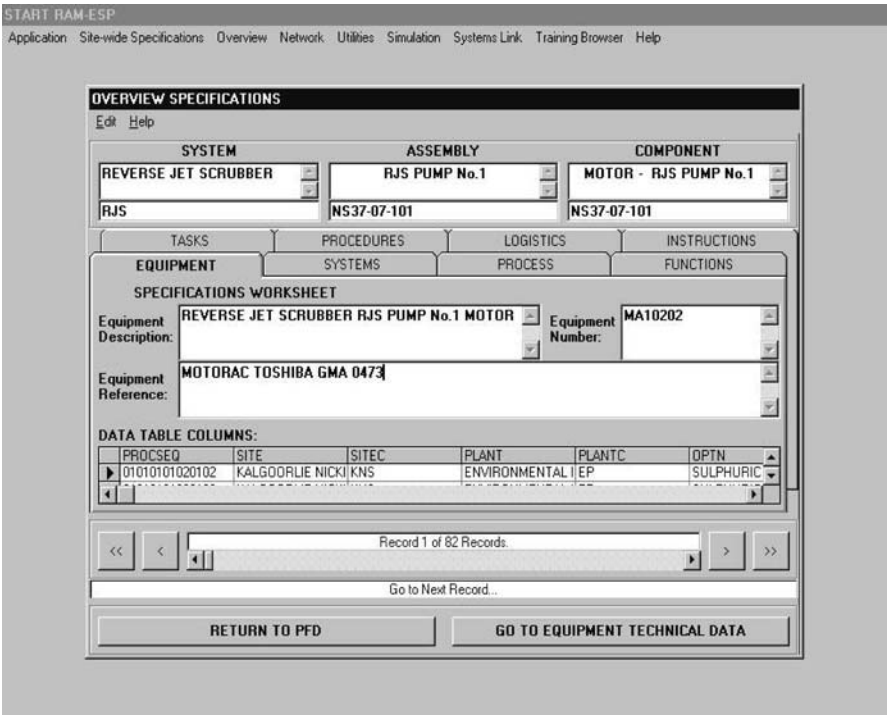


Fig. 3.58 Overview of selected equipment specifications

- acquisition data obtained from manufacturer/vendor data sheets, once the appropriate equipment technical specifications have been finalised during the detail design phase of the engineering design process.

The second category in the *RAMS analysis list* is the *analysis* option that enables selected users to access the major development tasks relative to the selected system of the section's PFD.

The options listed in the selection box in Fig. 3.61 appear after clicking on a selected system (in this case, the reverse jet scrubber), and include an analysis based on the following major development tasks:

Equipment (technical data sheets)	Tasks (maintenance/operational)
Systems (systems structures)	Procedures (reliability and safety)
Process (process characteristics)	Costs (parametric cost estimate risk)
Functions (physical/operational)	Strategy (operating/maintenance)
Conditions (physical/operational)	Logistics (critical/contract spares)
Criticality (consequence severity)	Instructions (safe work practices)

The major development tasks can be detailed into activities that constitute the overall RAMS analysis deliverables, not only to determine the integrity of engineering

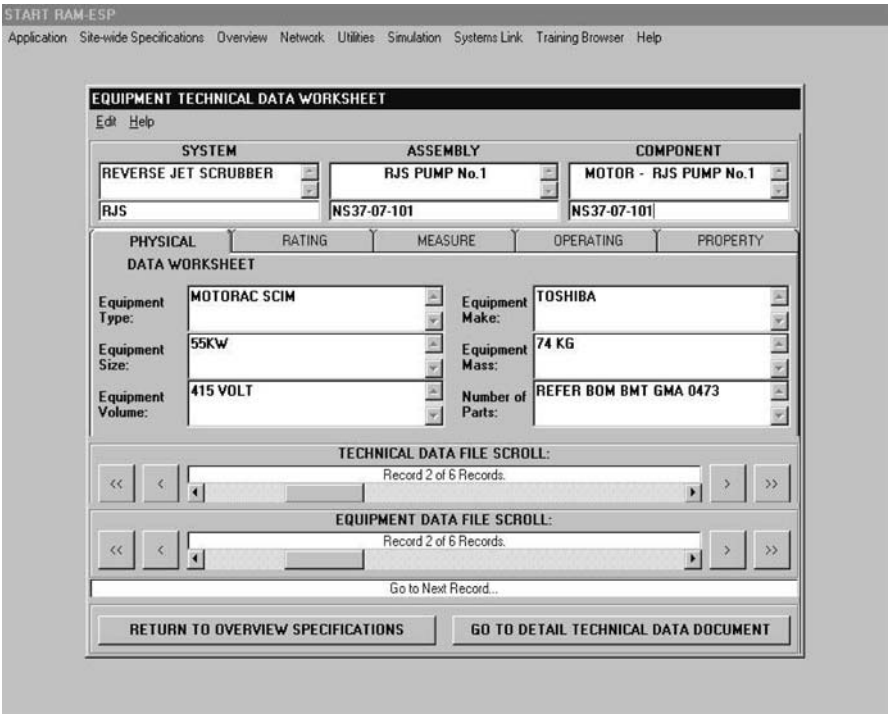


Fig. 3.59 Overview of the selected equipment technical data worksheet

design but also to verify and evaluate the commissioning of the plant. These tasks can also be applied sequentially in a RAMS analysis of process plant and general engineered installations that have been in operation for several years.

Some of these activities include the following:

- systems breakdown structure development,
- establishing equipment technical specifications,
- establishing process functional specifications,
- developing operating specifications,
- defining equipment function specifications,
- identifying failure characteristics and failure conditions,
- developing equipment fault diagnostics,
- developing equipment criticality,
- establishing equipment performance measures,
- identifying operating and maintenance tasks,
- developing operating procedures,
- developing maintenance procedures,
- establishing process cost models,
- developing operating and maintenance strategies,
- developing safe work practices,

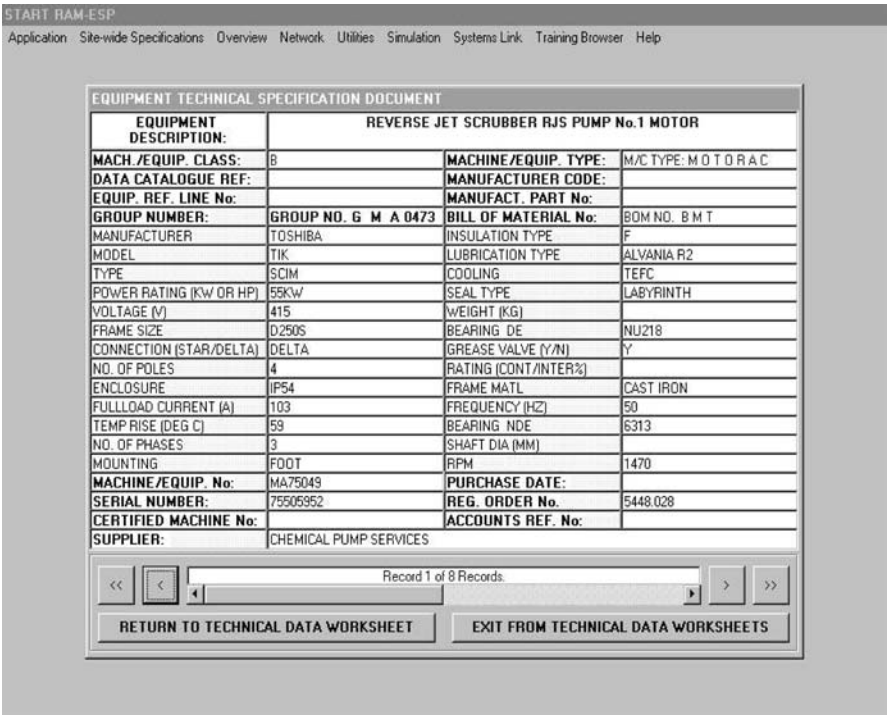


Fig. 3.60 Overview of the selected equipment technical specification document

- establishing standard work instructions,
- identifying critical spares,
- establishing spares requirements,
- providing for design modifications,
- simulating critical systems and processes.

The results of some of the more important activities will be considered in detail later, especially with respect to their correlation with the RAMS theory, and failure data that were obtained from the plant's distributed control system (DCS) operation and trip logs, 18 months after the plant was commissioned and placed into operation. The objective of the comparative analysis is to match the RAMS theory, specifically of systems and equipment criticality and reliability, with real-time operational data after plant start-up.

Analysis of selected functions of systems/assemblies/components is mainly a categorisation of functions into *operational functions* that are related to the item's working performance, and into *physical functions* that are related to the item's material design. The definition of function is given as "*the work that an item is designed to perform*". The primary purpose of *functions analysis* is to be able to define the failure of an item's function within specified limits of performance. This failure of an item's function is a failure of the work that the item is designed to perform, and

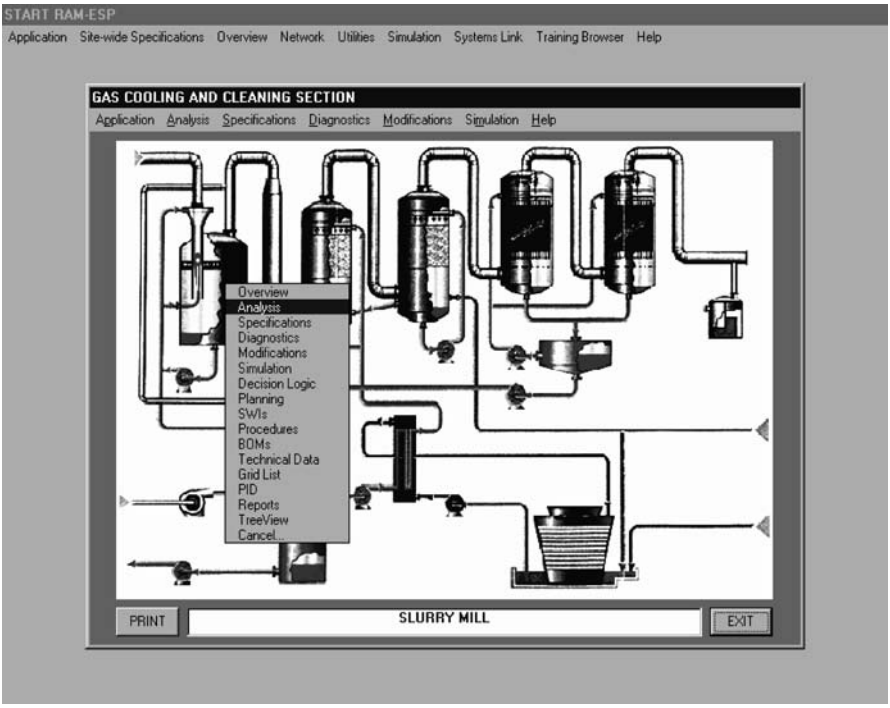


Fig. 3.61 Analysis of development tasks for the selected system

is termed a *functional failure*. Functional failure can thus be defined as "*the inability of an item to carry out the work that it is designed to perform within specified limits of performance*".

The result of functional failure can be assessed as either a complete loss of the item's function or a partial loss of the item's function. From these definitions it can be seen that a number of interrelated concepts have to be considered when defining functions in complex systems, and determining the *functional relationships* of the various items of a system (cf. Fig. 3.62).

The *functions* of a system and its related equipment (i.e. assemblies and components) can be grouped into two types, specifically *primary functions* and *secondary functions*. The primary function of a system considers the operational criteria of movement and work; thus, the primary function of the system is an operational function. The primary function of a system is therefore a concise description of the reason for existence of the system, based on the work it is required to perform. Primary functions for the sub-systems or assemblies that relate to the system's primary function must also be defined. It is at this level in the SBS where secondary functions are defined. Once the primary functions have been identified at the subsystem and assembly levels, the secondary functions are then defined, usually at component level (Fig. 3.63). Secondary functions can be both operational and physical, and relate back to the primary function of the sub-system or assembly. The

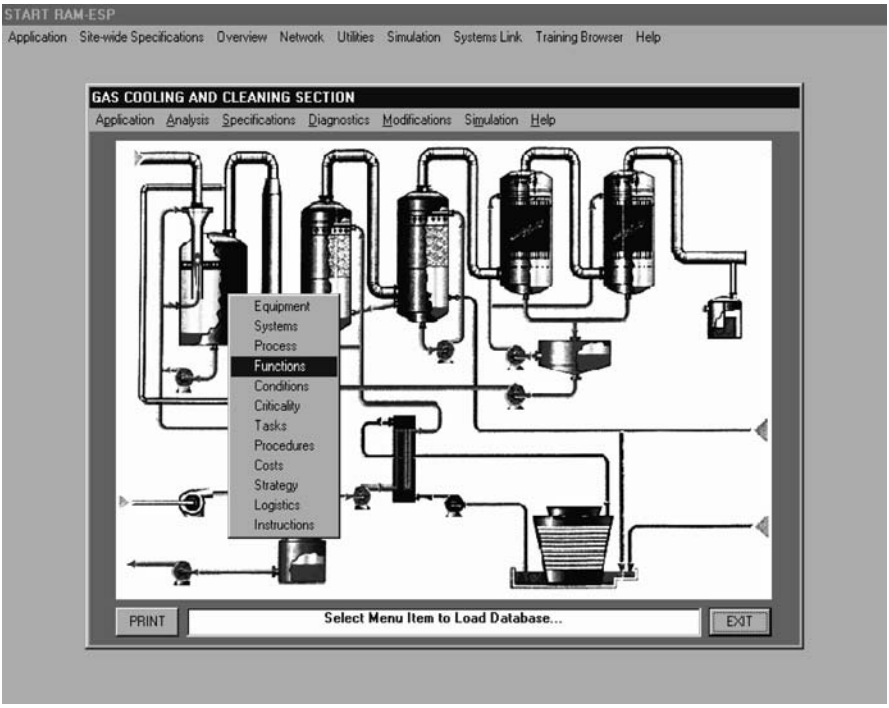


Fig. 3.62 Analysis of selected systems functions

secondary functions are related to the basic criteria of movement and work, or shape and consistency, depending on whether they are defined as operational or physical functions respectively.

The third category in the *RAMS analysis list* is the *specifications* option, which is similar to the overview option but with more drill-down access to the other activities in the program, and includes specifications as illustrated in Fig. 3.64 of selected major development tasks such as:

- Equipment specifications
- Systems specification
- Process specifications
- Function specifications
- Detailed tasks
- Detailed procedures
- Spares requirements
- Standard work instructions.

An engineering specification is an explicit set of design requirements to be satisfied by a material, product or service.

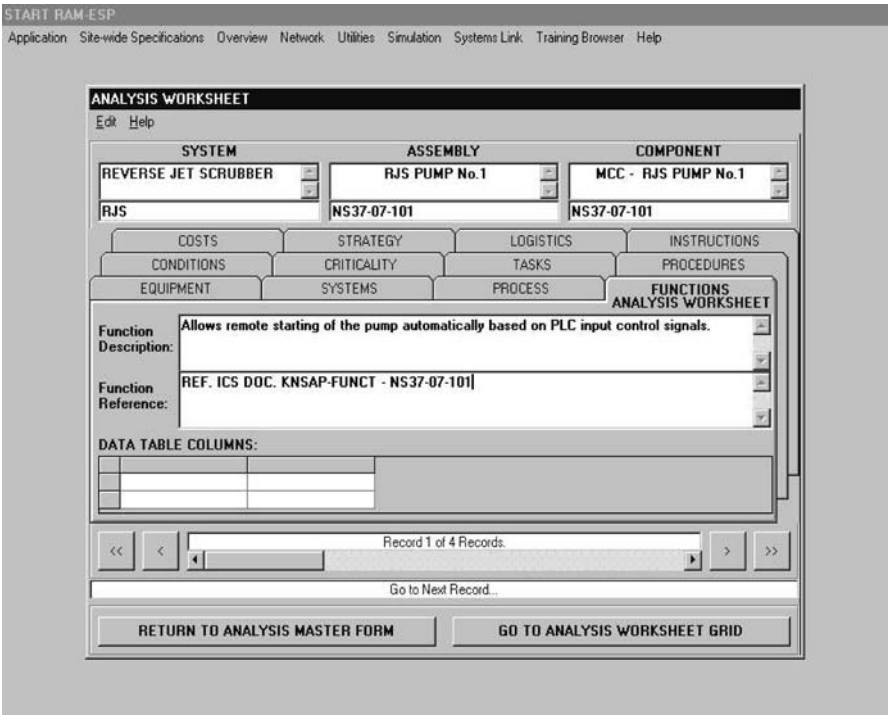


Fig. 3.63 Functions analysis worksheet of selected component

Typical engineering specifications might include the following:

- Descriptive title and scope of the specification.
- Date of last effective revision and revision designation.
- Person or designation responsible for questions on the specification updates, and deviations as well as enforcement of the specification.
- Significance or importance of the specification and its intended use.
- Terminology and definitions to clarify the specification content.
- Test methods for measuring all specified design characteristics.
- Material requirements: physical, mechanical, electrical, chemical, etc. targets and tolerances.
- Performance requirements, targets and tolerances.
- Certifications required for reliability and maintenance.
- Safety considerations and requirements.
- Environmental considerations and requirements.
- Quality requirements, inspections, and acceptance criteria.
- Completion and delivery.
- Provisions for rejection, re-inspection, corrective measures, etc.

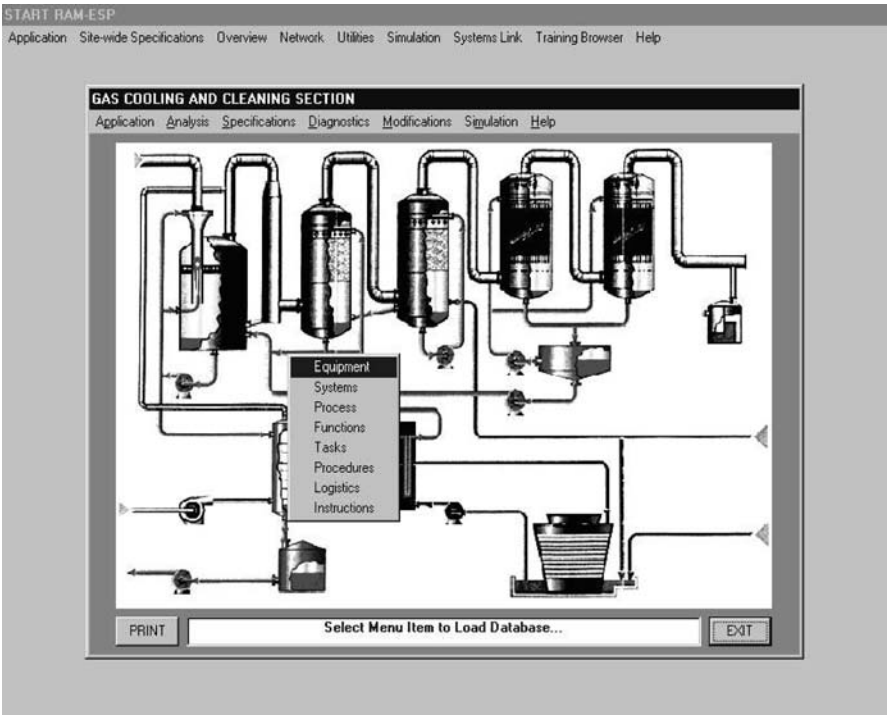


Fig. 3.64 Specifications of selected major development tasks

The specifications worksheet of selected equipment for consideration during the detail design phase of the engineering design process automatically integrates matched information pertaining to the equipment *type*, with respect to the following;

- equipment technical data and specifications, obtained from the technical data worksheet and technical specifications document,
- systems performance specifications relating to the specific process specifications,
- process performance specifications relating to the required design specifications,
- equipment functions specification relating to the basic functions from FMEA,
- typical required maintenance tasks and procedures specification from FMECA,
- the essential safety work instructions obtained from safety factor and risk analysis,
- installation logistical specifications with regard to the required contract warranty spares.

The *specifications worksheet* is a systems hierarchical layout of selected equipment, based on the outcome of the overall analysis of specifications of selected equipment for consideration during the detail design phase of the engineering design process. The worksheet (Fig. 3.65) is automatically generated, and serves as a systems-oriented pro-forma for electronically automated design reviews. Comprehensive design reviews are included at different phases of the engineering design

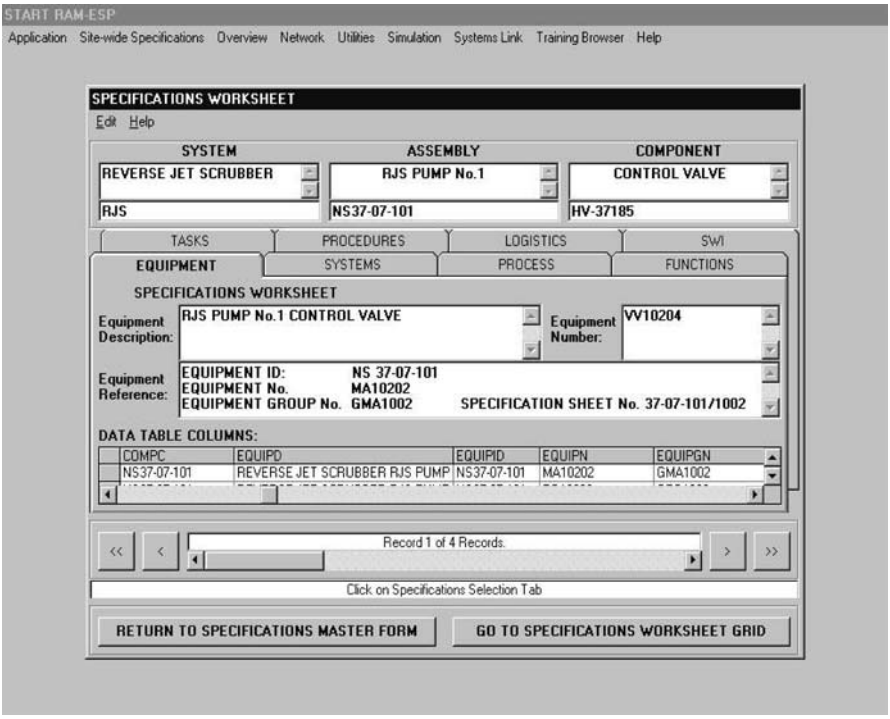


Fig. 3.65 Specifications worksheet of selected equipment

process, such as conceptual design, preliminary or schematic design, and final detail design. The concept of automated continual design reviews throughout the engineering design process is to a certain extent considered here, whereby the system allows for input of design data and schematics by remotely located multi-disciplinary groups of design engineers. However, it does not incorporate design implementation through knowledge-based expert systems, whereby each designed system or related equipment is automatically evaluated for integrity by the design group's expert system in an integrated *collaborative engineering design* environment.

The fourth category in the *RAMS analysis list* is the *diagnostics* option that enables the user to conduct a diagnostic review of selected major development tasks such as illustrated in Fig. 3.66:

- Systems and equipment condition
- Equipment hazards criticality
- Failure repair/replace costing
- Safety inspection strategies
- Critical spares requirement.

Typically, systems and equipment condition and hazards criticality analysis includes activities such as function specifications, failure characteristics and failure conditions, fault diagnostics, equipment criticality, and performance measures.

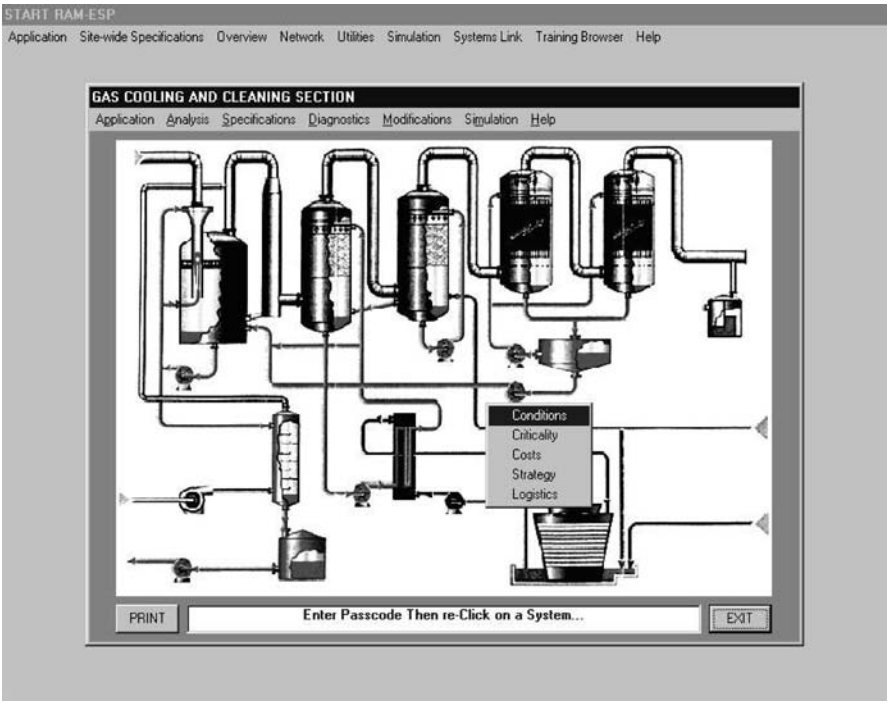


Fig. 3.66 Diagnostics of selected major development tasks

The following RAM analysis application model screens give detailed illustrations of a diagnostic analysis of selected major development tasks.

Condition diagnostics in engineering design relates to hazards criticality in the development of *failure modes and effects analysis (FMEA)*, and considers criteria such as system functions, component functional relationships, failure modes, failure causes, failure effects, failure consequences, and failure detection methods. These criteria are normally determined at the component level but the required operational specifications are usually identified at the sub-system or assembly level (Fig. 3.67).

Condition diagnostics, and related FMEA, should therefore theoretically be developed at the higher sub-system or assembly level in order to identify compliance with the operational specifications, and then to proceed with the development of FMEA at the component level, to determine potential failure criteria. In conducting the FMEA at the higher sub-system or assembly levels only, the possibility exists that some functional failures will not be considered, and the failure criteria will not be directed at some components that might be most applicable for design review.

It is necessary to conduct a *condition diagnostics*, and related FMEA, at the component level of the equipment SBS, since the failure criteria can be effectively identified only at this level, whereas for compliance to the required operational specifications, the results of the FMEA can be grouped to the sub-system or assembly levels. In practice, however, this can be substantially time consuming because a large

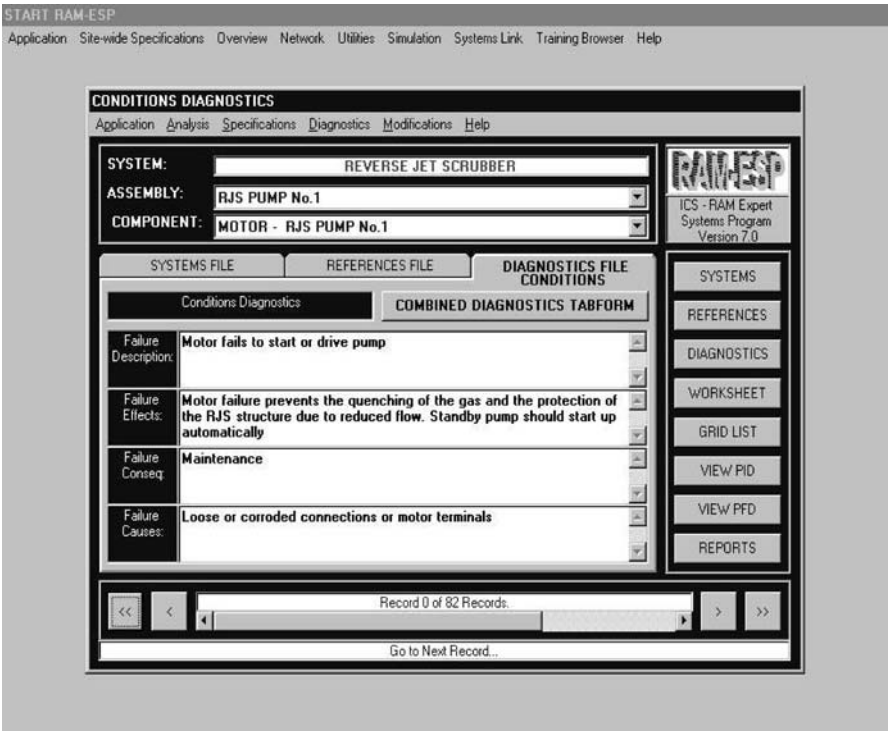


Fig. 3.67 Hazards criticality analysis assembly condition

portion of the FMEA results are very similar at both levels. Thus, in a *hazards criticality analysis* of the condition of selected components for inclusion in a design, the following component condition data illustrated in Fig. 3.68 are defined:

- Failure description
- Failure effects
- Failure consequences
- Failure causes.

Figure 3.68 illustrates a hazards criticality analysis of a common functional failure, "fails to open", of a HIPS control valve.

The *condition worksheet* in hazards criticality analysis is similar to the specifications worksheet of selected equipment for consideration during the detail design phase of the engineering design process, in that it automatically integrates matched information pertaining to the equipment *condition* and *criticality*, as illustrated in Fig. 3.69, with the necessary installation maintenance information concerning the following:

- Information from the equipment diagnostics worksheet relating to failure description, failure effects, failure consequences and failure causes

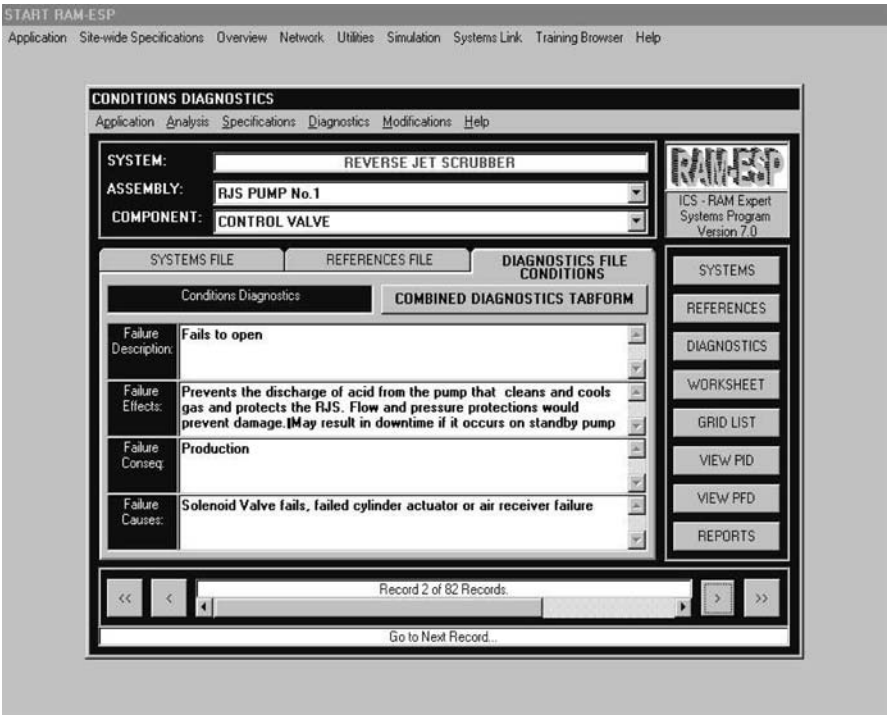


Fig. 3.68 Hazards criticality analysis component condition

- Information relating to equipment criticality
- Information relating to the necessary warranty maintenance strategy
- Information relating to the estimated required maintenance costs
- Information relating to the design's installation logistical support

The *hazards criticality analysis—condition spreadsheet* is a layout of selected components, based on the outcome of the condition worksheet of selected equipment for consideration during the detail design phase of the engineering design process. The condition spreadsheet (Fig. 3.70) is automatically generated, and serves as a FMEA pro-forma for electronically automated design reviews. The spreadsheet is variable, in that the data columns can be adjusted or hidden, but not deleted. These data columns include design integrity specification information such as failure description, failure mode, failure effects and consequences, as well as the relevant systems coding to identify the very many different elements of the systems breakdown structure (SBS) for equipment and spares acquisition during the manufacturing/construction stages, and for operations and maintenance procedure development during the warranty operations stages of the engineered installation. This design integrity specification information is automatically linked to the specific design process flow diagram (PFD) and pipe and instruments diagram (P&ID).

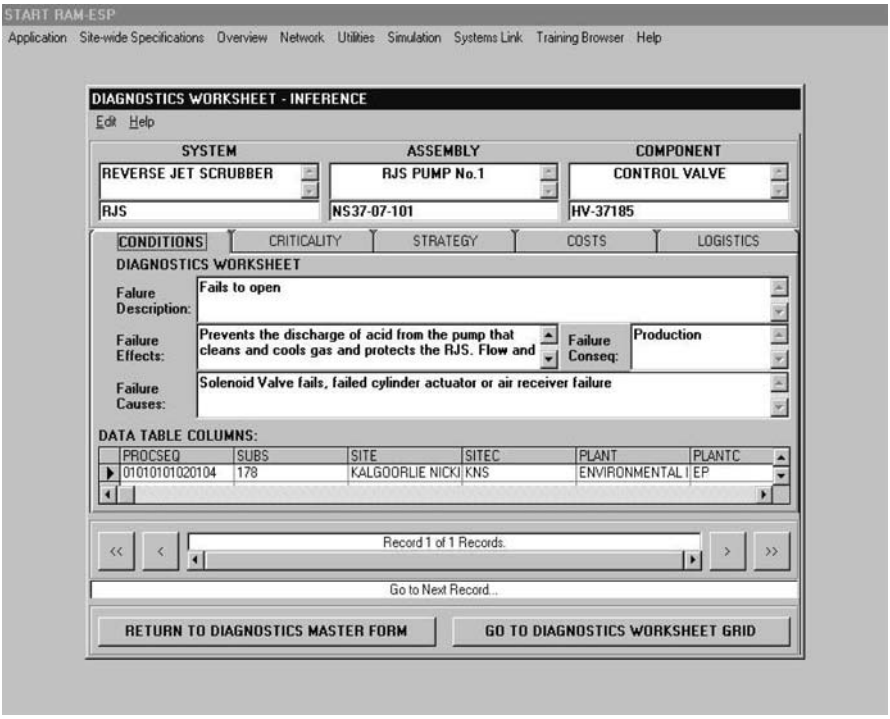


Fig. 3.69 Hazards criticality analysis condition diagnostic worksheet

The *criticality worksheet* in hazards criticality analysis automatically integrates matched information pertaining to equipment *criticality*, with equipment condition information and the necessary installation maintenance information of selected equipment for consideration during the detail design phase of the engineering design process. The information illustrated in Fig. 3.71 relates to FMECA and includes:

- Failure description
- Failure severity
- Consequence probability
- Risk of failure
- Yearly rate of failure
- Failure criticality.

The example in Fig. 3.71 is a typical hazards criticality analysis of a HIPS control valve showing failure severity and failure criticality.

The *hazards criticality analysis—criticality spreadsheet* is a layout of selected components, based on the outcome of the criticality worksheet of selected equipment for consideration during the detail design phase of the engineering design process. The criticality spreadsheet (Fig. 3.72) is automatically generated, and serves as a FMECA pro-forma for electronically automated design reviews. The spread-



Fig. 3.70 Hazards criticality analysis condition spreadsheet

sheet contains FMEA design integrity specification information such as the failure description, failure mode, failure effects and consequences, as well as the related failure downtime (including consequential damage), total downtime (repair time and damage), downtime costs for quality/injury losses, defects costs (material and labour costs per failure including damage), economic or production losses per failure, the probability of occurrence of the failure consequence (%), the failure rate or number of failures per year, the failure consequence severity, the failure consequence risk, the failure criticality, the total cost of failure per year and, finally, the overall failure criticality rating and the potential failure cost criticality rating.

The *hazards criticality analysis—strategy worksheet* automatically integrates matched information pertaining to the necessary warranty maintenance strategy of selected equipment for consideration during the detail design phase of the engineering design process, with equipment condition and criticality information, warranty maintenance costs and engineered installation logistical support information. The strategy information relates to FMECA and includes:

- Maintenance procedure description
- Maintenance procedure control
- Scheduled maintenance description
- Schedule maintenance control

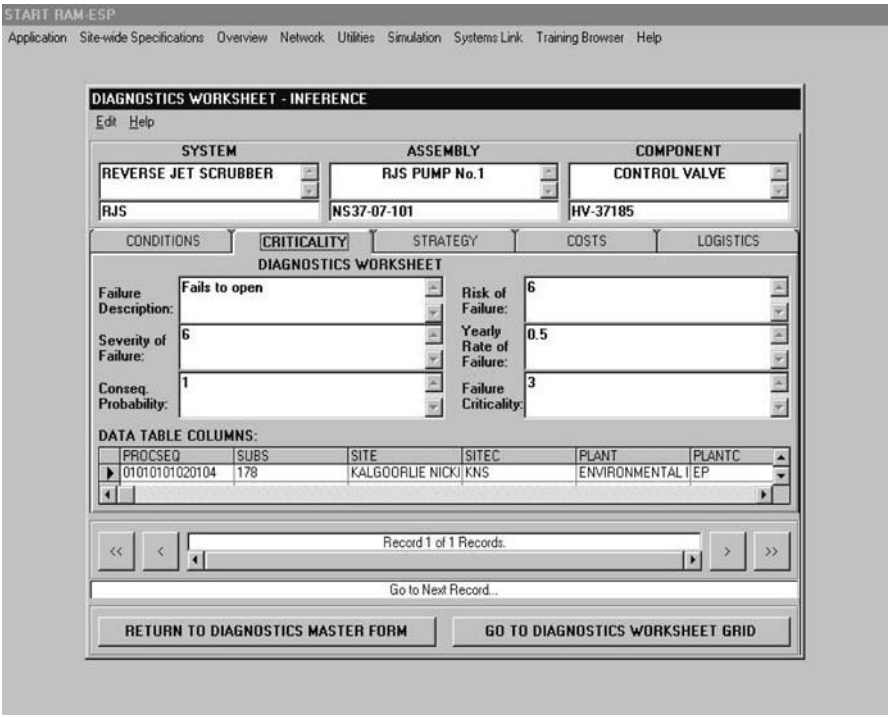


Fig. 3.71 Hazards criticality analysis criticality worksheet

- Scheduled maintenance frequency
- Schedule maintenance criticality.

Figure 3.73 illustrates a maintenance strategy worksheet for the HIPS control valve showing a *derived preventive maintenance strategy*.

The *hazards criticality analysis—strategy spreadsheet* is a layout of selected components, based on the outcome of the strategy worksheet of selected equipment for consideration during the detail design phase of the engineering design process. Similar to the criticality spreadsheet, the strategy spreadsheet (Fig. 3.74) is automatically generated, and serves as a FMECA pro-forma for electronically automated design reviews. The spreadsheet contains FMECA design integrity specification information such as the failure description, the relevant maintenance task description, the required maintenance craft type, the estimated frequency of the task, the maintenance procedure description (in which all the relevant maintenance tasks are grouped together, pertinent to the specific assembly and/or system that requires dismantling for a single task to be accomplished), the procedure identification coding, the grouped maintenance schedule (based on grouped tasks per procedure, and grouped procedures per system shutdown schedule), the maintenance schedule identification coding for computerised scheduling, and the overall planned downtime.

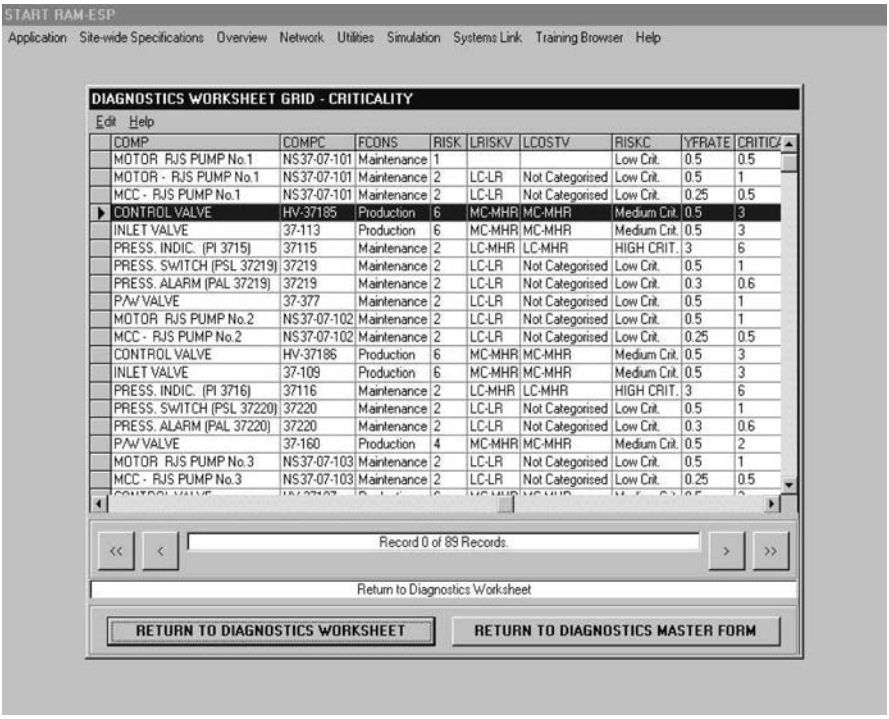


Fig. 3.72 Hazards criticality analysis criticality spreadsheet

The *hazards criticality analysis—costs worksheet* automatically integrates matched information pertaining to the necessary warranty maintenance costs of selected equipment for consideration during the detail design phase of the engineering design process, with equipment condition and criticality information, and the necessary warranty maintenance strategy and engineered installation logistical support information. The maintenance costs information relates to FMECA and includes the following:

- Estimated total costs per failure
- Estimated yearly downtime costs
- Estimated yearly maintenance labour costs
- Estimated yearly maintenance material costs
- Estimated yearly failure costs.

Figure 3.75 illustrates a maintenance costs for the HIPS control valve showing the *derived corrective maintenance costs and losses*.

The *hazards criticality analysis—costs spreadsheet* is a layout of selected components, based on the outcome of the costs worksheet of selected equipment for consideration during the detail design phase of the engineering design process. The spreadsheet (Fig. 3.76) is automatically generated, and serves as a FMECA pro-forma for electronically automated design reviews. The spreadsheet contains

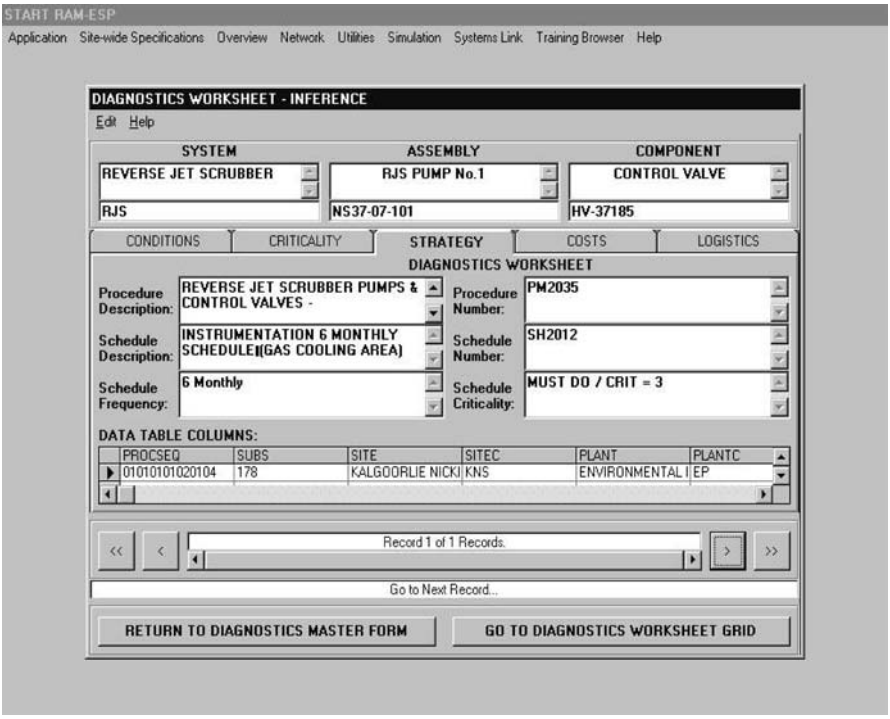


Fig. 3.73 Hazards criticality analysis strategy worksheet

FMECA design integrity specification information such as overall planned downtime, maintenance labour hours per task/procedure/schedule, the type of maintenance craft, the number of craft persons required, estimated maintenance material costs per task/procedure/schedule, the total maintenance downtime costs per task/procedure/schedule and, finally, the estimated total downtime costs per year, the estimated total maintenance labour costs per year, and the estimated total maintenance material costs per year. The summation of these estimated annual costs are then projected over a period of several years (usually 10 years) beyond the warranty operations period, based on estimates of declining early failures in stabilised operation.

The *hazards criticality analysis—logistics worksheet* automatically integrates matched information pertaining to the necessary logistical support of selected equipment for consideration during the detail design phase of the engineering design process, with equipment condition and criticality information, and the necessary warranty maintenance strategy and costs information. The logistical support information relates to FMECA and includes the following:

- Estimated required spares description
- Estimated required spares strategy
- Estimated spares BOM description

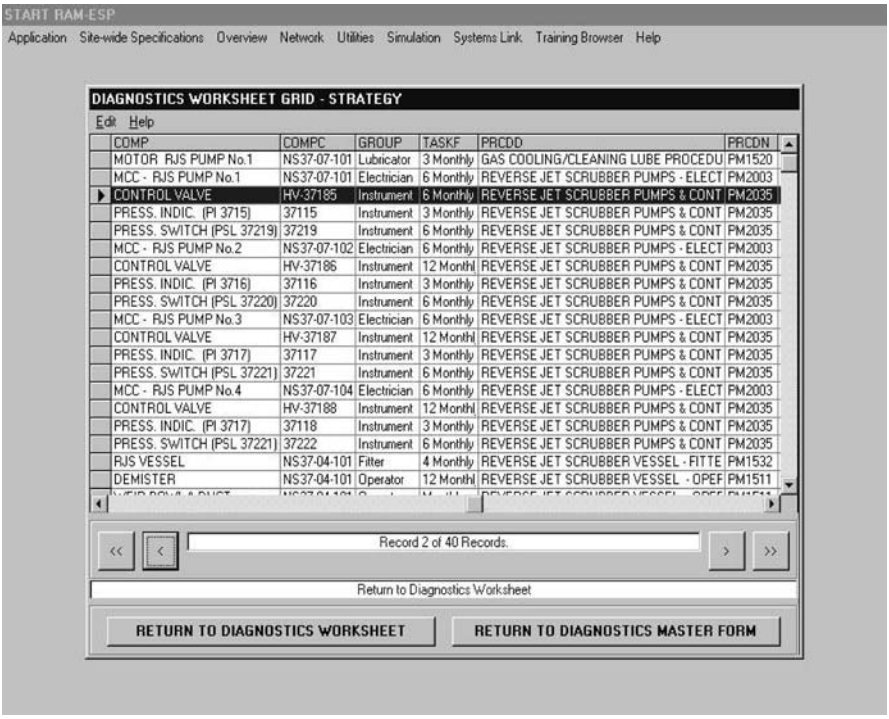


Fig. 3.74 Hazards criticality analysis strategy spreadsheet

- Estimated spares category
- Estimated spares costs.

Figure 3.77 illustrates spares requirements planning (SRP) for the HIPS control valve showing the *derived spares strategy*, *spares category for stores replenishment*, and recommended *bill of spares* (spares BOM).

The *hazards criticality analysis—logistics spreadsheet* is a layout of selected components, based on the outcome of the logistics worksheet of selected equipment for consideration during the detail design phase of the engineering design process. The spreadsheet (Fig. 3.78) is automatically generated, and serves as an FMECA pro-forma for electronically automated design reviews. The spreadsheet contains FMECA design integrity specification information such as the critical item of equipment requiring logistic support, the related spare parts by part description, the part identification number (according to the maintenance task code), parts specifications, parts quantities, the proposed manufacturer or supplier, the relevant manufacturer/supplier codes, the itemised stores description (for spare parts required for operations), the related bill of material (BOM) description and code for required stock items, the manufacturer's BOM description and code for non-stock items, the relevant manufacturer/supplier catalogue numbers and, finally, the estimated price per unit for the required spare parts.

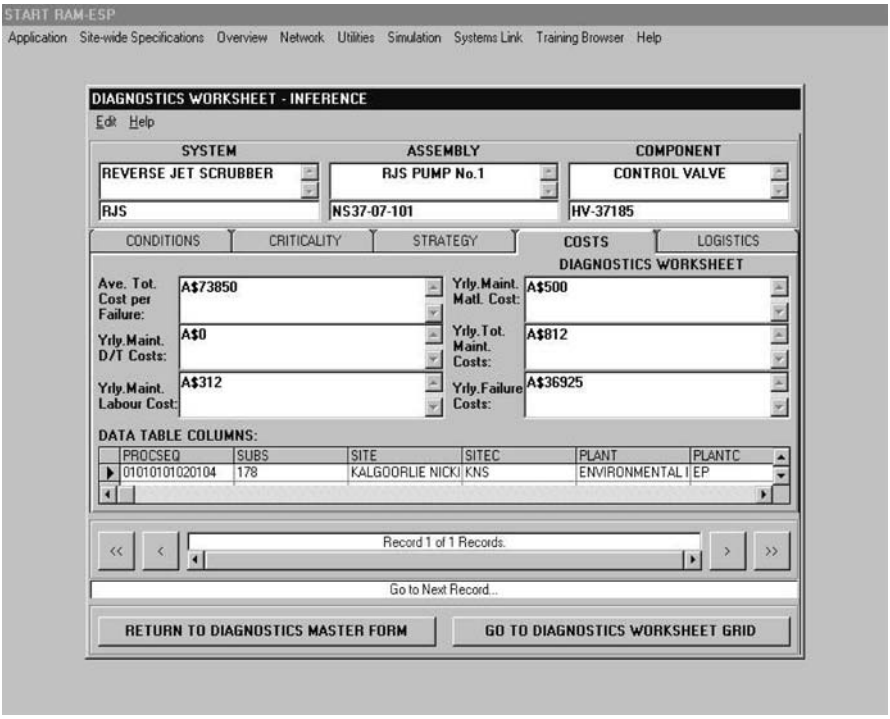


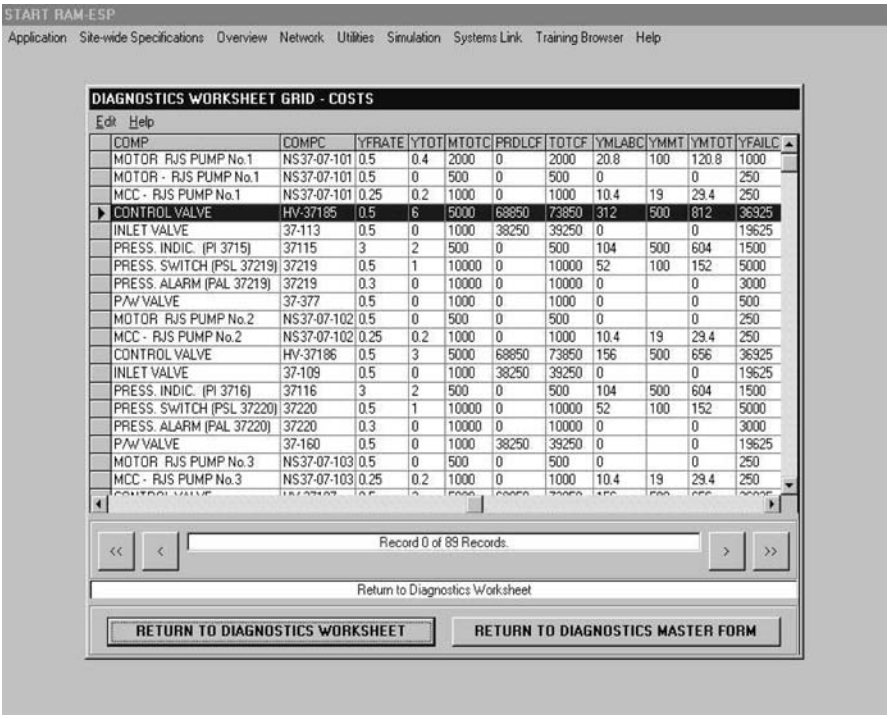
Fig. 3.75 Hazards criticality analysis costs worksheet

3.4.2 Evaluation of Modelling Results

a) Failure Modes and Effects Criticality Analysis

A case study FMEA was conducted on the environmental plant several months after completion of its design and installation where initially, prior to the design and construction of the plant, the process of sulphur dioxide to sulphuric acid conversion from a non-ferrous metal smelter emitted about 90 tonnes of sulphur gas into the environment per day, resulting in acid rain over a widespread area. The objective of the study was to determine the level of correlation between the design specifications and the actual installation's operational data, particularly with respect to systems criticality. The RAMS model initially captured the environmental plant's design criteria during design and commissioning of the plant, and was installed on the organisation intranet.

After a hierarchical structuring of the as-built systems into their assemblies and components, an FMEA was conducted, consisting mainly of identifying component failure descriptions, failure modes, failure effects, consequences and causes. Thereafter, a FMECA was conducted, which included an assessment of: the probability of occurrence of the *consequences of failure*, based on the relevant theory and



COMP	COMPC	YFRATE	YTOT	MTOTC	PRDLCF	TOTCF	YMLABC	YMMT	YMTOT	YFAILC
MOTOR RJS PUMP No.1	NS37-07-101	0.5	0.4	2000	0	2000	20.8	100	120.8	1000
MOTOR - RJS PUMP No.1	NS37-07-101	0.5	0	500	0	500	0		0	250
MCC - RJS PUMP No.1	NS37-07-101	0.25	0.2	1000	0	1000	10.4	19	29.4	250
CONTROL VALVE	HV-37185	0.5	6	5000	68850	73850	312	500	812	36925
INLET VALVE	37-113	0.5	0	1000	38250	39250	0		0	19625
PRESS. INDIC. (PI 3715)	37115	3	2	500	0	500	104	500	604	1500
PRESS. SWITCH (PSL 37219)	37219	0.5	1	10000	0	10000	52	100	152	5000
PRESS. ALARM (PAL 37219)	37219	0.3	0	10000	0	10000	0		0	3000
P/W VALVE	37-377	0.5	0	1000	0	1000	0		0	500
MOTOR RJS PUMP No.2	NS37-07-102	0.5	0	500	0	500	0		0	250
MCC - RJS PUMP No.2	NS37-07-102	0.25	0.2	1000	0	1000	10.4	19	29.4	250
CONTROL VALVE	HV-37186	0.5	3	5000	68850	73850	156	500	656	36925
INLET VALVE	37-109	0.5	0	1000	38250	39250	0		0	19625
PRESS. INDIC. (PI 3716)	37116	3	2	500	0	500	104	500	604	1500
PRESS. SWITCH (PSL 37220)	37220	0.5	1	10000	0	10000	52	100	152	5000
PRESS. ALARM (PAL 37220)	37220	0.3	0	10000	0	10000	0		0	3000
P/W VALVE	37-160	0.5	0	1000	38250	39250	0		0	19625
MOTOR RJS PUMP No.3	NS37-07-103	0.5	0	500	0	500	0		0	250
MCC - RJS PUMP No.3	NS37-07-103	0.25	0.2	1000	0	1000	10.4	19	29.4	250

Fig. 3.76 Hazards criticality analysis costs spreadsheet

analytic techniques previously considered, relating to uncertainty and probability assessment; the *failure rate* or number of failures per year, based on an extract of the failure records maintained by the installation's distributed control system (DCS; cf. Fig. 3.79); the *severity* of each failure consequence, based on the expected costs/loss of the failure consequence; the *risk* of the failure consequence, based on the product of the probability of its occurrence and its severity; the *criticality* of the failure, based on the failure rate and the failure's consequence severity; and the annual average *cost of failure*. From these FMEA and FMECA assessment values, a *failure criticality ranking* and potential *failure cost criticality* were established. The results of the case study presented in a failure modes and effects analysis (FMEA) and failure modes and effects criticality analysis (FMECA) are given in Tables 3.24 and 3.25. The results using the RAMS analysis model are shown in Figs. 3.80 through to 3.83. Only a very small portion (less than 1%) of the results of the FMEA is given in Table 3.24, Acid plant failure modes and effects analysis (ranking on criticality) and Table 3.25, Acid plant failure modes and effects criticality analysis, to serve as illustration.

Figure 3.79 illustrates a typical data sheet (in this case, of the reverse jet scrubber weak acid demister sprayers) in notepad format of the data accumulated by the installation's distributed control system (DCS).

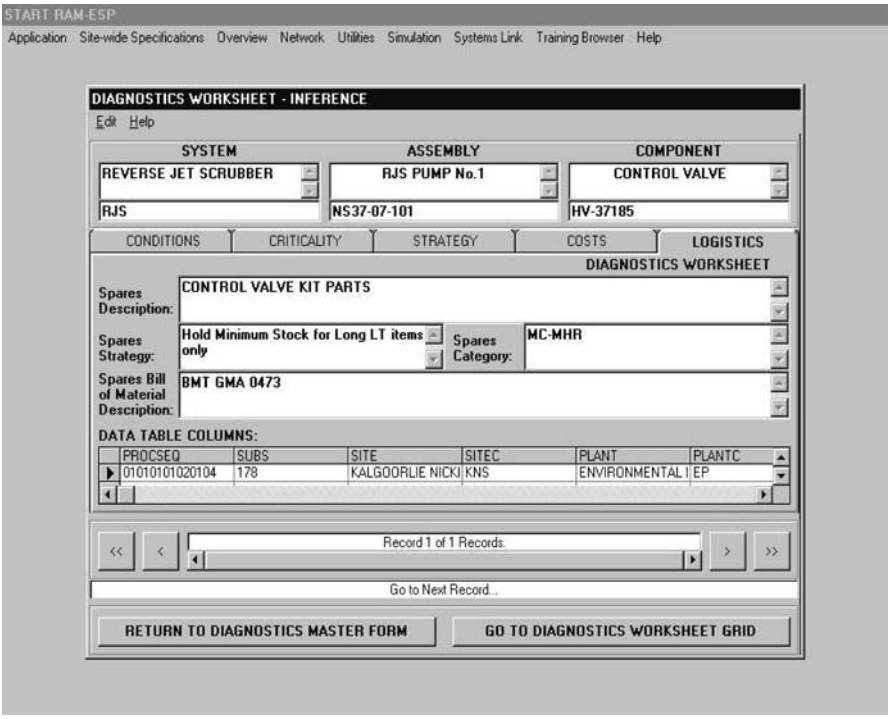


Fig. 3.77 Hazards criticality analysis logistics worksheet

Distributed control systems are dedicated systems used to control processes that are continuous or batch-oriented. A DCS is normally connected to sensors and actuators, and uses *set-point control* to control the flow of material through the plant. The most common example is a set-point control loop consisting of a pressure sensor, controller, and control valve. Pressure or flow measurements are transmitted to the controller, usually through the aid of a signal conditioning input/output (I/O) device. When the measured variable reaches a certain point, the controller instructs a valve or actuation device to open or close until the flow process reaches the desired set point. Programmable logic controllers (PLCs) have recently replaced DCSs, especially with SCADA systems.

A programmable logic controller (PLC), or programmable controller, is a digital computer used for automation of industrial processes. Unlike general-purpose controllers, the PLC is designed for multiple inputs and output arrangements, extended temperature ranges, immunity to electrical noise, and resistance to vibration and impact. PLC applications are typically highly customised systems, compared to specific custom-built controller design such as with DCSs. However, PLCs are usually configured with only a few analogue control loops; where processes require hundreds or thousands of loops, a DCS would rather be used. Data are obtained through a connected supervisory control and data acquisition (SCADA) system connected

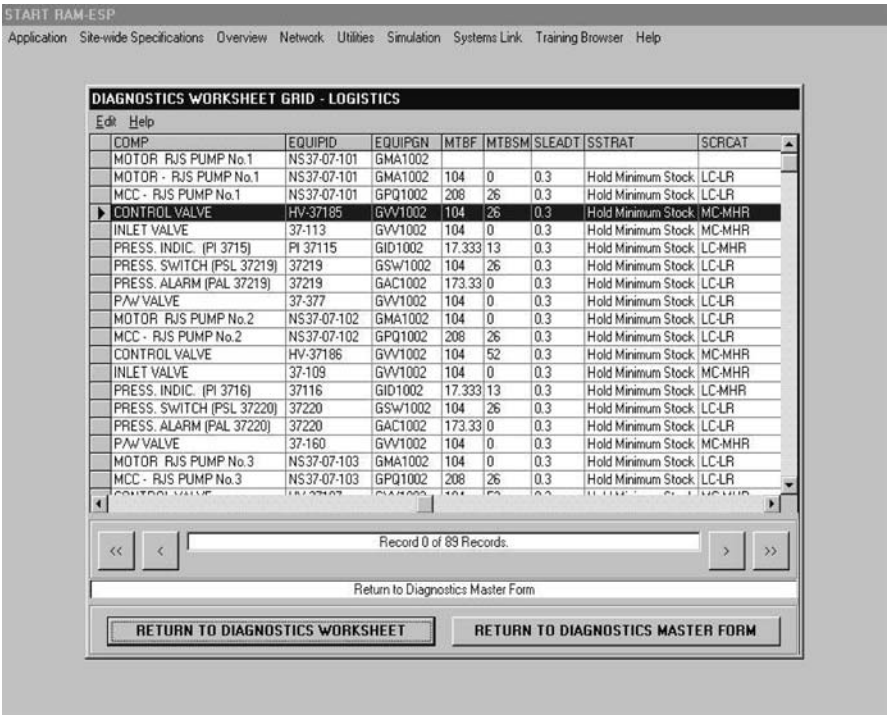


Fig. 3.78 Hazards criticality analysis logistics spreadsheet

to the DCS or PLC. The term SCADA usually refers to centralised systems that monitor and control entire plant, or integrated complexes of systems spread over large areas. Most site control is performed automatically by remote terminal units (RTUs) or by programmable logic controllers (PLCs). Host control functions are usually restricted to basic site overriding or supervisory level intervention. For example, a PLC may control the flow of cooling water through part of a process, such as the reverse jet scrubber, but the SCADA system allows operators to change the set points for the flow, and enables alarm conditions, such as loss of flow and high temperature, to be displayed and recorded. The feedback control loop passes through the RTU or PLC, while the SCADA system monitors the overall performance.

Using the SCADA data, a criticality ranking of the systems and their related assemblies was determined, which revealed that the highest ranking systems were the drying tower, hot gas feed, reverse jet scrubber, final absorption tower, and IPAT SO3 cooler. More specifically, the highest ranking critical assemblies and their related components of these systems were identified as the drying tower blowers' shafts, bearings (PLF) and scroll housings (TLF), the hot gas feed induced draft fan (PFC), the reverse jet scrubber's acid spray nozzles (TLF), the final absorption tower vessel and cooling fan guide vanes (TLF), and the IPAT SO3 cooler's cooling fan control vanes (TLF). These results were surprising, and further analysis was

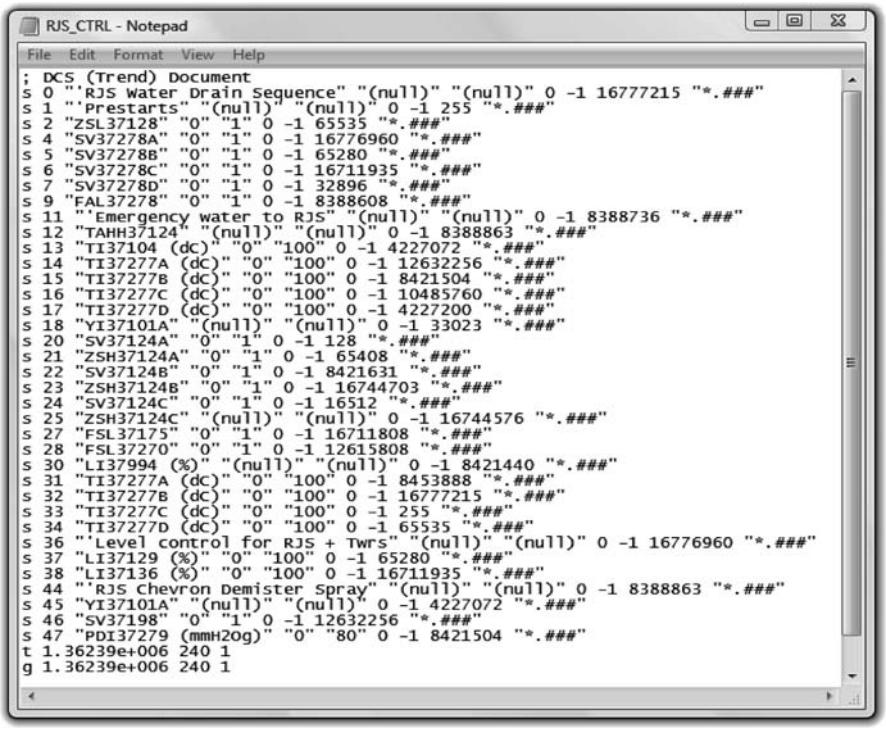


Fig. 3.79 Typical data accumulated by the installation's DCS

required to compare the results with the RAMS analysis design specifications. Despite an initial anticipation of non-correlation of the FMECA results with the design specifications, due to some modifications during construction, the RAM analysis appeared to be relatively accurate. However, further comparative analysis needed to be considered with each specific system hierarchy relating to the highest ranked systems, namely the drying tower, hot gas feed, reverse jet scrubber, final absorption tower, and IPAT SO3 cooler.

According to the design integrity methodology in the RAMS analysis, the *design specification FMECA* for the *drying tower* indicates an estimated criticality value of 32 for the no.1 SO2 blower scroll housing (TLF), which is the highest estimated value resulting in the topmost criticality ranking. The no.1 SO2 blower shaft seal (PLF) has a criticality value of 24, the shaft and bearings (PLF) a criticality value of 10, and the impeller (PLF) a criticality value of 7.5. From the FMECA case study extract given in Table 3.25, the topmost criticality ranking was determined as the drying tower blowers' shafts and bearings (PLF), and scroll housings (TLF) as 5th and 6th. The drying tower blowers' shaft seals (TLF) featured 9th and 10th, and the impellers did not feature at all.

Although the correlation between the RAMS analysis design specifications illustrated in Fig. 3.80 and the results of the case study is not quantified, a qualitative

Table 3.24 Acid plant failure modes and effects analysis (ranking on criticality)

System	Assembly	Component	Failure description	Failure mode	Failure effects	Failure consequences	Failure causes
Hot gas feed	Hot gas (ID) fan		Excessive vibration	PFC	Hot gas ID fan would trip on high vibration, as detected by any of four fitted vibration switches. Results in all gas directed to main stack	Production	Dirt accumulation on impeller due to excessive dust from ESPs
Reverse jet scrubber	Reverse jet scrubber	W/acid spray nozzles	Fails to deliver spray	TLF	Prevents the distribution of acid uniformly in order to provide protection to the RJS and cool the gases. Hot gas temp. exiting in RJS will be detected and shut down plant	Production	Nozzle blocks due to foreign materials in the weak acid supply or falls off due to incorrect installation
Drying tower	No.2 SO2 blower	Shaft & bearings	Fails to contain	PLF	No immediate effect but can result in equipment damage	Production	Leakage through seals due to breather blockage or seal joint deterioration
Drying tower	No.1 SO2 blower	Shaft & bearings	Excessive vibration	PFC	Can result in equipment damage and loss of acid production	Production	Loss of balance due to impellor deposits or permanent loss of blade material by corrosion/erosion
Drying tower	Drying tower		Restricted gas flow	PLF	Increased loading on SO2 blower	Production	Mist pad blockage due to ESP dust/chemical accumulation
Drying tower	No.1 SO2 blower	Scroll housing	Fails to contain	TLF	No effect immediate effect other than safety problem due to gas emission	Health hazard	Cracked housing due to operation above design temperature limits or restricted expansion
Drying tower	No.1 SO2 blower	Shaft seal	Fails to contain	TLF	No effect immediate effect other than safety problem due to gas emission	Health hazard	Carbon ring wear-out due to rubbing friction between shaft sleeve and carbon surface

Table 3.24 (continued)

System	Assembly	Component	Failure description	Failure mode	Failure effects	Failure consequences	Failure causes
Final absorb. tower	Final absorb. tower		Fails to absorb SO3 from the gas stream	TLF	Will result in poor stack appearance, loss in acid production and plant shutdown due to environmental reasons	Environment	Loss of absorbing acid flow or non uniform distribution of flow due to absorbing acid trough or header collapsing
Final absorb. tower	FAT cool. fan piping	Inlet guide vanes	Vanes fail to rotate	TLF	Loss of flow control leading to loss of efficiency of the FAT leading to possible SO2 emissions. This will lead to plant shutdown if the emissions are excessive or if temp. is $>220\ ^{\circ}C$	Environment	Seized adjustment ring due to roller guides worn or damaged due to lack of lubrication
Final absorb. tower	FAT cool. fan piping	Inlet guide vanes	Vanes fail to rotate	TLF	Loss of flow control leading to loss of efficiency of the FAT leading to possible SO2 emissions. This will lead to plant shutdown if the emissions are excessive or if temp. is $>220\ ^{\circ}C$	Environment	Seized vane stem sleeve due to deteriorated shaft stem sealing ring and ingress of chemical deposits
Final absorb. tower	FAT cool. fan piping	Inlet guide vanes	Operation outside limits of control	TLF	Loss of flow control leading to loss of efficiency of the FAT leading to possible SO2 emissions. This will lead to plant shutdown if the emissions are excessive or if temp. is $>220\ ^{\circ}C$	Environment	Loose or incorrectly adjusted vane link pin due to incorrect installation process or over-stroke condition
I/P absorb. tower	I/PASS absorb. tower		Fails to absorb SO3 from the gas stream	TLF	Will result in additional loading of converter 4th pass and final absorbing tower with possible stack emissions	Environment	Loss of absorbing acid flow due to absorbing acid trough or header collapsing

Table 3.24 (continued)

System	Assembly	Component	Failure description	Failure mode	Failure effects	Failure consequences	Failure causes
Drying tower	Drying tower		Fails to remove moisture from the gas stream	TLF	Will result in blower vibration problems, deterioration of catalyst and loss of acid production	Quality	Damage, blockage or dislodged mist pad due to high temp./excessive inlet gas flow, or gas quality
Drying tower	Drying tower		Fails to remove moisture from the gas stream	TLF	Will result in blower vibration problems, deterioration of catalyst and loss of acid production	Quality	Damage, blockage or dislodged mist pad due to improper installation of filter pad retention ring
IPAT SO3 cooler	SO3 cool. fan piping	Inlet guide vanes	Vanes fail to rotate	TLF	Loss of IPAT efficiency due to poor temperature control of the gas stream. Temperature control loop would cut gas supply if gas discharge temperature at IPAT cooler too high	Quality	Seized adjustment ring due to roller guides worn or damaged due to lack of lubrication
IPAT SO3 cooler	SO3 cool. fan piping	Inlet guide vanes	Vanes fail to rotate	TLF	Loss of IPAT efficiency due to poor temperature control of the gas stream. Temperature control loop would cut gas supply if gas discharge temperature at IPAT cooler too high	Quality	Seized vane stem sleeve due to worn shaft stem sealing ring and ingress of chemical deposits
IPAT SO3 cooler	SO3 cool. fan piping	Inlet control vanes	Operation outside limits of control	TLF	Loss of IPAT efficiency due to poor temperature control of the gas stream. Temperature control loop would cut gas supply if gas discharge temperature at IPAT cooler too high	Quality	Loose or incorrectly adjusted vane link pin due to incorrect installation process or over-stroke condition

Table 3.25 Acid plant failure modes and effects criticality analysis

System	Assembly	Component	Failure consequences	Probability	Failures/ year	Severity	Risk	Crit. value	Failure cost/year	Crit. rate	Fail cost
Drying tower	No.1 SO2 blower	Shaft & bearings	Production	100%	12	5	5.0	60.0	$287,400	High crit.	High cost
Drying tower	No.2 SO2 blower	Shaft & bearings	Production	100%	12	5	5.0	60.0	$287,400	High crit.	High cost
Hot gas feed	Hot gas (ID) fan		Production	100%	12	4	4.0	48.0	$746,400	High crit.	High cost
Reverse jet scrubber	Reverse jet scrubber	W/acid spray nozzles	Production	100%	6	6	6.0	36.0	$465,000	High crit.	High cost
Drying tower	No.1 SO2 blower	Scroll housing	Health hazard	80%	4	10	8.0	32.0	$1,235,600	High crit.	High cost
Drying tower	No.2 SO2 blower	Scroll housing	Health hazard	80%	4	10	8.0	32.0	$1,235,600	High crit.	High cost
Drying tower	No.1 SO2 blower	Shaft & bearings	Production	100%	7	4	4.0	28.0	$449,400	High crit.	High cost
Drying tower	No.2 SO2 blower	Shaft & bearings	Production	100%	7	4	4.0	28.0	$449,400	High crit.	High cost
Drying tower	No.1 SO2 blower	Shaft seal	Health hazard	80%	3	10	8.0	24.0	$366,300	High crit.	High cost
Drying tower	No.2 SO2 blower	Shaft seal	Health hazard	80%	3	10	8.0	24.0	$366,300	High crit.	High cost
Drying tower	Drying tower		Quality	80%	4	7	5.6	22.4	$620,200	High crit.	High cost
IPAT SO3 cooler	SO3 cool. fan piping	Inlet guide vanes	Quality	100%	3	7	7.0	21.0	$219,600	High crit.	High cost
IPAT SO3 cooler	SO3 cool. fan piping	Inlet control vanes	Quality	100%	3	7	7.0	21.0	$215,100	High crit.	High cost
I/P absorb. tower	I/PASS absorb. tower		Environment	60%	4	8	4.8	19.2	$915,600	High crit.	High cost
Final absorb. tower	FAT cool. fan piping		Environment	80%	3	8	6.4	19.2	$216,600	High crit.	High cost

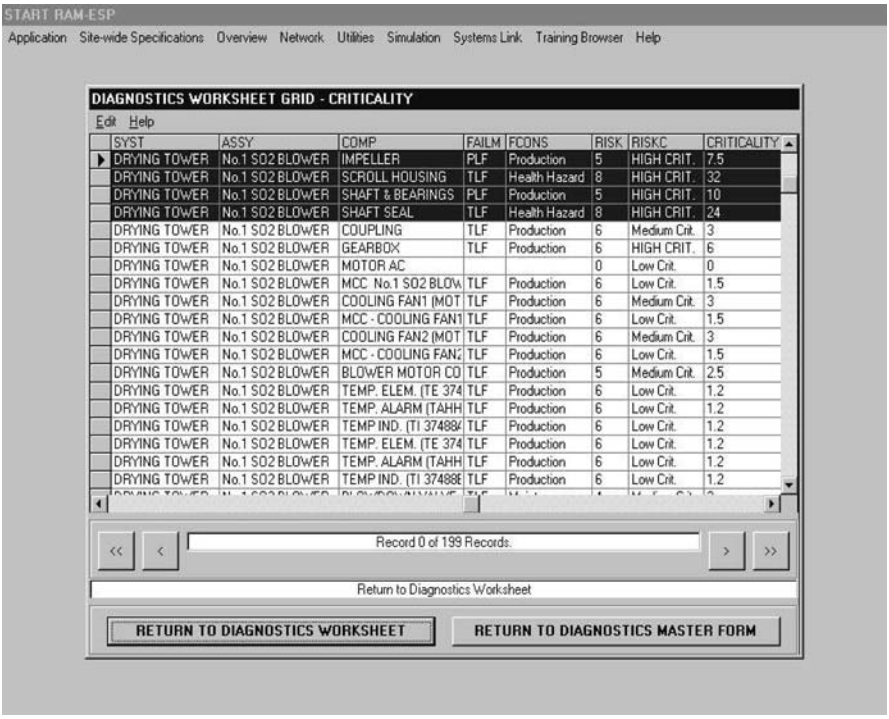


Fig. 3.80 Design specification FMECA—drying tower

assessment of the design integrity methodology of the RAMS analysis can be described as *accurate*.

The RAMS analysis *design specification FMECA* for the *hot gas feed* indicates an estimated criticality value of 6 for both the SO2 gas duct pressure transmitter and temperature transmitter. From the FMECA case study extract given in Table 3.25, the criticality for the hot gas feed's induced draft fan (PFC) ranked 3rd out of the topmost 15 critical items of equipment, whereas the design specification FMECA ranked the induced draft fan (PFC) as a mere 3, which is not illustrated in Fig. 3.81. The hot gas feed's SO2 gas duct pressure and temperature transmitters, illustrated in Fig. 3.81, had a criticality rank of 6, whereas they do not feature in the FMECA case study extract given in Table 3.25.

Although this does indicate some vulnerability of accuracy in the assessment and evaluation of design integrity at the lower levels of the systems breakdown structure (SBS), especially with respect to an assessment of the critical failure mode, the identification of the hot gas feed induced draft fan as a high failure critical and high cost critical item of equipment is valid.

The RAMS analysis *design specification FMECA* for the *reverse jet scrubber* indicates an estimated criticality value of 6 for both the RJS pumps' pressure indicators. From the FMECA case study extract given in Table 3.25, the criticality for

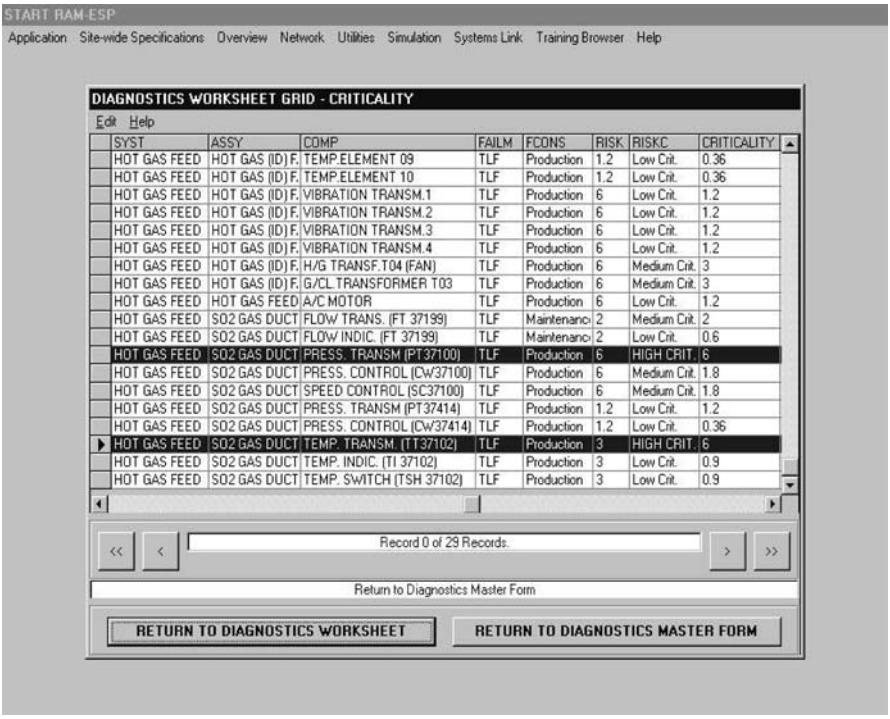


Fig. 3.81 Design specification FMECA—hot gas feed

the reverse jet scrubber's acid spray nozzles (TLF) ranked 4th out of the topmost 15 critical items of equipment, whereas the design specification FMECA ranked the acid spray nozzles (TLF) as 4.5, which is not illustrated in Fig. 3.82. Similar to the hot gas feed system, this again indicates some vulnerability of accuracy in the assessment and evaluation of design integrity at the lower levels of the systems breakdown structure (SBS), especially with respect to an assessment of the critical failure mode.

The identification of the reverse jet scrubber's pumps as a high failure critical item of equipment (with respect to pressure instrumentation), illustrated in Fig. 3.82, is valid, as the RJS pumps have a reliable design configuration of 3-up with two operational and one standby.

The RAMS analysis *design specification FMECA* for the *final absorption tower* indicates an estimated criticality value of 2.475, as illustrated in Fig. 3.83, which gives a criticality rating of medium criticality. The highest criticality for components of the final absorption tower system is 4.8, which is for the final absorption tower temperature instrument loop. From the FMECA case study criticality ranking given in Table 3.25, the final absorption tower ranked 15th out of the topmost 15 critical items of equipment, whereas the design specification FMECA does not list the final absorption tower as having a high criticality.

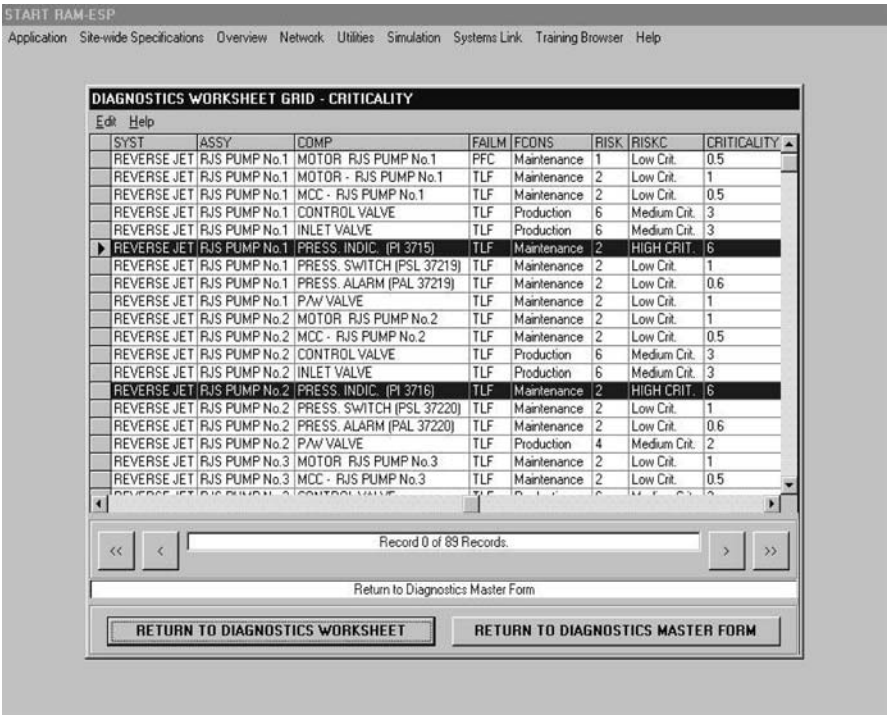


Fig. 3.82 Design specification FMECA—reverse jet scrubber

Similar to the hot gas feed system and the reverse jet scrubber system, this once more indicates some vulnerability of accuracy in the assessment and evaluation of design integrity at the lower levels of the systems breakdown structure (SBS). However, the identification of the final absorption tower as a critical system in the RAMS design specification was verified by an evaluation of the plant's failure data.

b) Failure Data Analysis

Failure data in the form of time (in days) before failure of the critical systems (drying tower, hot gas feed, reverse jet scrubber, final absorption tower, and IPAT SO3 cooler) were accumulated over a period of 2 months. These data are given in Table 3.26, which shows acid plant failure data (repair time RT and time before failure TBF) obtained from the plant's distributed control system.

A Weibull distribution fit to the data produces the following results:

Acid plant failure data statistical analysis

Number of failures $= 72$
Number of suspensions $= 0$

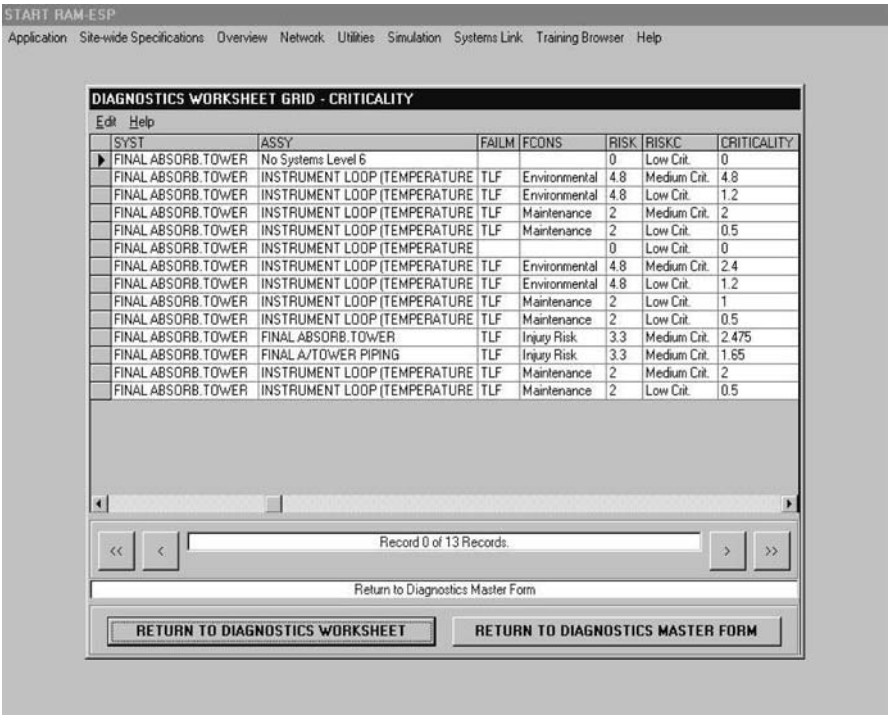


Fig. 3.83 Design specification FMECA—final absorption tower

Total failures + suspensions $= 72$
Mean time to failure (MTTF) $= 2.35$ (days)

The Kolmogorov–Smirnov goodness-of-fit test The Kolmogorov–Smirnov (K–S) test is used to decide if a sample comes from a population with a specific distribution. The K–S test is based on the empirical distribution function (e.c.d.f.) whereby, given N *ordered* data points $Y_1, Y_2, \ldots Y_N$, the e.c.d.f. is defined as:

$$\mathrm{E}N = n(i)/N\,, \tag{3.212}$$

where $n(i)$ is the number of points less than Y_i, and the Y_i are ordered from smallest to largest value. This is a step function that increases by $1/N$ at the value of each ordered data point. An attractive feature of this test is that the distribution of the K–S test statistic itself does not depend on the cumulative distribution function being tested. Another advantage is that it is an exact test; however, the goodness-of-fit test depends on an adequate sample size for the approximations to be valid. The K–S test has several important limitations, specifically:

- It applies only to continuous distributions.
- It tends to be more sensitive near the centre of the distribution than at the tails.

Table 3.26 Acid plant failure data (repair time RT and time before failure TBF)

Failure time	RT (min)	TBF (day)	Failure time	RT (min)	TBF (day)	Failure time	RT (min)	TBF (day)
7/28/01 0:00	38	0	9/25/01 0:00	31	5	11/9/01 0:00	360	1
7/30/01 0:00	35	2	9/27/01 0:00	79	2	11/10/01 0:00	430	1
7/31/01 0:00	148	1	9/29/01 0:00	346	2	11/20/01 0:00	336	10
8/1/01 0:00	20	1	9/30/01 0:00	80	1	11/26/01 0:00	175	6
8/5/01 0:00	27	4	10/1/01 0:00	220	1	11/28/01 0:00	118	2
8/7/01 0:00	15	2	10/4/01 0:00	63	3	12/1/01 0:00	35	3
8/11/01 0:00	5	4	10/7/01 0:00	176	3	12/2/01 0:00	556	1
8/12/01 0:00	62	1	10/8/01 0:00	45	1	12/5/01 0:00	998	3
8/13/01 0:00	580	1	10/10/01 0:00	52	2	12/6/01 0:00	124	1
8/14/01 0:00	897	1	10/10/01 0:00	39	0	12/11/01 0:00	25	5
8/15/01 0:00	895	1	10/11/01 0:00	55	1	12/12/01 0:00	120	1
8/16/01 0:00	498	1	10/12/01 0:00	36	1	12/17/01 0:00	35	5
8/17/01 0:00	308	1	10/14/01 0:00	10	2	12/26/01 0:00	10	9
8/19/01 0:00	21	2	10/18/01 0:00	1,440	4	1/2/02 0:00	42	7
8/21/01 0:00	207	2	10/19/01 0:00	590	1	1/18/02 0:00	196	16
8/22/01 0:00	346	1	10/22/01 0:00	43	3	1/29/02 0:00	22	11
8/23/01 0:00	110	1	10/24/01 0:00	107	2	2/9/02 0:00	455	11
8/25/01 0:00	26	2	10/29/01 0:00	495	5	2/10/02 0:00	435	1
8/28/01 0:00	15	3	10/30/01 0:00	392	1	2/13/02 0:00	60	3
9/4/01 0:00	41	7	10/31/01 0:00	115	1	2/13/02 0:00	30	0
9/9/01 0:00	73	5	11/1/01 0:00	63	1	2/17/02 0:00	34	4
9/12/01 0:00	134	3	11/2/01 0:00	245	1	2/24/02 0:00	71	7
9/19/01 0:00	175	7	11/4/01 0:00	40	2	3/4/02 0:00	18	8
9/20/01 0:00	273	1	11/8/01 0:00	50	4	3/9/02 0:00	23	5

- The distribution must be fully specified—that is, if location, scale, and shape parameters are estimated from the data, the critical region of the K–S test is no longer valid, and must be determined by Monte Carlo (MC) simulation.

Goodness-of-fit results The K–S test result of the acid plant data given in Table 3.26 is the following:

Kolmogorov-Smirnov (D) statistic	= 347			
Modified D statistic	= 2.514			
Critical value of modified D	= 1.094			
Confidence levels	= 90%	95%	97.5%	99%
Tabled values of K–S statistic	= 0.113	0.122	0.132	0.141
Observed K–S statistic	= 325			
Mean absolute prob. error	= 0.1058			
Model accuracy	= 89.42% (poor)			

The hypothesis that the data fit the two-Weibull distribution is rejected with 99% confidence.

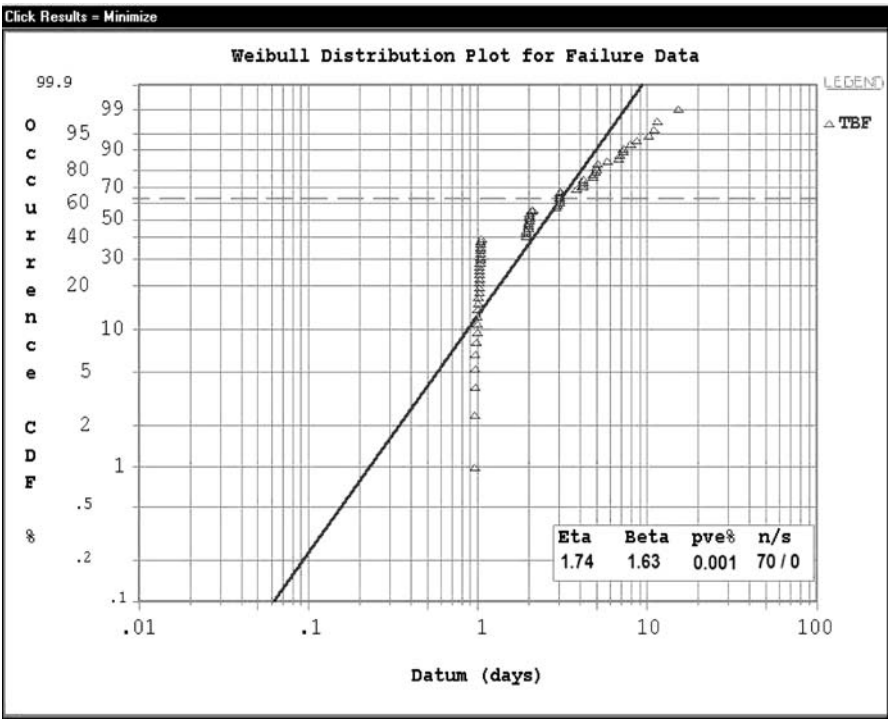


Fig. 3.84 Weibull distribution chart for failure data

Three-parameter Weibull fit—ungrouped data (Fig. 3.84):

Minimum life	= 0.47 (days)
Shape parameter BETA	= 1.63
Scale parameter ETA	= 1.74 (days)
Mean life	= 2.03 (days)
Characteristic life	= 2.21 (days)
Standard deviation	= 0.98 (days)

Test for random failures The hypothesis that failures are random is rejected at 5% level.

3.4.3 Application Modelling Outcome

The acid plant failure data do not suitably fit the Weibull distribution, with 89% model accuracy. However, the failures are not random (i.e. the failure rate is not constant), and it is essential to determine whether failures are in the early phase or in the wear-out phase of the plant's life cycle—especially so soon after its installa-

tion (less than 24 months). The distribution must be fully specified—that is, the K–S test is no longer valid, and must be determined by Monte Carlo (MC) simulation. However, prior to simulation, a closer definition of the source of most of the failures of the critical systems (determined through the case study FMECA) is necessary. Table 3.27 shows the total downtime of the acid plant's critical systems. The downtime failure data grouping indicates that the highest downtime is due to the hot gas feed induced draft fan, then the reverse jet scrubber, the drying tower blowers, and final absorption.

Engineered Installation Downtime

Table 3.27 Total downtime of the environmental plant critical systems

Downtime reason description	Total hours	Direct hours	Indirect hours
Hot gas feed, hot gas fan total	1,514	1,388	126
Gas cleaning, RJS total	680	581	99
Drying tower, SO2 blowers total	496	248	248
Gas absorption, final absorption total	195	100	95
Total	*2,885*	*2,317*	*568*

Monte Carlo simulation With the K–S test, the distribution of the failure data must be fully specified—that is, if location, scale and shape parameters are estimated from the data, the critical region of the K–S test is no longer valid, and must be determined by Monte Carlo (MC) simulation.

MC simulation emulates the chance variations in the critical systems' time before failure (TBF) by generating random numbers that form a uniform distribution that is used to select values from the sample TBF data, and for which various TBF values are established to develop a large population of representative sample data. The model then determines if the representative sample data come from a population with a specific distribution (i.e. exponential, Weibull or gamma distributions). The outcome of the M C simulation gives the following distribution parameters (Tables 3.28 and 3.29):

Time Between Failure Distribution

Table 3.28 Values of distribution models for time between failure

Distribution model	Parameter	Parameter value
1. Exponential model	Gamma	4.409E-03
2. Weibull model	Gamma	1.548E+00
	Theta	3.069E+02
3. Gamma model	Gamma	7.181E-01
	Theta	3.276E+02

Repair Time Distribution

Table 3.29 Values of distribution models for repair time

Distribution model	Parameter	Parameter value
1. Exponential model	Gamma	2.583E-01
2. Weibull model	Gamma	8.324E-01
	Theta	3.623E+00
3. Gamma model	Gamma	4.579E-01
	Theta	8.720E+00

The results of the MC simulation are depicted in Fig. 3.85. The representative sample data come from a population with a *gamma distribution*, as illustrated. The median (MTTF) of the representative data is given as approximately 2.3, which does not differ greatly from the MTTF for the three-parameter Weibull distribution for ungrouped data, which equals 2.35 (days). This Weibull distribution has a shape parameter, BETA, of 1.63, which is greater than 1, indicating a wear-out condition in the plant's life cycle.

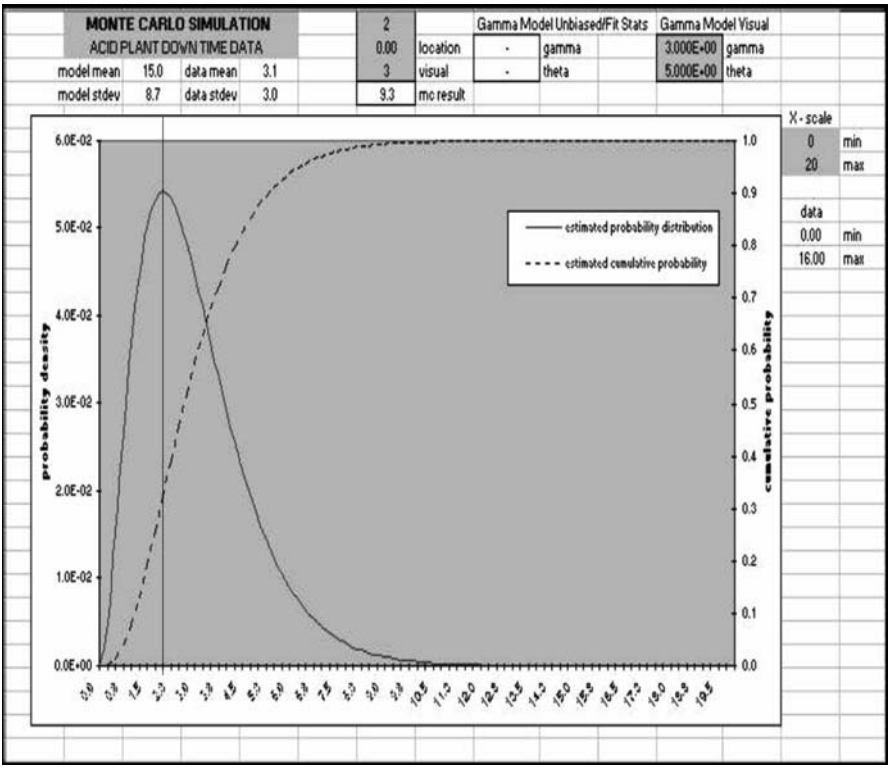


Fig. 3.85 Monte Carlo simulation spreadsheet results for a gamma distribution best fit of TBF data

Conclusion From the case study data, the assumption can be made that the critical systems' specific high-ranking critical components are inadequately designed from a design integrity point of view, as they indicate wear-out too early in the plant's life cycle. This is with reference to the items listed in Table 3.25, particularly the drying tower blowers' shafts, bearings (PLF) and scroll housings (TLF), the hot gas feed induced draft fan (PFC), the reverse jet scrubber's acid spray nozzles (TLF), the final absorption tower vessel and cooling fan guide vanes (TLF), and the IPAT SO3 cooler's cooling fan control vanes (TLF).

Figure 3.85 shows a typical Monte Carlo simulation spreadsheet of the critical systems' time before failure and MC results for a gamma distribution best fit of TBF data.

3.5 Review Exercises and References

Review Exercises

1. Discuss total cost models for design reliability with regard to risk cost estimation and project cost estimation.
2. Give a brief account of interference theory and reliability modelling.
3. Discuss system reliability modelling based on system performance.
4. Compare functional failure and functional performance.
5. Consider the significance of functional failure and reliability.
6. Describe the benefits of a system breakdown structure (SBS).
7. Give reasons for the application and benefit of Markov modelling (continuous-time and discrete states) in designing for reliability.
8. Discuss the binomial method with regard to series networks and parallel networks.
9. Give a brief account of the principal steps in failure modes and effects analysis (FMEA).
10. Discuss the different types of FMEA and their associated benefits.
11. Discuss the advantages and disadvantages of FMEA.
12. Compare the significant differences between failure modes and effects analysis (FMEA) and failure modes and effects criticality analysis (FMECA).
13. Compare the advantages and disadvantages of the RPN technique with those of the military standard technique.
14. Discuss the relevance of FMECA data sources and users.
15. Consider the significance of fault-tree analysis (FTA) in reliability, safety and risk assessment.
16. Describe the fundamental fault-tree analysis steps.
17. Explain the basic properties of the hazard rate function and give a brief description of the main elements of the hazard rate curve.
18. Discuss component reliability and failure distributions.

19. Define the application of the exponential failure distribution in reliability analysis and discuss the distribution's statistical properties.
20. Define the application of the Weibull failure distribution in reliability analysis and discuss the distribution's statistical properties.
21. Explain the Weibull shape parameter and its use.
22. Discuss the significance of the Weibull distribution function in hazards analysis.
23. Describe the principal properties and use of the Weibull graph chart.
24. Consider the application of reliability evaluation of two-state device networks.
25. Describe the fundamental differences between two-state device series networks, parallel networks, and *k*-out-of-*m* unit networks.
26. Consider the application of reliability evaluation of three-state device networks.
27. Briefly describe three-state device parallel networks and three-state device series networks.
28. Discuss system performance measures in designing for reliability.
29. Consider pertinent approaches to determination of the most reliable design in conceptual design.
30. Discuss conceptual design optimisation.
31. Describe the basic comparisons of conceptual designs.
32. Define labelled interval calculus (LIC) with regard to constraint labels, set labels, and labelled interval inferences.
33. Consider the application of labelled interval calculus in designing for reliability.
34. Give a brief description with supporting examples of the methods for:
    a. Determination of a data point: two sets of limit intervals.
    b. Determination of a data point: one upper limit interval.
    c. Determination of a data point: one lower limit interval.
    d. Analysis of the interval matrix.
35. Give reasons for the application of FMEA and FMECA in engineering design analysis.
36. Define reliability-critical items.
37. Describe algorithmic modelling in failure modes and effects analysis with regard to numerical analysis, order of magnitude, qualitative simulation, and fuzzy techniques.
38. Discuss qualitative reasoning in failure modes and effects analysis.
39. Give a brief account of the concept of uncertainty in engineering design analysis.
40. Discuss uncertainty and incompleteness in knowledge.
41. Give a brief overview of fuzziness in engineering design analysis.
42. Describe fuzzy logic and fuzzy reasoning in engineering design.
43. Define the theory of approximate reasoning.
44. Consider uncertainty and incompleteness in design analysis.
45. Give a brief account of modelling uncertainty in FMEA and FMECA.
46. In the development of the qualitative FMECA, describe the concepts of logical expression and expression of uncertainty in FMECA.
47. Give an example of uncertainty in the extended FMECA.
48. Describe the typical results expected of a qualitative FMECA.

49. Define the proportional hazards model with regard to non-parametric model formulation and parametric model formulation.
50. Define the maximum likelihood estimation parameter.
51. Briefly describe the characteristics of the one-parameter exponential distribution.
52. Explain the process of estimating the parameter of the exponential distribution.
53. Consider the approach to determining the maximum likelihood estimation (MLE) parameter.
54. Compare the characteristics of the two-parameter Weibull distribution with those of the three-parameter Weibull model.
55. Give a brief account of the procedures to calculate the Weibull parameters β, μ and γ.
56. Describe the procedure to derive the mean time between failures (MTBF) μ from the Weibull distribution model.
57. Describe the procedure to obtain the standard deviation σ from the Weibull distribution model.
58. Give a brief account of the method of qualitative analysis of the Weibull distribution model.
59. Consider expert judgment as data.
60. Discuss uncertainty, probability theory and fuzzy logic in designing for reliability.
61. Describe the application of fuzzy logic in reliability evaluation.
62. Describe the application of fuzzy judgment in reliability evaluation.
63. Give a brief account of elicitation and analysis of expert judgment in designing for reliability.
64. Explain initial reliability calculation using Monte Carlo simulation.
65. Give an example of fuzzy judgment in reliability evaluation.

References

Abernethy RB (1992) New methods for Weibull and log normal analysis. ASME Pap no 92-WA/DE-14, ASME, New York

Agarwala AS (1990) Shortcomings in MIL-STD-1629A: guidelines for criticality analysis. In: Reliability Maintainability Symp, pp 494–496

AMCP 706-196 (1976) Engineering design handbook: development guide for reliability. Part II. Design for reliability. Army Material Command, Dept of the Army, Washington, DC

Andrews JD, Moss TR (1993) Reliability and risk assessment. American Society of Mechanical Engineers

Artale A, Franconi E (1998) A temporal description logic for reasoning about actions and plans. J Artificial Intelligence Res JAIR, pp 463–506

Ascher W (1978) Forecasting: an appraisal for policymakers and planners. John Hopkins University Press, Baltimore, MD

Aslaksen E, Belcher R (1992) Systems engineering. Prentice Hall of Australia

Barnett V (1973) Comparative statistical inference. Wiley, New York

Barringer PH (1993) Reliability engineering principles. Barringer, Humble, TX

Barringer PH (1994) Management overview: reliability engineering principles. Barringer, Humble, TX

Barringer PH, Weber DP (1995) Data for making reliability improvements. Hydrocarbons Processing Magazine, 4th Int Reliability Conf, Houston, TX

Batill SM, Renaud JE, Xiaoyu Gu (2000) Modeling and simulation uncertainty in multidisciplinary design optimization. In: 8th AIAA/NASA/USAF/ISSMO Symp Multidisciplinary Analysis and Optimisation, AIAA, Long Beach, CA, AIAA-200-4803, pp 5–8

Bement TR, Booker JM, Sellers KF, Singpurwalla ND (2000a) Membership functions and probability measures of fuzzy sets. Los Alamos Nat Lab Rep LA-UR-00-3660

Bement TR, Booker JM, Keller-McNulty S, Singpurwalla ND (2000b) Testing the untestable: reliability in the 21st century. Los Alamos Nat Lab Rep LA-UR-00-1766

Bennett BM, Hoffman DD, Murthy P (1992) Lebesgue order on probabilities and some applications to perception. J Math Psychol

Bezdek JC (1993) Fuzzy models—what are they and why? IEEE Transactions Fuzzy Systems vol 1, no 1

Blanchard BS, Fabrycky WJ (1990) Systems engineering and analysis. Prentice Hall, Englewood Cliffs, NJ

Boettner DD, Ward AC (1992) Design compilers and the labeled interval calculus. In: Tong C, Sriram D (eds) Design representation and models of routine design. Artificial Intelligence in Engineering Design vol 1. Academic Press, San Diego, CA, pp 135–192

Booker JM, Meyer MA (1988) Sources and effects of inter-expert correlation: an empirical study. IEEE Trans Systems Man Cybernetics 8(1):135–142

Booker JM, Smith RE, Bement TR, Parkinson WJ, Meyer MA (1999) Example of using fuzzy control system methods in statistics. Los Alamos Natl Lab Rep LA-UR-99-1712

Booker JM, Bement TR, Meyer MA, Kerscher WJ (2000) PREDICT: a new approach to product development and lifetime assessment using information integration technology. Los Alamos Natl Lab Rep LA-UR-00-4737

Bowles JB, Bonnell RD (1994) Failure mode effects and criticality analysis. In: Annual Reliability and Maintainability Symp, pp 1–34

Brännback M (1997) Strategic thinking and active decision support systems. J Decision Systems 6:9–22

BS5760 (1991) Guide to failure modes, effects and criticality analysis (FMEA and FMECA). British Standard BS5760 Part 5

Buchanan BG, Shortliffe EH (1984) Rule-based expert systems. Addison-Wesley, Reading, MA

Buckley J, Siler W (1987) Fuzzy operators for possibility interval sets. Fuzzy Sets Systems 22:215–227

Bull DR, Burrows CR, Crowther WJ, Edge KA, Atkinson RM, Hawkins PG, Woollons DJ (1995a) Failure modes and effects analysis. Engineering and Physical Sciences Research Council GR/J58251 and GR/J88155

Bull DR, Burrows CR, Crowther WJ, Edge KA, Atkinson RM, Hawkins PG, Woollons DJ (1995b) Approaches to automated FMEA of hydraulic systems. In: Proc ImechE Congr Aerotech 95 Seminar, Birmingham, Pap C505/9/099

Carlsson C, Walden P (1995a) Active DSS and hyperknowledge: creating strategic visions. In: Proc EUFIT'95 Conf, Aachen, Germany, August, pp 1216–1222

Carlsson C, Walden P (1995b) On fuzzy hyperknowledge support systems. In: Proc 2nd Int Worksh Next Generation Information Technologies and Systems, Naharia, Israel, June, pp 106–115

Carlsson C, Walden P (1995c) Re-engineering strategic management with a hyperknowledge support system. In: Christiansen JK, Mouritsen J, Neergaard P, Jepsen BH (eds) Proc 13th Nordic Conf Business Studies, Denmark, vol II, pp 423–437

Carter ADS (1986) Mechanical reliability. Macmillan Press, London

Carter ADS (1997) Mechanical reliability and design. Macmillan Press, London

Cayrac D, Dubois D, Haziza M, Prade H (1994) Possibility theory in fault mode effects analyses—a satellite fault diagnosis application. In: Proc 3rd IEEE Int Conf Fuzzy Systems FUZZ-IEEE '94, Orlando, FL, June, pp 1176–1181

Cayrac D, Dubois D, Prade H (1995) Practical model-based diagnosis with qualitative possibilistic uncertainty. In: Besnard P, Hanks S (eds) Proc 11th Conf Uncertainty in Artificial Intelligence, pp 68–76

Cayrol M, Farency H, Prade H (1982) Fuzzy pattern matching. Kybernetes, pp 103–106

Chiueh T (1992) Optimization of fuzzy logic inference architecture. Computer, May, pp 67–71

Coghill GM, Chantler MJ (1999a) Constructive and non-constructive asynchronous qualitative simulation. In: Proc Int Worksh Qualitative Reasoning, Scotland

Coghill GM, Shen Q, Chantler MJ, Leitch RR (1999b) Towards the use of multiple models for diagnoses of dynamic systems. In: Proc Int Worksh Principles of Diagnosis, Scotland

Conlon JC, Lilius WA (1982) Test and evaluation of system reliability, availability and maintainability. Office of the Under Secretary of Defense for Research and Engineering, DoD 3235.1-H

Cox DR (1972) Regression models and life tables (with discussion). J R Stat Soc B 34:187–220

Davis E (1987) Constraint propagation with interval labels. Artificial Intelligence 32:281–331

de Kleer J, Brown JS (1984) A qualitative physics based on confluences. Artificial Intelligence 24:7–83

Dhillon BS (1983) Reliability engineering in systems design and operation. Van Nostrand Reinhold, Berkshire

Dhillon BS (1999a) Design reliability: fundamentals and applications. CRC Press, LLC 2000, NW Florida

Dubois D, Prade H (1988) Possibility theory—an approach to computerized processing of uncertainty. Plenum Press, New York

Dubois D, Prade H (1990) Modelling uncertain and vague knowledge in possibility and evidence theories. Uncertainty in Artificial Intelligence vol 4. Elsevier, Amsterdam, pp 303–318

Dubois D, Prade H (1992a) Upper and lower images of a fuzzy set induced by a fuzzy relation: applications to fuzzy inference and diagnosis. Information Sci 64:203–232

Dubois D, Prade H (1992b) Fuzzy rules in knowledge-based systems modeling gradedness, uncertainty and preference. In: Zadeh LA (ed) An introduction to fuzzy logic applications in intelligent systems. Kluwer, Dordrecht, pp 45–68

Dubois D, Prade H (1992c) Gradual inference rules in approximate reasoning. Information Sci 61:103–122

Dubois D, Prade H (1992d) When upper probabilities are possibility measures. Fuzzy Sets Systems 49:65–74

Dubois D, Prade H (1993a) Fuzzy sets and probability: misunderstandings, bridges and gaps. Report (translated), Institut de Recherche en Informatique de Toulouse (I.R.I.T.) Université Paul Sabatier, Toulouse

Dubois D, Prade H (1993b) A fuzzy relation-based extension of Reggia's relational model for diagnosis. In: Heckerman, Mamdani (eds) Proc 9th Conf Uncertainty in Artificial Intelligence, WA, pp 106–113

Dubois D, Prade H, Yager RR (1993) Readings in fuzzy sets and intelligent systems. Morgan Kaufmann, San Mateo, CA

Dubois D, Lang J, Prade H (1994) Automated reasoning using possibilistic logic: semantics, belief revision and variable certainty weights. IEEE Trans Knowledge Data Eng 6:64–69

EPRI (1974) A review of equipment aging theory and technology. Nuclear Safety & Analysis Department, Nuclear Power Division, Electricity Power Research Institute, Palo Alto, CA

Fishburn P (1986) The axioms of subjective probability. Stat Sci 1(3):335–358

Fullér R (1999) On fuzzy reasoning schemes. In: Carlsson C (ed) The State of the Art of Information Systems in 2007. Turku Centre for Computer Science, Abo, TUCS Gen Publ no 16, pp 85–112

Grant Ireson W, Coombs CF, Moss RY (1996) Handbook of reliability engineering and management. McGraw-Hill, New York

ICS (2000) The RAMS plant analysis model. ICS Industrial Consulting Services, Gold Coast City, Queensland

IEEE Std 323-1974 (1974) IEEE Standard for Qualifying Class IE Equipment for Nuclear Power Generating Stations. Institute of Electrical and Electronics Engineers, New York

Kerscher W, Booker J, Bement T, Meyer M (1998) Characterizing reliability in a product/process design-assurance program. In: Proc Int Symp Product Quality and Integrity, Anaheim, CA, and Los Alamos Lab Rep LA-UR-97-36

Klir GJ, Yuan B (1995) Fuzzy sets and fuzzy logic theory and application. Prentice Hall, Englewood Cliffs, NJ

Kuipers B (1990) Qualitative simulation. Artificial Intelligence 29(3):289–338 (1986), reprinted in Qualitative reasoning about physical systems, Morgan Kaufman, San Mateo, CA, pp 236–260

Laviolette M, Seaman J Jr, Barrett J, Woodall W (1995) A probabilistic and statistical view of fuzzy methods. Technometrics 37:249–281

Lee RCT (1972) Fuzzy logic and the resolution principle. J Assoc Computing Machinery 19:109–119

Liu JS, Thompson G (1996) The multi-factor design evaluation of antenna structures by parameter profile analysis. Proc Inst Mech Engrs Part B, J Eng Manufacture 210:449–456

Loginov VI (1966) Probability treatment of Zadeh membership functions and their use in pattern recognition. Eng Cybernetics 68–69

Martz HF, Almond RG (1997) Using higher-level failure data in fault tree quantification. Reliability Eng System Safety 56(1):29–42

Mavrovouniotis M, Stephanopoulos G (1988) Formal order of magnitude reasoning in process engineering. Computers Chem Eng 12:867–881

Meyer MA, Booker JM (1991) Eliciting and analyzing expert judgment: a practical guide. Academic Press, London

Meyer MA, Butterfield KB, Murray WS, Smith RE, Booker JM (2000) Guidelines for eliciting expert judgement as probabilities or fuzzy logic. Los Alamos Natl Lab Rep LA-UR-00-218

MIL-STD-721B (1980) Definition of terms for reliability and maintainability. Department of Defense (DoD), Washington, DC

MIL-STD-1629 (1980) Procedures for performing a failure mode, effects, and criticality analysis. DoD, Washington, DC

Moore R (1979) Methods and applications of interval analysis. SIAM, Philadelphia, PA

Moss TR, Andrews JD (1996) Reliability assessment of mechanical systems. Proc Inst Mech Engrs vol 210

Natvig B (1983) Possibility versus probability. Fuzzy Sets Systems 10:31–36

Norwich AM, Turksen IB (1983) A model for the measurement of membership and the consequences of its empirical implementation. Fuzzy Sets Systems 12:1–25

Orchard RA (1998) FuzzyCLIPS Version 6.04A. Integrated Reasoning, Institute for Information Technology, National Research Council Canada

Ortiz NR, Wheeler TA, Breeding RJ, Hora S, Meyer MA, Keeney RL (1991) The use of expert judgment in NUREG-1150. Nuclear Eng Design 126:313–331 (revised from Sandia Natl Lab Rep SAND88-2253C, and Nuclear Regulatory Commission Rep NUREG/CP-0097 5, pp 1–25

Pahl G, Beitz W (1996) Engineering design. Springer, Berlin Heidelberg New York

Payne S (1951) The art of asking questions. Princeton University Press, Princeton, NJ

Raiman O (1986) Order of magnitude reasoning. In: Proc 5th National Conf Artificial Intelligence AAAI-86, pp 100–104

ReliaSoft Corporation (1997) Life data analysis reference. ReliaSoft Publ, Tucson, AZ

Roberts FS (1979) Measurement theory. Addison-Wesley, Reading, MA

Ryan M, Power J (1994) Using fuzzy logic—towards intelligent systems. Prentice-Hall, Englewood Cliffs, NJ

Shen Q, Leitch R (1993) Fuzzy qualitative simulation. IEEE Trans Systems Man Cybernetics 23(4), and J Math Anal Appl 64(2):369–380 (1993)

Shortliffe EH (1976) Computer-based medical consultation: MYCIN. Elsevier, New York

Simon HA (1981) The sciences of the artificial. MIT Press, Cambridge, MA

Smith RE, Booker JM, Bement TR, Meyer MA, Parkinson WJ, Jamshidi M (1998) The use of fuzzy control system methods for characterizing expert judgment uncertainty distributions. In: Proc PSAM 4 Int Conf, September, pp 497–502

Sosnowski ZA (1990) FLISP—a language for processing fuzzy data. Fuzzy Sets Systems 37:23–32

Steele AD, Leitch RR (1996) A strategy for qualitative model-based diagnosis. In: Proc IFAC-96 13th World Congr, San Francisco, CA, vol N, pp 109–114
Steele AD, Leitch RR (1997) Qualitative parameter identification. In: Proc QR-97 11th Int Worksh Qualitative Reasoning About Physical Systems, pp 181–192
Thompson G, Geominne J, Williams JR (1998) A method of plant design evaluation featuring maintainability and reliability. Proc Inst Mech Engrs vol 212 Part E
Thompson G, Liu JS, Hollaway L (1999) An approach to design for reliability. Proc Inst Mech Engrs vol 213 Part E
Walden P, Carlsson C (1995) Hyperknowledge and expert systems: a case study of knowledge formation processes. In: Nunamaker JF (ed) Information systems: decision support systems and knowledge-based systems. Proc 28th Annu Hawaii Int Conf System Sciences, IEEE Computer Society Press, Los Alamitos, CA, vol III, pp 73–82
Whalen T, Schott B (1983) Issues in fuzzy production systems. Int J Man-Machine Studies 19:57
Whalen T, Schott B, Ganoe F (1982) Fault diagnosis in fuzzy network. Proc 1982 Int Conf Cybernetics and Society, IEEE Press, New York
Wirth R, Berthold B, Krämer A, Peter G (1996) Knowledge-based support of system analysis for failure mode and effects analysis. Eng Appl Artificial Intelligence 9(3):219–229
Wolfram J (1993) Safety and risk: models and reality. Proc Inst Mech Engrs vol 207
Yen J, Langari R, Zadeh LA (1995) Industrial applications of fuzzy logic and intelligent systems. IEEE Press, New York
Zadeh LA (1965) Fuzzy sets. Information Control 8:338–353
Zadeh LA (1968) Probability measures of fuzzy events. J Math Anal Appl 23:421–427
Zadeh LA (1973) Outline of a new approach to the analysis of complex systems and decision processes. IEEE Trans Systems Man Cybernetics 2:28–44
Zadeh LA (1975) The concept of a linguistic variable and its application to approximate reasoning I–III. Elsevier, New York, Information Sci 8:199–249, 9:43–80
Zadeh LA (1978) Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets Systems 1:3–28
Zadeh LA (1979) A theory of approximate reasoning. In: Hayes J, Michie D, Mikulich LI (eds) Machine Intelligence, vol 9. Wiley, New York, pp 149–194

Chapter 4
Availability and Maintainability in Engineering Design

Abstract Evaluation of operational engineering availability and maintainability is usually considered in the detail design phase, or *after* installation of an engineering design. It deals with the *prediction* and *assessment* of the design's availability, or the probability that a system will be in operational service during a scheduled operating period, as well as the design's maintainability, or the probability of system restoration within a specified downtime. This chapter considers in detail the concepts of availability and maintainability in engineering design, as well as the various criteria essential to designing for availability and designing for maintainability. Availability in engineering design has its roots in designing for reliability. If the design includes a durability feature related to its availability and reliability, then it fulfils, to a large extent, the requirements for engineering design integrity. Availability in engineering design is thus considered from the perspective of the design's functional and operational characteristics, and designing for availability, particularly engineering process availability, considers measurements of process throughput, output, input and capacity. Designing for availability is a 'top-down' approach from the design's systems level to its equipment or assemblies level whereby constraints on the design's functional and operational performance are determined. Maintainability in engineering design is the relative ease and economy of time and resources with which an engineered installation can be retained in, or restored to, a specified condition through scheduled and unscheduled maintenance. In this context, maintainability is a function of engineering design. Therefore, designing for maintainability requires that the installation is serviceable and can be easily repaired, and also supportable in that it can be cost-effectively and practically kept in or restored to a usable condition. Maintainability is fundamentally a design parameter, and designing for maintainability defines the time an installation could be inoperable.

R.F. Stapelberg, *Handbook of Reliability, Availability, Maintainability and Safety in Engineering Design*, © Springer 2009

4.1 Introduction

The foregoing chapter dealt with the analysis of engineering design with respect to the *prediction*, *assessment* and *evaluation* of reliability and systems functional performance, without considering *repair* in the event of failure. This chapter deals with repairable systems and their equipment in engineering design, which can be restored to operational service after failure. It covers the *prediction* and *assessment* of availability (the probability that a system will be in operational service during a scheduled operating period), and maintainability (the probability of system restoration within a specified downtime). Evaluation of operational availability and maintainability is normally considered in the detail design phase, or *after* installation of the engineering design, such as during the design's operational use or during process ramp-up and production in process engineering installations.

Availability in engineering design has its roots in *designing for reliability* as well as *designing for maintainability*, in which a 'top-down' approach is adopted, predominantly from the design's systems level to its equipment level (i.e. assembly level), and constraints on systems *operational* performance are determined. Availability in engineering design was initially developed in defence and aerospace design (Conlon et al. 1982), whereby availability was viewed as a measure of the degree to which a system was in an operable state at the beginning of a mission, whenever called for at any random point in time.

Traditional reliability engineering considered *availability* simply as a special case of *reliability* while taking the *maintainability* of equipment into account. Availability was regarded as the parameter that translated system reliability and maintainability characteristics into an index of system *effectiveness*. Availability in engineering design is fundamentally based on the question 'what must be considered to ensure that the equipment will be in a working condition when needed for a specific period of time?'.

The ability to answer this question for a particular system and its equipment represents a powerful concept in engineering design integrity, with resulting additional side-benefits. One important benefit is the ability to use availability analysis during the engineering design process as a platform to support design for reliability and design for maintainability parameters, as well as trade-offs between these parameters.

Availability is intrinsically defined as "*the probability that a system is operating satisfactorily at any point in time when used under stated conditions, where the time considered includes the operating time and the active repair time*" (Nelson et al. 1981).

While this definition is conceptually rather narrow, especially concerning the *repair* time, the thrust of the approach of *availability in engineering design* is to initially consider *inherent availability* in contrast to *achieved* and *operational availability* of processes and systems. A more comprehensive approach would need to include a measure for the quantification of *uncertainty*, which involves considering the concept of availability as a decision analysis problem. This results in identifying different options for improving availability by evaluating respective outcomes with specific criteria such as *costs* and *benefits*, and quantifying their likelihood of

occurrence. Economic incentive is the primary basis for the growing interest in more deliberate and systematic availability analysis in engineering design.

Ensuring a proper analysis in the determination of *availability in engineering design* is one of the few alternatives that design engineers may have for obtaining an increase in process and/or systems capacity, without incurring significant increases in capital costs. From the definition, it is evident that any form of availability analysis is time-related.

Figure 4.1 illustrates the breakdown of a total system's equipment time into time-based elements on which the analysis of availability is based. It must be noted that the time designated as 'off time' does not apply to availability analysis because, during this time, system operation is not required. It has been included in the illustration, however, as this situation is often found in complex integrated systems, where the reliability concept of 'redundancy' is related to the availability concept of 'standby'.

The basic relationship model for availability is (Eq. 4.1):

$$\text{Availability} = \frac{\text{Up Time}}{\text{Total Time}} = \frac{\text{Up Time}}{\text{Up Time} + \text{Down Time}} \tag{4.1}$$

Analysis of availability is accomplished by substituting the time-based elements defined above into various forms of the basic relationship, where different combinations formulate various definitions of availability.

Designing for availability predominantly considers whether a design has been configured at *systems* level to meet certain *availability* requirements based on specific process or systems *operating* criteria. Designing for availability is mainly considered at the design's systems and higher equipment level (i.e. assembly level, and *not* component level), whereby availability requirements based on expected systems performance are determined, which eventually affects all of the items in the systems hierarchy. Similar to *designing for reliability*, this approach does not depend on having to initially identify all the design's components, and is suitable for the conceptual or preliminary design stage (Huzdovich 1981).

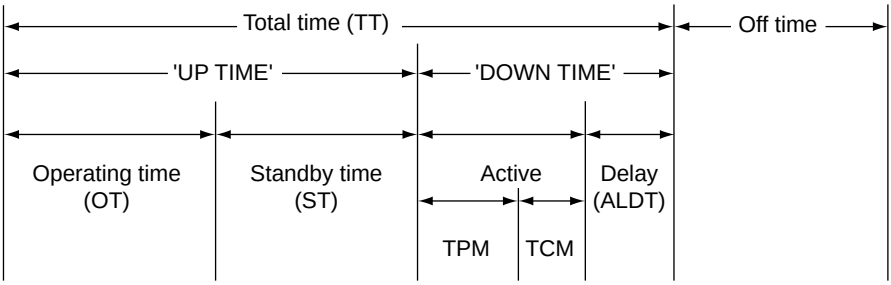


Fig. 4.1 Breakdown of total system's equipment time (DoD 3235.1-H 1982) where UP TIME = operable time, DOWN TIME = inoperable time, OT = operating time, ST = standby time, ALDT = administrative and logistics downtime, TPM = total preventive maintenance and TCM = total corrective maintenance

However, it is observed practice in most large continuous process industries that have complex integrations of systems, particularly the power-generating industry and the chemical process industries, that the concept of availability is closely related to reliability, whereby many 'availability' measures are calculated as a 'bottom-up' *evaluation*. In such cases, *availability in engineering design* is approached from the design's lower levels (i.e. assembly and/or component levels) *up* the systems hierarchy to the design's higher levels (i.e. system and process levels), whereby the collective effect of all the equipment availabilities is determined. Clearly, this approach is feasible only once all the design's equipment have been identified, which is well into the detail design stage.

In order to establish the most applicable methodology for determining the integrity of engineering design at different stages of the design process, particularly with regard to the development of *designing for availability*, or to the assessment of *availability in engineering design* (i.e. 'top-down' or 'bottom-up' approaches in the systems hierarchy respectively), some of the basic availability analysis techniques applicable to either of these approaches need to be identified by definition and considered for suitability in achieving the goal of this research.

Furthermore, it must also be noted that these techniques do *not* represent the total spectrum of availability analysis, and selection has been based on their application in conjunction with the selected reliability techniques, (reliability prediction, assessment and evaluation), in order to determine the integrity of engineering design at the relative design phases.

The definitions of availability are qualitative in distinction, and indicate significant differences in approaches to the determination of designing for availability at different levels of the systems hierarchy, such as:

- *prediction of inherent availability* of systems based on a *prognosis of systems operability* and *systems performance* under conditions subject to various *performance criteria*;
- *assessment of achieved availability* based on *inferences of equipment usage with respect to downtime and maintenance*;
- *evaluation of operational availability* based on *measures of time that are subject to delays*, particularly with respect to anticipated values of administrative and logistics downtime.

Maintainability in engineering design is described in the USA military handbook 'Designing and developing maintainable products and systems' (MIL-HDBK-470A 1997) as "*the relative ease and economy of time and resources with which an item can be retained in, or restored to, a specified condition when maintenance is performed by personnel having specified skill levels, using prescribed procedures and resources, at each prescribed level of maintenance and repair. In this context, it is a function of design*".

Maintainability refers to the measures taken during the design, development and manufacture of an engineered installation that reduce the required maintenance, repair skill levels, logistic costs and support facilities, to ensure that the installation meets the requirements for its intended use. A key consideration in the maintain-

ability measurement of a system is its active *downtime*, i.e. the time required to bring a failed system back to its operational state or capability. This active downtime is normally attributed to *maintenance* activities.

An effective way to increase a system's *availability* is to improve its maintainability by minimising the downtime. This minimised downtime does not happen at random; it is *designed* to happen by actively ensuring that proper and progressive consideration be given to *maintainability* requirements during the conceptual, schematic and detail design phases. Therefore, the inherent maintainability characteristics of the system and its equipment must be assured. This can be achieved only by the implementation of specific design practices, and verified and validated through maintainability assessment and evaluation methods respectively, utilising both analyses and testing.

The following topics cover some of these assurance activities:

- Maintainability analysis
- Maintainability modelling
- Designing for maintainability.

Maintainability analysis includes the *prediction* as well as the *assessment* and *evaluation* of maintainability criteria throughout the engineering design process, and would normally be implemented by a well-defined program, and captured in a maintainability program plan (MPP).

Maintainability analysis differs significantly from one design phase to the next, particularly with respect to a *systems-level* approach during the early conceptual and schematic design phases, in contrast to an *equipment-level* approach during the later schematic and detail design phases. These differences in approach have a significant impact on *maintainability in engineering design* as well as on contractor/manufacturer responsibilities. Maintainability is a design consideration, whereas *maintenance* is a consequence of that design. However, at the early stages of engineering design, it is important to identify the maintenance concept, and derive the initial system maintainability requirements and related design attributes. This constitutes maintainability analysis.

Maintainability, from a maintenance perspective, can be defined as "*the probability that a failed item will be restored to an operational effective condition within a given period of time*".

This restoration of a failed item to an operational effective condition is normally when *repair action*, or *corrective action* in *maintenance* is performed in accordance with prescribed standard procedures. The item's operational effective condition in this context is also considered to be the item's *repairable condition*. Maintainability is thus the *probability* that an item will be restored to a *repairable condition* through *corrective maintenance action*, in accordance with prescribed standard procedures, within a given period of time.

Corrective maintenance action is the *action* to rectify or set right defects in the equipment's *operational* and *physical conditions*, on which its functions depend, in accordance with a standard. Similarly, it can also be discerned, from the description of *corrective maintenance action* in maintenance, that *maintainability* is achieved

through restorative *corrective maintenance action* through some or other *repair action*. This *repair action* is, in fact, action to rectify or set right defects in accordance with a standard.

The *repairable condition* of equipment is determined by the *mean time to repair (MTTR)*, which is a measure of its *maintainability*.

Maintainability is thus a measure of the repairable condition of an item that is determined by MTTR, and is established through corrective maintenance action.

Maintainability modelling for a repairable system is, to a certain extent, a form of applied probability analysis, very similar to the probability assessment of uncertainty in reliability. It includes Bayesian methods applied to Poisson processes, as well as Weibull analysis and Monte Carlo simulation, which is used extensively in availability analysis. Maintainability modelling also relates to *queuing theory*. It can be compared to the problem of determining the occupancy, arrival and service rates in a queue, where the service performed is repair, the server is the maintenance function, and the patrons of the queue are the systems and equipment that are repaired at random intervals, coincidental to the random occurrences of failures.

Applying maintainability models enhances the capability of designing for maintainability through the appropriate consideration of design criteria such as *visibility*, *accessibility*, *testability* and *interchangeability*. Using maintainability prediction techniques, as well as specific quantitative maintainability analysis models relating to the operational requirements of a design can greatly enhance not only the integrity of engineering design but also the confidence in the operational capabilities of a design. Maintainability predictions of the operational requirements of a design during its conceptual design phase can aid in design decisions where several design options need to be considered. Quantitative maintainability analysis during the schematic and detail design phases consider the assessment and evaluation of maintainability from the point of view of *maintenance* and *logistics support* concepts.

Designing for maintainability requires a product that is *serviceable* (must be easily repaired) and *supportable* (must be cost-effectively kept in, or restored to, a usable condition). If the design includes a *durability* feature related to *availability* (degree of operability) and *reliability* (absence of failures), then it fulfils, to a large extent, the requirements for engineering design integrity. Maintainability is primarily a design parameter, and designing for maintainability defines how long the equipment is expected to be down. *Serviceability* implies the speed and ease of maintenance, whereby the amount of time expected to be spent by an appropriately trained maintenance function working within a responsive supply system is such that it will achieve minimum downtime in restoring failed equipment. In designing for maintainability, the *type of maintenance* must be considered, and must have an influential role in considering *serviceability*.

For example, the stipulation that a system should be capable of being isolated to the component level of each circuit card in its control sub-system may not be justified if a faulty circuit card is to be replaced, rather than repaired. Such a design would impose added developmental cost in having to accommodate a redundant feature in its functional control.

Supportability has a design subset involving *testability*, a design characteristic that allows verification of the operational status to be determined and faults within the system's equipment to be isolated in a timely and effective manner. This is achieved through the use of built-in-test equipment, so that an installed item can be monitored with regard to its status (operable, inoperable or degraded).

Designing for maintainability also needs to take cognisance of the item's operational *durability* whereby the period (downtime) in which equipment will be down due to unavailability and/or unreliability needs to be considered. Unavailability in this context occurs when the equipment is down for periodic maintenance and for repairs. Unreliability is associated with system failures where the failures can be associated with unplanned outages (corrective action) or planned outages (preventive action). Relevant criteria in designing for maintainability need to be verified through maintainability *design reviews*. These design reviews are conducted during the various design phases of the engineering design process, and are critical components of modern design practice. The primary objective of maintainability design reviews is to determine the relevant progress of the design effort, with particular regard to designing for maintainability, at the completion of each specific design phase. As with design reviews in general (i.e. design reviews concerned with designing for reliability, availability, maintainability and safety), maintainability design reviews fall into three distinct categories: initial or conceptual design reviews, intermediate or schematic design reviews, and final or detail design reviews (Hill 1970).

Initial or *conceptual design reviews* need to be conducted immediately *after* formulation of the conceptual design, from initial process flow diagrams (PFDs). The purpose is to carefully examine the functionality of the intended design, feasibility of the criteria that must be met, initial formulation of design specifications at process and systems level, identification of process design constraints, existing knowledge of similar systems and/or engineered installations, and cost-effective objectives.

Intermediate or *schematic design reviews* need to be conducted immediately *after* the schematic engineering drawings are developed from firmed-up PFDs and initial pipe and instrument diagrams (P&IDs), and when primary specifications are fixed. This is to compare formulation of design criteria in specification requirements with the proposed design. These requirements involve assessments of systems performance, reliability, inherent and achieved availability, maintainability, hazardous operations (HazOps) and safety, as well as cost estimates.

Final or *detail design reviews*, referred to as the *critical design review* (Carte 1978), are conducted immediately *after* detailed engineering drawings are developed for review (firmed PFDs and firmed P&IDs) and most of the specifications have been fixed. At this stage, results from preceding design reviews, and detail costs data are available. This review considers evaluation of design integrity and due diligence, hazards analyses (HazAns), value engineering, manufacturing methods, design producibility/constructability, quality control and detail costing.

The essential criteria that need to be considered with *maintainability design reviews* at the completion of the various engineering design phases include the following (Patton 1980):

- Design constraints and specified systems interfaces
- Verification of maintainability prediction results
- Evaluation of maintainability trade-off studies
- Evaluation of FMEA results
- Maintainability problem areas and maintenance requirements
- Physical design configuration and layout schematics
- Design for maintainability specifications
- Verification of maintainability quantitative characteristics
- Verification of maintainability physical characteristics
- Verification of design ergonomics
- Verification of design configuration accessibility
- Verification of design equipment interchangeability
- Evaluation of physical design factors
- Evaluation of facilities design dictates
- Evaluation of maintenance design dictates
- Verification of systems testability
- Verification of health status and monitoring (HSM)
- Verification of maintainability tests
- Use of automatic test equipment
- Use of built-in-test (BIT) methods
- Use of onboard monitoring and fault isolation methods
- Use of online repair with redundancy
- Evaluation of maintenance strategies
- Selection of assemblies and parts kits
- Use of unit (assembly) replacement strategies
- Evaluation of logistic support facilities.

4.2 Theoretical Overview of Availability and Maintainability in Engineering Design

For repairable systems, availability is generally considered to be the ratio of the actual operating time, to the scheduled operating time, exclusive of preventive or planned maintenance. Since availability represents the probability of a system being in an operable state when required, it fundamentally has the same connotation, from a quantitative analysis viewpoint, as the reliability of a non-repairable system. The difference, however, is that reliability is a measure of a system's or equipment's functional performance subject to failure, whereas availability is subject to both failure *and* repair (or restoration). Thus, determining the confidence level for availability prediction is more complicated than it is for reliability prediction, as an extra probability distribution is involved. Because of this, closed formulae for determining confidence in the case of a twofold uncertainty are not easily established, even in the simplest case when both failure and repair events are exponential. It is for this reason that the application of Monte Carlo simulation is resorted to in the analysis

of systems availability. Maintainability, on the other hand, is similar to reliability in that both relate the occurrence of a single type of event over time. It is thus necessary to consider in closer detail the various definitions of availability (Conlon et al. 1982).

Inherent availability can be defined as "*the prediction of expected system performance or system operability over a period which includes the predicted system operating time and the predicted corrective maintenance down time*".

Achieved availability can be defined as "*the assessment of system operability or equipment usage in a simulated environment, over a period which includes its predicted operating time and active maintenance down time*".

Operational availability can be defined as "*the evaluation of potential equipment usage in its intended operational environment, over a period which includes its predicted operating time, standby time, and active and delayed maintenance down time*".

These definitions indicate that the *availability* of an item of equipment is concerned either with expected *system performance* over a period of expected *operational time*, or with *equipment usage* over a period of expected *operational time*, and that the expected *utilisation* of the item of equipment is its expected *usage* over an accountable period of *total time inclusive of downtime*. This aspect of *usage over an accountable period* relates the concepts of *availability* to *utilisation* of an item of equipment, where the *accountable period* is a measure of the ratio of the *actual input* to the *standard input* during the *operational time* of successful system performance. The process measure of *operational input* is thus included in the concept of availability. By grouping selected availability techniques into these three different qualitative definitions, it can be readily discerned which techniques, relating to each of the three terms, can be logically applied in the different stages of the design process, either independently or in conjunction with *reliability* and *maintainability* analysis.

As with *reliability prediction*, the techniques for predicting *inherent availability* would be more appropriate during *conceptual or preliminary design*, when alternative systems in their general context are being identified in preliminary block diagrams, such as first-run process flow diagrams (PFDs), and estimates of the probability of successful performance or operation of alternative designs are necessary.

Techniques for the assessment of *achieved availability* would be more appropriate during *schematic design*, when the PFDs are frozen, process functions defined with relevant specifications relating to specific process performance criteria, and process availability assessed according to expected equipment usage over an accountable period of operating time, inclusive of predicted active maintenance downtime.

Techniques for the evaluation of *operational availability* would be more appropriate during *detail design*, when components of equipment detailed in pipe and instrument drawings (P&IDs) are being specified according to equipment design criteria, and equipment reliability, availability and maintainability are evaluated from a determination of the frequencies with which failures occur over a predicted period of operating time, based on known component failure rates, and the frequencies with

which component failures are repaired during active corrective maintenance downtime. This must also take into account preventive maintenance downtime, as well as delayed maintenance downtime.

Maintainability analysis is a further method of determining the integrity of engineering design by considering all the relevant maintainability characteristics of the system and its equipment. This would include an analysis of the following (MIL-STD-470A; MIL-STD-471A):

- Quantitative characteristics
- Physical characteristics.

Quantitative characteristics considered for a system design are its specific maintainability performance characteristics, which include aspects such as *mean time to repair*, *maximum time to repair*, *built-in-test* and *health status and monitoring*:

- *Mean time to repair (MTTR)*:
  This is calculated by considering the times needed to implement the corrective maintenance and preventive maintenance tasks for each level of maintenance appropriate to the respective systems hierarchical levels.
- *Maximum time to repair*:
  This is an important part of the quantitative characteristics of maintainability performance, in that it gives an indication of the 'worst-case' scenario.
- *Built-in-test (BIT)*:
  The establishment of a BIT capability is important. For example, the principal means of fault detection and isolation at the component level requires the use of self-diagnostics or built-in-testing. This capability, in terms of its effectiveness, may need to be quantified.
- *Health status and monitoring (HSM)*:
  Incorporated into the design of the system could be a HSM capability. This could be a relatively simple concept, such as monitoring the temperature of the shaft of a turbine to safeguard against the main bearings overheating. Other HSM systems may employ a multitude of sensors, such as strain gauges, thermal sensors, accelerometers, etc., to measure electrical and mechanical stresses on a particular component of the assembly or system.

Physical characteristics take into consideration issues and characteristics that will accommodate ease of maintenance, such as *ergonomics* and *visibility*, *testability*, *accessibility* and *interchangeability*:

- *Ergonomics*:
  Ergonomics addresses the physical characteristics of concern to the maintenance function. This could range from the weight of components and required lifting points to the clearance between electrical connectors, to the overall design configuration of assemblies and components for maximum visibility during inspections and maintenance. Visibility is an element of maintainability design that allows the maintenance function visual access to assemblies and components for ease of maintenance action. Even short-duration tasks can increase downtime if the

component is blocked from view. Designing for visibility greatly reduces maintenance times. Human engineering design criteria, as well as human engineering requirements, are well established for military systems and equipment, as presented in the different military standards for systems, equipment and facilities (MIL-STD-1472D; MIL-STD-46855B).

- *Testability*:
  Testability is a measure of the ability to detect system faults and to isolate these at the lowest replaceable component. The speed with which faults are diagnosed can greatly influence downtime and maintenance costs. As technology advances continue to increase the capability and complexity of systems, the use of automatic diagnostics as a means of fault detection, isolation and recovery (FDIR) substantially reduces the need for highly trained maintenance personnel and can decrease maintenance costs by reducing the need to replace components. FDIR systems include both internal diagnostic systems, referred to as built-in-test (BIT) or built-in-test-equipment (BITE), and external diagnostic systems, referred to as automatic test equipment (ATE), or offline test equipment. The equipment are used as part of a reduced support system, all of which will minimise downtime and cost over the operational life cycle.
- *Test point*:
  Test points must be interfaced with the testability engineering effort. A system may require some manual diagnostic interaction, where specific test points will be required for fault diagnostic and isolation purposes.
- *Test equipment*:
  Test equipment assessment is of how test instrumentation would interface with the process system or equipment.
- *Accessibility*:
  Accessibility is perhaps the most important attribute. With complex integration of systems, the design of a single system must avoid the need to remove another system's equipment to gain access to a failed item. Furthermore, the ability to permit the use of standard hand tools must be observed throughout. Accessibility is the ease with which an item can be accessed during maintenance, and can greatly impact maintenance times if not inherent in the design, especially on systems where in-process maintenance is required. When accessibility is poor, other failures are often caused by isolation/disconnection/removal and installation of other items that might hamper access, causing rework. Accessibility of all replaceable, maintainable items will provide time and energy savings.
- *Interchangeability*:
  Interchangeability refers to the ability and ease with which a component can be replaced with a similar component without excessive time or undue retrofit or recalibration. This flexibility in design reduces the number of maintenance procedures and, consequently, reduces maintenance costs. Interchangeability also allows for system expansion with minimal associated costs, due to the use of standard or common end-items.

Maintainability has true design characteristics. Attempts to improve the inherent maintainability of a product/item after the design is frozen are usually expensive, inefficient and ineffective, as demonstrated so often in engineering installations when the first maintenance effort requires the use of a cutting torch to access the item requiring replacement.

In the application of *maintainability analysis*, there are basically two approaches to predicting the mean time to repair (MTTR). The first is a work study method that analyses each repair task into definable work elements. This requires an extensive databank of average times for a wide range of repair tasks for a particular type of equipment. In the absence of sufficient data of average repair times, the work study method of *comparative estimation* is applied, whereby repair times are simulated from failures of similar types of equipment.

The second approach is empirical and involves rating a number of maintainability factors against a *checklist*. The resulting *maintainability scores* are converted into an estimated MTTR by means of a *nomograph* obtained by regression analysis of many different repair times. This second approach is described in detail in the USA military handbook titled 'Maintainability prediction' (MIL-HDBK-472), of which the referenced Procedure 3 is considered to be appropriate for general engineering application. In this procedure, the predicted repair time for each task is arrived at by considering a checklist of *maintainability criteria*, and by scoring points for each criterion. The score for each criterion increases with the degree of conformity to an expected standard. The criteria in the checklist are grouped under three headings:

- Physical design factors
- Design dictates—facilities
- Design dictates—maintenance skills.

The points scored under each heading are appropriately weighted, and relate to the predicted repair time by means of a regression equation that is presented in the form of a nomograph. The checklist is used for accumulating the scores of all the various repair tasks of a particular item, and is reproduced in part below (MIL-HDBK-472). Scoring will apply to maintainability design concepts for ease of maintenance. This is concerned with design for maintainability criteria such as visual and manipulative actions, which would normally precede maintenance actions. The regression equation to calculate the predicted downtime is of the form:

$$\mathrm{Mct} = \mathrm{antilog}(3.54651 - 0.02512\mathrm{A} - 0.03055\mathrm{B} - 0.01093\mathrm{C})$$

where Mct is corrective maintenance time, and A, B and C are scores of the relevant checklists.

CHECKLIST—MIL 472 PROCEDURE 3

Checklist A—scoring physical design factors:

1. Access (external)
2. Latches and fasteners (external)
3. Latches and fasteners (internal)
4. Access (internal)
5. Packaging
6. Units/parts (failed)
7. Visual displays
8. Fault and operation indicators
9. Test points availability
10. Test points identification
11. Labelling
12. Adjustments
13. Testing in circuit
14. Protective devices
15. Safety personnel.

Checklist B—scoring design dictates—facilities:

1. External test equipment
2. Connectors
3. Jigs and fixtures
4. Visual contact
5. Assistance operations
6. Assistance technical
7. Assistance supervisory.

Checklist C—scoring design dictates—maintenance skills:

1. Arm-leg-back strength
2. Endurance and energy
3. Eye-hand coordination
4. Visual requirements
5. Logic application
6. Memory retention
7. Planning
8. Precision
9. Patience
10. Initiative.

The nomograph given in Fig. 4.2 includes scales against which scores for the physical design factors and the design dictates are marked.

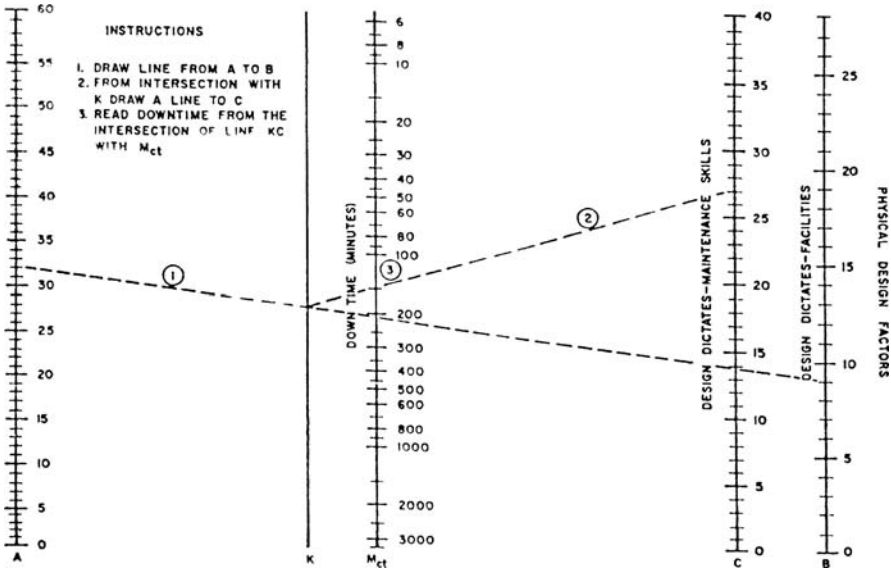


Fig. 4.2 Regression equation of predicted repair time in nomograph form

4.2.1 Theoretical Overview of Availability and Maintainability Prediction in Conceptual Design

Availability and maintainability prediction attempts to quantify the measures of successful *operational performance of systems* under conditions of failure and subject to restoration criteria. Although similar but not identical to reliability prediction, availability and maintainability predictions are predominantly considered in the preliminary design phase, and usually extend through to the schematic design phase of the engineering design process, together with estimations of expected *failure rates* and expected *repair rates*. The most applicable methodology for availability and maintainability prediction in the conceptual design phase includes basic concepts of mathematical modelling such as:

i. Cost modelling for design availability and maintainability
ii. Availability modelling based on system performance
iii. Inherent availability modelling with uncertainty
iv. Preliminary maintainability modelling.

4.2.1.1 Cost Modelling for Design Availability and Maintainability

Design availability and maintainability are directly related to *capital* and *operating* costs of the engineered installation. (In power-generating installations, the consequence of low availability is directly related to the cost of *replacement-power costs*.

However, replacement power costs contribute mostly to overall costs in cases of unplanned low availabilities, whereas capital and operating costs are the dominant contributors in designing for availability and maintainability.) The basic assumption underlying a reliability, availability, maintainability and safety (*RAMS*) analysis of engineering design is that there is an optimal availability or *availability range* for which capital and operating costs are at a minimum.

Designing for availability affects the engineering design's installation capital costs with respect to performance relating to process capability, functional effectiveness, and operational condition.

Designing for maintainability has an effect on the engineering design's installation capital costs with respect to systems configuration, equipment selection, maintenance, and the initial provision of contract spares. Capital costs are influenced by systems configuration, such as provision for equipment redundancy where standby equipment is required to increase *reliability*, or provision for parallel systems where *maintainability* is increased through an increase in the *modes of operation* whereby selective maintenance shutdowns are accommodated without decreasing productivity. Equipment that is more reliable is generally more costly because of higher strength and corrosion resistant materials, or because of more stringent manufacturing specifications.

Equipment selection helps to differentiate between critical sub-systems where highly reliable equipment are required, and non-critical sub-systems where the use of less reliable equipment might reduce capital costs without appreciably sacrificing system availability. Strategic application of scheduled maintenance, particularly partial and total shutdowns and provisioning of initial contract spares in the form of complete assemblies, improves maintenance downtime but also increases capital costs.

Availability and maintainability prediction methodologies in *designing for availability* and *designing for maintainability* can assist in the prediction of the engineering design's installation capital costs (with respect to systems performance, configuration, equipment selection, maintenance, and the initial provision of contract spares), to balance unavailability costs against excessive capital costs. While provisions for maintenance and initial contract spares are part of an engineering design's installation *capital costs*, scheduled maintenance and replenishment of contract spares inventory beyond the installation's warranty period usually becomes part of *operating costs*, particularly operating and maintenance (O&M) costs.

A well-developed *preventive maintenance plan*, established during the engineering design stage, will reduce overall *life-cycle costs* by improving equipment operational reliability and availability for periods of high demand, thereby reducing *operating costs*. For high-demand equipment (particular to continuous processes), the consequence of *low* availability is normally *high replacement costs*. This cost may be minimised by strategically scheduled preventive maintenance in the form of shutdowns during periods of low demand, if possible.

a) Economic Loss and the Cost of Dependency

Loss in production is due to the *unavailability* of plant and equipment as a result of the need for scheduled maintenance shutdowns, or for unplanned shutdowns because of economic operational and physical consequences of functional failure. Costs due to the unavailability of a plant as a result of unplanned shutdowns at times of high demand generally incur higher replacement costs, as they occur when the replacement cost is still considered to be less than the *loss*. Loss in production depends upon the type of process, type of equipment, the design layout or equipment configuration, the process capacity of equipment, as well as the capacity/demand relationship. The *cost* of a *loss in production* (i.e. loss in product and in production effort) during the period of lost production time is fundamentally the *cost of waste* in dependent productive resources. The cost of waste is the cost of the loss incurred as a result of *dependency* on these productive resources. The cost of the loss incurred as a result of the dependency on productive resources constitutes an *economic loss*. Economic loss can be quantified as the *cost of dependency* on productive resources (Huggett and Edmundson 1986).

Systems economic loss in production can be quantified as the cost of relying upon the system or equipment with regard to its systems configuration, the system's process output, the system's capacity surplus, and the demand on the system. *Systems dependency* can be formulated as

$$\text{Dependency} = \frac{\text{Output} - \text{Surplus}}{\text{Demand}} \times 100\% \,. \tag{4.2}$$

The measure of *systems dependency* is the system's output minus the system's capacity surplus as a ratio to the demand on the system, expressed as a percentage. The system's *capacity surplus* is the system's *design capacity* minus the *demand* on the system.

The same principle is valid at equipment level, or for the process as a whole. This can be formulated as

$$\text{Capacity Surplus} = \text{Desing Capacity} - \text{Demand} \,. \tag{4.3}$$

The measure of design capacity of a *series* of systems, sub-systems or equipment in a process is the value of the smallest design capacity of the individual capacities in the process, measured as the process output. The measure of design capacity of *parallel* systems, sub-systems or equipment in a process is the *sum* of the individual capacities in the process, measured as the process output. The measure of process output can be quantified in the form of system, sub-system or equipment output based on its *production cycle time*, *sequencing* and *utilisation*.

The *economic loss of production* can be quantified as the *cost of dependency*

$$\text{Economic loss} = \text{Cost of dependency} \tag{4.4}$$

The *cost of dependency* is the cost of a loss in production during the period that the system or equipment is down due to total or partial shutdowns. This *cost of*

dependency is, in fact, the *relative lost time cost* due to total or partial shutdown of the system or equipment at its *relative value of dependency*

$$\text{Cost of dependency} = \text{Relative lost time cost} \tag{4.5}$$

The *relative lost time cost* is calculated as the product of the *lost time* multiplied by the cost of a loss in production during the period of lost time of the system or equipment at its *relative value of dependency*. The cost of a *loss in production* during the period of lost time of the system or equipment at its *relative value of dependency* is determined by the product of the cost of the loss in production at the dependency of 100%, multiplied by the *dependency* of the system or equipment.

Thus, *relative lost time cost* can be formulated as

$$\begin{aligned}\text{Relative lost time cost} = {} & \text{Lost time} \\ & \times \text{Cost of production loss at 100\% dependency} \\ & \times \text{System or equipment dependency}\,.\end{aligned} \tag{4.6}$$

The cost of production loss at 100% dependency is considered to be the *value of lost time* of the system or equipment at 100% dependency

$$\begin{aligned}\text{Relative lost time cost} = {} & \text{Lost time} \\ & \times \text{Value of lost time} \\ & \times \text{System or equipment dependency}\,.\end{aligned} \tag{4.7}$$

Example problem In the illustration below (Fig. 4.3), three systems are in parallel configuration with a total parallel system process capacity of 1,500 tons (t) of product. System A1 has a design capacity of 600 t, system A2 has a design capacity of 500 t, and system A3 has a design capacity of 400 t. The total demand on the parallel system process is 1,000 t. What would be the *economic loss of production*, or *cost of dependency*, in the event of system A1 being down for 5 days as a result of shutdown, and then systems A1 and A2 being down for 5 days as a result of shutdown? The value of process lost time is estimated at $20,000 per day.

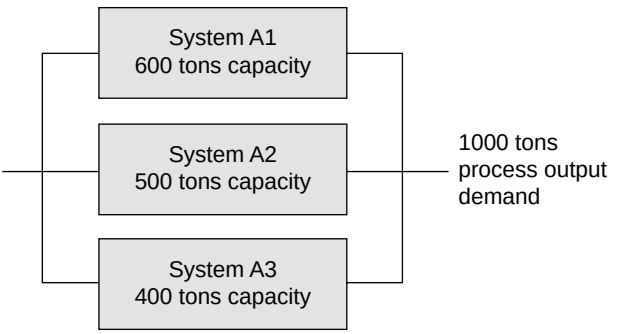


Fig. 4.3 Three-system parallel configuration system

Solution In the three-system parallel configuration system:

The dependency on system A1 is:

$$\begin{aligned}\text{Dep. A1} &= \frac{600-(1{,}500-1{,}000)}{1{,}000}\times 100\% \\ &= 10\%\,.\end{aligned}$$

The dependency on system A2 is:

$$\begin{aligned}\text{Dep. A2} &= \frac{500-(1{,}500-1{,}000)}{1{,}000}\times 100\% \\ &= 0\%\,.\end{aligned}$$

The dependency on system A3 is:

$$\begin{aligned}\text{Dep. A3} &= \frac{400-(1{,}500-1{,}000)}{1{,}000}\times 100\% \\ &= -10\%\,.\end{aligned}$$

The negative value for the dependency on system A3 is an indication of it being superfluous or redundant in the three-system configuration process, as there already exists surplus capacity from system A1 and system A2.

What is the *economic loss of production* in the event of system A1 being down for 5 days as a result of downtime? The *economic loss of production* can be quantified as the *cost of dependency* on the systems, where the *cost of dependency* is the *relative lost time cost* due to downtime of the systems at their *relative value of dependency*:

$$\text{Economic loss} = \text{Cost of dependency} = \text{Relative lost time cost}\,.$$

The *relative lost time cost* is calculated as the product of the actual lost time multiplied by the *value* of the actual lost time of the systems at 100% dependency, multiplied by the systems' *dependency*:

$$\begin{aligned}\text{Relative lost time cost} &= \text{Lost time} \\ &\quad \times \text{Value of lost time} \\ &\quad \times \text{System or equipment dependency}\end{aligned}$$

$$\begin{aligned}\text{Relative lost time cost} &= 5\text{ days}\times \$20{,}000/\text{day}\times 10\% \\ \text{for system A1:} &= \$10{,}000\,.\end{aligned}$$

In the event that system A1 experiences downtime, the dependencies on systems A2 and A3 change drastically. With system A1 down, now the parallel system capacity of systems A2 and A3 is 500 plus 400 t. The process output demand still remains at 1,000 t. What is the dependency on each of systems A2 and A3 in the two-system parallel configuration process?

The dependency on system A2 is:

$$\begin{aligned}\text{Dep. A2} &= \frac{500-(900-1{,}000)}{1{,}000}\times 100\% \\ &= 60\%\,.\end{aligned}$$

The dependency on system A3 is:

$$\begin{aligned}\text{Dep. A3} &= \frac{400-(900-1{,}000)}{1{,}000}\times 100\% \\ &= 50\%\,.\end{aligned}$$

What is the *economic loss of production* in the event of systems A1 *and* A2 being down for 5 days as a result of downtime?

$$\begin{aligned}\text{Relative lost time cost} &= 5\text{ days}\times \$20{,}000/\text{day}\times 10\% \\ \text{for system A1:} &= \$10{,}000\end{aligned}$$

$$\begin{aligned}\text{Relative lost time cost} &= 5\text{ days}\times \$20{,}000/\text{day}\times 60\% \\ \text{for system A2:} &= \$60{,}000\end{aligned}$$

$$\begin{aligned}\text{Relative lost time cost} \\ \text{for systems A1 and A2:} &= \$70{,}000\end{aligned}$$

A point of interest is that the dependencies and relative lost time costs are calculated from the viewpoint that *first* system A1 goes down, then *secondly* system A2. Would there be a difference in the calculations if system A2 went down first, followed by system A1? Thus, what is the *economic loss of production* in the event of system A2 being down for 5 days, and then systems A2 *and* A1 being down for 5 days?

In the three-system parallel configuration process:

The dependency on system A2 is:

$$\begin{aligned}\text{Dep. A2} &= \frac{500-(1{,}500-1{,}000)}{1{,}000}\times 100\% \\ &= 0\%\,.\end{aligned}$$

The *cost of dependency* is the *relative lost time cost* due to functional failure of the equipment at its *relative value of dependency.*

$$\begin{aligned}\text{Relative lost time cost} &= 5\text{ days}\times \$20{,}000/\text{day}\times 0\% \\ \text{for system A2:} &= \$0\,.\end{aligned}$$

If system A2 experiences downtime first, what is the dependency on system A1 in the two-system parallel configuration process?

The dependency on system A1 is:

$$\begin{aligned}\text{Dep. A1} &= \frac{600-(900-1{,}000)}{1{,}000}\times 100\% \\ &= 70\%\,.\end{aligned}$$

What is the *economic loss of production* in the event of systems A2 *and* A1 being down for 5 days as a result of downtime?

$$\begin{aligned}\text{Relative lost time cost} &= 5\ \text{days}\times \$20{,}000/\text{day}\times 0\% \\ \text{for system A2:} &= \$0\end{aligned}$$

$$\begin{aligned}\text{Relative lost time cost} &= 5\ \text{days}\times \$20{,}000/\text{day}\times 70\% \\ \text{for system A1:} &= \$70{,}000\end{aligned}$$

$$\begin{aligned}&\text{Relative lost time cost} \\ &\text{for systems A2 and A1:} = \$70{,}000\end{aligned}$$

Thus, the relative lost time cost for systems A1 and A2 remains the same irrespective of which system goes down first.

b) Life-Cycle Analysis and Life-Cycle Costs

Cost modelling for design availability and maintainability needs to take into consideration scheduled as well as unscheduled shutdowns that involve an indirect economic loss, such as the loss in production, as well as the direct cost of maintenance action. This maintenance action implies a direct cost that includes the cost of maintenance labour and maintenance materials such as lubricants, greases, etc., and spare parts. Traditional analysis of engineering design has focused primarily on a system's operational performance without much consideration of the costs of the manufacturing and installation stages downstream from design. In contrast, life-cycle analysis of an engineered installation, particularly during its initial development, can play a crucial role in determining the installation's overall life-cycle cost and useful lifespan inclusive of the concept of residual life. The design and development of engineered installations involve balancing a series of factors to specify, manufacture and install systems that perform a specific set of operational functions. These factors influence both the overall system definition, as well as each stage within the system's development life cycle. These design and development factors include (Lee et al. 1993):

- *Design requirements*:
  - input demand
  - output volume

  - required functionality
  - operating environment
  - design integrity.

- *Time constraints*:
  - design phases
  - development stages
  - manufacture lead time
  - operational life
  - maintenance downtime.

- *Operational issues*:
  - evolutionary/revolutionary design
  - new/proven technologies
  - operations experience
  - development/support infrastructure.

- *Life-cycle costs*:
  - design/development
  - manufacture/construction
  - fabrication/installation
  - operation/maintenance
  - renewal/rehabilitation
  - disposal/salvage.

The assessment of system performance from a total life-cycle perspective (i.e. across all life-cycle stages) is defined as *system life-cycle analysis*. System life-cycle analysis is viewed as a superset of analysis methods centred about a system's life-cycle stages. The analysis seeks to qualitatively and quantitatively measure performance both at the system and/or equipment life-cycle stages, as well as across the total engineered installation life cycle, from design to possible salvage.

For system life-cycle analysis, the primary focus is on determining the optimal design of a system with respect to the required design criteria, while concurrently measuring the impact of design decisions on the other life-cycle stages, such as manufacture/construction/fabrication/installation/operation/maintenance/renewal/rehabilitation. Similarly, the procedure of measuring the effects of design *and* development decisions on a system's *operational performance* in an overall life-cycle context is defined as *life-cycle engineering analysis* (Lee et al. 1993).

This is an extension of engineering analysis methods that are applied during the conceptual, preliminary and detail design phases, and are used to quantify system operational performance such as static and dynamic loading behaviour, thermal operational performance, system control response, etc. Life-cycle engineering analysis extends current engineering analysis approaches by applying these to other life-cycle stages (such as thermal behaviour analyses under manufacturing processes and burn-in testing), and assessing life-cycle performance trade-offs, particularly at

the renewal/rehabilitation stages. Engineering design project management includes life-cycle engineering analysis as the measurement of system operational performance in a life-cycle context. The issues critical to life-cycle engineering analysis include system performance analysis and performance regimes, system life-cycle data modelling and analysis, performance trade-off measurement, and problems of life-cycle engineering analysis in the context of complex integrated systems.

Life-cycle costs *Life-cycle costs (LCC)* are total costs from inception to disposal for both equipment and projects. The objective of *LCC* analysis is to choose the most cost-effective approach from a series of alternatives so that the least long-term cost of ownership is achieved. Analytical estimates of total costs are some of the methods for life-cycle costs (Barringer et al. 1996).

LCC is strongly influenced by equipment design, installation/use practices, and maintenance practices. Life-cycle costs are estimated total costs that are incurred in the design, development, production, operation, maintenance and renewal/disposal of a system over its anticipated *useful life*. *LCC* analysis in engineering design helps designers justify equipment and process selection based on total costs, rather than estimated procurement costs. The sum of operation, maintenance and disposal costs far exceeds procurement costs. Procurement costs are widely used as the primary (and sometimes only) criteria for equipment or system selection because they are relatively simple criteria, though often resulting in insufficient financial data for proper decision-making.

Life-cycle costs consist fundamentally of *acquisition* and *sustaining* costs, which are not mutually exclusive. Acquired equipment always includes extra costs to sustain the acquisition. Acquisition and sustaining costs are determined by evaluating the life-cycle costs and conducting sensitivity analysis to identify the relative cost drivers (Fabrycky et al. 1991).

In general, acquisition costs include the following:

- Capital investment and financial management
- Research & development, engineering design, and pilot tests
- Permits, leases and legal fees, indemnity and statutory costs
- Engineering and technical data sheets and specifications
- Manufacturing/construction, fabrication and installation
- Ramp-up and warranty, modifications and improvements
- Support facilities and utilities and support equipment
- Operations training and maintenance logistics
- Computer management and control systems.

In general, sustaining costs include the following:

- Management, consultation and supervision
- Engineering and technical documentation
- Operations and consumption materials
- Facility usage and energy consumption
- Servicing and maintenance consumables
- Equipment replacement and renewal

- Scheduled and unscheduled maintenance
- Logistic support and spares supply
- Labour, materials and overhead
- Environmental green and clean
- Remediation and recovery
- Disposal, wrecking and salvage.

The cost of sustaining equipment can be from 2 to 20 times the equipment acquisition cost over its useful lifespan. The first obvious cost of hardware acquisition is usually the smallest amount that will be spent during the life of the acquisition, whereas most sustaining expenses are not obvious. For sustaining costs, the categories most difficult to quantify are facility usage and energy consumption costs, equipment replacement and renewal costs, scheduled and unscheduled maintenance costs, and logistic support and supply costs.

Most capital equipment estimates ignore major portions of the sustaining costs, as they lack sufficient quantification to justify their inclusion. Even when provisions for failure costs are included, they appear as a percentage of the initial costs, and are spread evenly as economic loss due to shutdowns throughout the typical life of the engineered installation. However, for wear-out failure modes, the analysis is censored by not including failures in the proper time span. Most of the total estimated costs are usually fixed when the equipment is specified during design, and any decisions concerned with equipment selection are then based on acquisition costs that constitute the smallest portion of total *LCC* (Barringer 1998).

c) Life-Cycle Cost Elements in Engineering Design

In order to estimate life-cycle costs during the engineering design process, all the appropriate *cost items* must be identified. As indicated previously, *LCC* consist fundamentally of *acquisition* and *sustaining* costs, which are made up of a number of cost items that can be grouped into cost categories as illustrated in Fig. 4.4. A cost item is the smallest cost that is calculated or estimated as a separate entity. The number of cost items used depends upon the particular phase in the engineering design process at which the calculation is carried out. The set of cost items is developed in parallel with the development of a *work breakdown structure (WBS)*, and it is essential to tie a cost item to the design project *scope of work* and related design *work packages* at a certain system hierarchy level of the *WBS* (Aslaksen et al. 1992).

The level is chosen so that responsibility for a cost item can be individually assigned to a specific *task*. However, while a task is analysed by decomposing it into activities chosen from a predefined set, the cost of executing a task is calculated by decomposing it into *cost types*, chosen from a predefined set. This set is in itself developed in a structured or hierarchical fashion as the engineering design process develops. At the highest level, there are only three cost types: labour costs, material costs, and capital costs.

A cost item is thus identified by one element from each of two index sets—the set of tasks and the set of cost types. In addition, there must be an indication of *when* each cost item is to be incurred in the life cycle of the engineered installation. Consequently, a cost item is identified by three index values: the task at a certain level of the *WBS*, the cost type relating to the particular task, and the occurrence of the task in the life-cycle span of the engineered installation. In other words, the representation of life-cycle cost items is three-dimensional. In developing the set of cost items, the most difficult part is to develop the *WBS* in conjunction with the design project scope of work, as this *WBS* must encompass all the work associated with designing, manufacturing, constructing, installing, commissioning, operating and maintaining the system over its lifetime. Thus, for *LCC*, it is not enough to consider only the procurement costs of the equipment, or the costs of the engineering effort—instead, all of the *acquisition* and *sustaining* costs relevant to the cost categories illustrated in Fig. 4.4 must be considered.

Complementary to the *acquisition* and *sustaining* cost items listed previously, some typical life-cycle cost items that should be identified during the engineering design process, relevant to the defined cost categories for the engineered installation in its total life cycle, are the following.

LIFE CYCLE STAGES	COST CATEGORIES
Feasibility / Conceptualisation Preliminary and Detail Design	Specification Costs
Construction / Fabrication Procurement / Installation Commissioning / Warranty	Establishment Costs
Operation / Utilisation Maintenance / Modification Renewal / Rehabilitation	Utilisation Costs
Decommissioning Disposal / Salvage / Demolition	Salvage Costs

Fig. 4.4 Life-cycle costs structure

Specification costs

- *Research and development*:
  The costs of any investigations and feasibility studies carried out specifically to support or create the technology needed for the engineered installation, an allocated share of the costs of more general R&D programs, and license fees for the use of technology.
- *Analysis*:
  The costs of *financial and technical due diligence* evaluations, environmental impact studies, market investigations, inspecting existing systems, system analysis, and developing the system specification and initial conceptual design studies.
- *Design*:
  The costs of all activities connected with producing the complete set of system specifications, such as modelling, simulation, optimisation and mock-ups; developing databases; producing drawings, parts lists, engineering reports and test requirements; and developing the specifications per se.
- *Integration and tests*:
  The costs associated with setting up test facilities, rental of test equipment, interface verification, sub-system tests, modifications resulting from unsatisfactory test results, system acceptance tests and test documentation.

Establishment costs

- *Construction*:
  The costs associated with site establishment, site works, general construction, support structures, onsite fabrication, inspection, camp accommodation, wet mess, transportation, office buildings, permanent accommodation, water supply, workshop facilities, special fixtures, stores, and any costs resulting from setting up auxiliary facilities for the supply and storage of support services.
- *Fabrication*:
  The costs associated with fabricating systems and assemblies, setting up specialised manufacturing facilities, manufacturing costs, quality inspections, transportation, storage and handling.
- *Procurement*:
  The costs associated with acquiring material and system components, including warehousing, demurrage, site storage, handling, transport and inspection.
- *Installation*:
  The costs of auxiliary equipment and facilities (e.g. air-conditioning, power, lighting, conduits, cabling), site inspections, development of installation instructions and drawings.
- *Commissioning*:
  The costs associated with as-built non-service inspections, in-service inspections, wet-run tests, and initial start-up costs (utilities, fuel).
- *Quality assurance*:
  The costs of carrying out quality assurance, such as vendor qualification, inspections and verifications, test equipment calibration, and the documentation of standards, and all types of quality assurance audits.

Utilisation costs

- *Operation*:
  All costs associated with the human operation of the system (e.g. wages and salaries, social costs, amenities, transportation, transit accommodation), material and fuel costs, as well as energy costs, taxes, licenses, rents and leasing costs, and continual site preparation costs for later restoring the site to its original condition.
- *Maintenance*:
  All costs resulting from carrying-out essential warranty maintenance, as well as routine, preventive and corrective maintenance, including the costs of materials (i.e. consumables and spare parts), labour, and monitoring and fault-reporting systems.
- *Documentation*:
  The costs associated with developing, producing and maintaining all documentation, such as operating and maintenance manuals, spare parts lists, cabling schedules, etc.
- *Training and induction*:
  The costs of developing training courses, writing training manuals, conducting training, assessing training needs and providing training facilities, as well as the costs of attending induction training.

Recovery costs

- *Decommissioning and site amelioration*:
  The costs associated with decommissioning engineered installations including all payments due to termination of operations, such as dismantling and disposing of equipment, environmental protection, plus costs associated with restoring a site to its original condition.

Life-cycle cost models LCC models may vary according to different system applications in engineered installations. There are thus various LCC models used to estimate costs based on the specific needs of designers, manufacturers and users of an engineered installation. In principle, the general LCC model may be formulated as representing either *acquisition* and *sustaining* costs, or the previously defined cost categories for the engineered installation in its total life cycle.

The LCC model representing *acquisition* and *sustaining* costs can be formulated as

$$\mathrm{LCC}_{\alpha\beta} = \alpha + \beta \,, \tag{4.8}$$

where

$$\alpha = \sum_{i=1}^{m} C_{\mathrm{A}_i} \tag{4.9}$$

m = number of acquisition cost categories
C_{A_i} = ith acquisition cost element

and

$$\beta = \sum_{j=1}^{n} C_{\mathrm{S}_j} \tag{4.10}$$

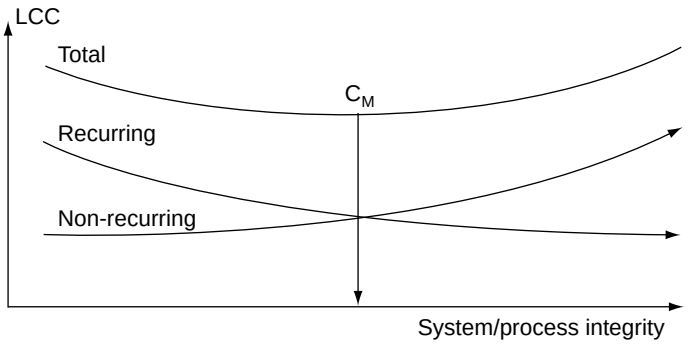


Fig. 4.5 Cost minimisation curve for non-recurring and recurring LCC

n = number of acquisition cost categories
C_{S_j} = jth sustaining cost element.

The LCC model representing *acquisition* and *sustaining* costs, where the *acquisition* costs can be considered to be non-recurring costs, and the *sustaining* costs to be recurring costs, has an optimum when compared to overall system or process integrity (availability and maintainability).

LCC and design integrity, as a figure-of-merit, is considered later. This may be represented as a cost minimisation curve, which is illustrated in Fig. 4.5 (Dhillon 1983).

The LCC model representing the previously defined cost categories for the engineered installation in its total life cycle can be formulated as

$$\text{LCC} = C_S + C_E + C_U + C_R \tag{4.11}$$

where:

C_S = specification costs
C_E = establishment costs
C_U = utilisation costs
C_R = recovery costs.

d) Present Value Calculations for Life-Cycle Costs

It is not sensible or even very useful to simply add up all the estimated costs for the life cycle of the system. Because of the cost of capital (i.e. interest) and inflation, costs incurred at different times have a different relative value and, to compare these, they must be discounted with the appropriate discount rate. To determine an effective cost of capital, the investment capital is discounted by a commercial interest rate that depends on the *risk* associated with the project, plus any commissions and charges. These effective costs of capital, as well as ownership costs (i.e. the

recurring costs of operating and maintaining the system), are not necessarily equal amounts per unit of time. For simplicity, discounting by a series of equal payments may be applied by introducing an *effective discount rate*.

For the purpose of optimising the *LCC* of an engineered installation, the accounting approach of *discounted cash flow (DCF)* is adopted. There is no intrinsic advantage in using either present value calculations or the future value. However, expressing the cost of capital as a separate cost item does have advantages in that the periodic value of this cost is an accounting item that will affect the cash flow in each period, and costs associated with providing capital (e.g. fees for available, unused credit) may be easily and consistently accounted for and included in the *LCC*.

In the approach to using present value calculations for a discounted cash flow analysis, the yearly cash flows are discounted back to the beginning of year 1 (or end of year 0), using a *present value factor* that takes into consideration the inflation rate, usually modified to reflect compound interest (calculated and added to, or subtracted from the capital) every unit of time. The result is *net present value (NPV)* (Bussey 1978). A major impediment is that the magnitudes and timing of all the cash flows are not correctly taken into account. This is essentially true of all but three decision criteria methods—net present value (NPV), internal rate of return (IRR), and profitability index (or the *benefit-cost ratio*). Under certain conditions, these three criteria can be properly applied to the design project acceptance problem. This is particularly the case with estimates of *NPV* and *IRR* of estimated life-cycle costs during the engineering design stage. These criteria are the so-called *rational criteria* because they take into account the two attributes most often absent in other criteria:

- The entire cash flow for the life of the project
- The time value of money.

The net present value criterion The general expression for *net present value* (NPV), P_0, is the following

$$P_0 = \sum_{t=0}^{N} \frac{Y_t}{\prod_{j=0}(1+i_j)} \tag{4.12}$$

where:

$Y_t =$ the net cash flow at the end of period t
$i_j =$ the interest (discount rate) for period j
$N =$ the life of the project
$j =$ points in time prior to t (i.e. $j = 0, 1, 2, \ldots, t$)
$t =$ the point in time (i.e. $t = 0, 1, 2, \ldots, N$).

Thus, in the general form, it is not necessary for the interest rates to be equal, which permits a period-by-period evaluation in which the interest rate can take on different values. Usually for project evaluation, however, the interest rate is assumed to be constant throughout, whereby the general expression for *NPV* reduces to

$$P_0 = \sum_{t=0}^{N} Y_t (1+i)^{-t} \tag{4.13}$$

where:

$Y_t =$ the net cash flow at the end of period t.

In applying *NPV*, the net cash flows are usually available as input data, which are assumed to occur instantaneously at the ends of the periods t. Also usually known is an estimate of the discounting rate, i, to be used. Under these conditions, finding the *NPV* is straightforward. The result is a point estimate of a single value at a particular interest rate i. While the point estimate of *NPV* is informative, in that one can determine if it is positive, negative, zero or indeterminate, the behaviour of the criterion as a function is more informative. For the case of unconstrained assumptions concerning project acceptance, the general rule is to accept the project if the *NPV* is positive. This is true when the present value of the cash inflows exceeds the present value of the cash outflows (Bussey 1978).

The *NPV* decreases with increasing discount rate. This is true of any project for which the cash flow increases on average throughout the project. Secondly, if the cash flow is negative in the first part of the project, as is true of any project requiring an initial capital investment, there exists some discount rate for which the *NPV* becomes zero. This is known as the *internal rate of return (IRR)*. The *IRR* constitutes the most useful single characterisation of the financial viability of a project. It represents the *break-even discount rate* that will just allow repayment of the initial investment. If the actual discount rate (i.e. interest rate plus any other related financial charges) is less than the *IRR*, a profit will result. However, if the discount rate is higher than the *IRR*, the *NPV* will be negative, and the project will result in a loss, prompting the need for redesign of critical systems of the proposed engineered installation, or outright rejection of an engineering project's particular technology, or even of the project itself.

In the alternative approach of using future value, rather than present value, the estimated life-cycle costs over the project lifetime are reflected for each significant period in the project's life-cycle stages, calculated from the required capital and the interest rate for that period. Subtracting these estimated life-cycle cost of capital from each period's expected net cash flows yields the *future net value*. As expected, it also goes to zero as the discount rate reaches the *IRR*, which is independent of the method of calculation. Net profit, or future net value, results from subtracting the cost of capital from the net cash flow.

The internal rate of return criterion The net present value, described in the previous sub-section, depends upon the knowledge of an external interest rate for its application (i.e. external to the project, such as the cost of capital). The *internal rate of return (IRR)* method is closely related to *NPV* in that it also is a discounted cash flow method, but it seeks to avoid the arbitrary choice of an interest rate. Instead, it attempts to find some interest rate, initially unknown, which is internal to the project.

The procedure is to find an interest rate that will make the present value of the cash flow of a project zero—that is, some interest rate that will cause the present value of cash inflows to equal the present value of cash outflows. *IRR* is defined as the interest rate i that will cause the net present value P to become zero. Thus, *IRR*

is the value i such that

$$P(i) = \sum_{t=0}^{N} Y_t (1+i)^{-t} = 0 \,. \tag{4.14}$$

The *IRR* must be found by trial and error or by a computer search algorithm technique, since it is an unknown root (or roots) of a polynomial in i. Thus, if we start with known values of each cash flow, we can possibly find one or more values that will make the above equation true. These values, if they exist as real numbers, are known as the project's *internal rate of return*, or the *economic yield* of the project (in contrast to the *economic loss* considered previously). The selection criterion is to accept the design project if the *IRR* is greater than the *marginal investment rate*; otherwise, the project is rejected or, as in the case of a negative *NPV*, the result is the redesign of critical systems or the rejection of the project's specific technology, rather than of the project itself.

Internal rate of return (IRR) has long been advocated as a project acceptance criterion because, in this criterion, the interest rate is the unknown value that relates project returns to capital investment outlays. In the sense that it is the functional value to be established by the expected cash outlays and inflows of the project itself, it has been called the *internal* rate of return. However, in many cases, the economic meaning of *IRR* as a selection criterion of a design project or proposed engineered installation is not fully understood. For example, the *IRR* is not only the interest rate that causes the net present value of the cash flow of a project to be zero, but it is also the interest rate that causes exact recovery of investment over the life of the project, plus a return on the un-recovered investment balances during the life of the project. A common misinterpretation of *IRR* is that it is an interest rate expressing a rate of return on the *initial* investment. This is not so. If the *IRR* is applied periodically to the initial investment only, then the cash flows fail to recover the initial investment plus interest at the end of the project life. The fundamental economic meaning of *IRR* is the rate of interest earned on the time-varying, un-recovered balances of investment, such that the final investment balance is zero at the *end* of the project's life. Since the *IRR* does not measure the return on initial investment, it has meaning only when the level of investment is considered along with all the other cash flows of the project, relative to the project's total life-cycle costs. These are estimated total costs incurred in the design, development, production, operation, maintenance, support and final disposition of the proposed engineered installation over its anticipated *useful life* (Aslaksen et al. 1992).

Internal rate of return as a figure-of-merit Under the assumptions of *certainty*, it is sometimes possible to use *internal rate of return* as a figure-of-merit for determining whether a particular design project should be undertaken. It could thus be viewed as an economic trade-off measure to assess the conditions under which the *IRR* may be used as a selection criterion, and *when it may not*. One of the main problems encountered in using *IRR* as such a criterion is the concept of *multiple and indeterminate rates of return*. When attempting to obtain the internal rate of return with certain forms of cash flow, it is possible to find either that a unique solution does not exist for the *IRR*, and more than one interest rate will satisfy the

formula, or that no solution can be found at all. When more than one solution exists mathematically, the cash flow is said to yield *multiple rates of return* and, when no solution exists, it is said to have an *indeterminate rate of return*.

e) Trade-Off Measurement for Life-Cycle Costs

LCC needs to be calculated early in the engineering design stage, to influence final design outcomes of the proposed engineered installation. Making major changes in *LCC* after engineering design projects reach the production stage is often not possible. *LCC* helps determine optimal maintenance and repair shutdown cycles of inadequately assessed engineering installations subject to frequent repair at great expense. Sufficient financing is seldom available to design the project correctly but, somehow, there is always money available to make major modifications to poorly configured engineering design installations (Barringer 1998). Consequently, trade-off measurement methods for *LCC* early in the life cycle become essential. The *cost effectiveness (CE) equation* is one method for *LCC* trade-off calculations involving operational and failure probabilities. It offers a figure-of-merit, and measures the chances of achieving the intended final design results against predefined life-cycle costs. The effectiveness equation has been described in several different formats (Aslaksen et al. 1992; Kececioglu 1995; Pecht 1995; Blanchard et al. 1995). Each element is a probability. The issue, however, is finding a system effectiveness value that gives lowest long-term cost of ownership with trade-off considerations.

Cost effectiveness and life-cycle costs *Cost effectiveness* (CE), as viewed from a systems engineering perspective, can be defined as the ratio of *system effectiveness* (SE) to its *life-cycle cost* (LCC; Aslaksen et al. 1992), which is expressed in the following relationship

$$\mathrm{CE} = \frac{\mathrm{SE}}{\mathrm{LCC}} \,. \tag{4.15}$$

In this context, SE is expressed in dollars; so, CE will be a dimensionless parameter. It is apparent that the evaluation of CE could be separated into the evaluation of SE and the evaluation of LCC. The definition therefore leads to a conceptually simple, universal criterion governing all engineering decisions—the decision is good if it results in an increase in cost effectiveness. This criterion is appropriate for engineering decisions—however, it may not always be entirely suitable for investment decisions, and there is a significant difference between cost effectiveness and *IRR* as a figure-of-merit.

System effectiveness *System effectiveness (SE)* can be defined as a measure of how well a system will perform the functions that it was designed for, or how well it will meet the requirements of the system specification. It is often expressed as the *probability* that the system can successfully meet an operational demand within a given operating time under specified conditions. This definition implies a number of important aspects:

- Operating time may be critical, and *SE* is often a function of time.
- Maintenance is not excluded; and the specified conditions will in most cases include both scheduled and unscheduled shutdowns.
- Operational demand implies that there are two separate classes of system failures:
  - The system can be in an inoperable condition when needed.
  - The system can fail to perform sufficiently during the required operating period.
- The inclusion of both operational demand and specified conditions shows that possible failure (i.e. failure to meet operational demand) and the conditions under which the system is intended to be utilised (i.e. operational stresses) are related.

It thus follows that, while *SE* is obviously influenced by system design, it is equally influenced by the way the system will be used and maintained by the logistic system that supports its operation. The definition expressed in terms of a probability is particularly useful for systems that are required to operate for a prescribed, relatively short period (i.e. systems fulfilling an intermittent task, as is the case for periodic operational requirements). For most other systems, however, the period of operation is the lifetime of the system, and this is usually very long, compared with the timescales for other events affecting the system, such as shutdowns, etc.

As a result, the system settles into a steady state that is characterised by an average performance or, more specifically, by an average deviation from design specification performance. However, as the performance of a system is usually a complex multi-dimensional variable, measuring it in terms of a probability is not very appropriate. The proper approach is to determine the decrease in the value of the system as a function of the decrease in performance. The definition of *SE* can thus be formulated as the *value of the system* over its *design lifetime* (Aslaksen et al. 1992).

Factors affecting the value of a system: Every system has some value—otherwise, its development would not even be contemplated. Furthermore, this value must in some way depend on how well the system performs; if it did not perform at all, its value would be zero. The problem arises in trying to move from a qualitative statement, such as 'improved availability', to a quantitative statement such as 'increase in availability from 0.85 to 0.90 is worth $3.285 million'. It is then found that the value of a system, particularly its dependence on various performance parameters, is often a highly subjective matter. Nevertheless, it is a problem that must be solved because, without assigning some value to system performance, there is no basis for taking rational engineering decisions with respect to its cost effectiveness.

Design effectiveness and life-cycle costs *Design effectiveness* (DE) for LCC trade-off calculations involves probabilities of design integrity criteria (i.e. reliability, availability, maintainability and safety) offering a figure-of-merit that measures the chances of achieving the intended final design results against integrity constraints (Blanchard et al. 1995). Such an effectiveness equation is of the following format

$$\text{Design effectiveness (DE)} = \frac{\text{System effectiveness (SE)}}{\text{Life-cycle costs (LCC)}} \,. \tag{4.16}$$

LCC in this case is a measure of resource usage that cannot include all possible cost elements but must include critical cost items.

System effectiveness System effectiveness in this case is a measure of integrity (although it rarely includes all integrity elements, as many are often too difficult to quantify). Based on probability, it varies from 0 to 1. Thus:

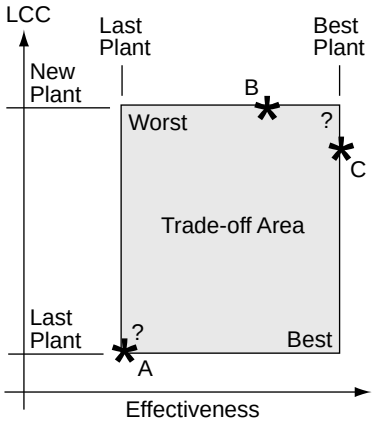
$$\text{System effectiveness} = \text{Design integrity} \times \text{Capability} .$$

Design integrity is reliability/availability/maintainability/safety, and capability is product of efficiency multiplied by utilisation.

System effectiveness quantifies important elements of design integrity and life-cycle costs to find areas for improvement to increase overall effectiveness and to reduce *economic loss*.

For example, if availability is 98% and capability is 65%, the opportunity for improving capability is usually much greater than for improving availability. System effectiveness in this context is helpful for understanding benchmarks and future possible status for *LCC* trade-off information. Figure 4.6 gives a graphical presentation of effectiveness and life-cycle costs. Although the preference is to select engineering designs, or projects that have low life-cycle costs and high effectiveness, this may often not be accomplished in reality (Barringer 1998).

Capability deals with productive output compared to *inherent* productive output. This index measures the systems capability to perform the intended function on a system basis, and can be expressed as the product of efficiency multiplied by utilisation. Efficiency measures the expected productive work output versus the work input. Utilisation is the ratio of expected time spent on productive effort to the total operational time. For example, suppose efficiency is estimated at 80% and utilisation is 82.19% because the operation is conducted 300 days per year out of 365 days: the capability is $0.8 \times 0.8219 = 65.75\%$. Capability measures *how well* the



Parameter	A Last Plant	B New Plant	C Best Plant
Availability	0.95	0.95	0.98
Reliability	0.3	0.4	0.6
Maintainability	0.7	0.7	0.7
Capability	0.7	0.8	0.6
Effectiveness	0.14	0.22	0.25
LCC	80	100	95

Fig. 4.6 Design effectiveness and life-cycle costs (Barringer 1998)

production activity is performed compared to the datum (Barringer 1998). A more comprehensive and, in fact, mathematically correct definition of *process capability* is considered in Sect. 4.2.1.2.

Availability and maintainability compared to IRR as figure-of-merit for LCC Putting aside the elements of reliability and safety in the design integrity equation in this chapter, the significance of availability and maintainability in design effectiveness and life-cycle costs is specifically considered. Availability deals with the duration of *uptime* for systems and equipment. Availability characteristics are usually determined by the expected operational conditions, which then impact upon operational procedures and the expected durations of productive time. Availability measures how productive time is used. Thus, as availability increases, because the systems and equipment are functional and operational for a longer period of time, so also does the potential for an increase in the *IRR*.

Maintainability deals with the duration of *downtime* for maintenance outages, or *how long* it takes to complete maintenance actions compared to a standard. Maintainability characteristics are usually determined by engineering design, which then impacts upon maintenance procedures and the expected durations of shutdowns. A key maintainability figure-of-merit is the mean time to repair (MTTR) compared to a limit for the maximum repair time. Qualitatively, it refers to the ease with which systems and equipment are restored to a functioning state. Quantitatively, it is a probability measure based on the total downtime for maintenance. Maintainability measures the probability of timely repairs. Thus, as maintainability increases, because systems and equipment are down for a shorter period of time, so also does the potential for increase in the *IRR*.

4.2.1.2 Availability Modelling Based on System Performance

System performance, in the context of *designing for availability*, can be perceived as the combination of a system's *process capability* with regard to the process characteristics of *capacity*, *input*, *throughput*, *output* and *quality*, a system's *functional effectiveness* with regard to the functional characteristics of *efficiency* and *utilisation*, as well as consideration of a system's *operational condition* with regard to operational measures such as temperatures, pressures, flows, etc. All these characteristics may serve as useful indicators in designing for availability *without having to formulate the specific operational variables of each individual system*, and to consider instead a system's capability, effectiveness and condition.

In order for designers to be confident about using novel manufacturing processes, and still achieve the necessary availability constraints during the design of engineered installations, a more intimate dialogue between engineering design and manufacturing is necessary. Ideally, all aspects of the manufacturing process should be accessible and understood. For example, designers should be able to run process simulations, at either a superficial or detailed level, on partial or whole designs. Design engineers should be able to obtain 'manufacturability' and 'constructability'

rules and guidelines that can be loaded directly into the relative engineering designs' *computer aided design (CAD)* environments. Furthermore, design engineers should be able to load processing constraints (e.g. materials or feature dimensions) into their *CAD* systems and have these checked and enforced before submitting designs to manufacturers. In this way, designers can become familiar with the design's specific manufacturing and construction requirements. In some cases, an overview description of the process capabilities may suffice. In other cases, when it is essential to minimise manufacturing costs, or to meet stringent demands on design specifications or material properties, the designer must have detailed information on the specific design's manufacturing characteristics and constraints. An important aspect of being able to submit designs with confidence that the end result will meet design specifications is the adoption of conservative design rules that specify design features that are manufacturable (Mead 1994).

A variety of *CAD* tools can be used that provide a standard mechanism by which designers can obtain process *capability models* from disparate processes, and load these into their *CAD* environments. Specifically, the mechanism should enable designers to acquire capability models that can be used to compute a system's process capability, functional effectiveness, operational condition and manufacturability of evolving designs with accuracies necessary to meet the design requirements. (In this context, 'manufacturability' includes both the ease of fabrication and the ease of assembly/construction, as considered by Taguchi's methodology for implementing *robust design*; Taguchi 1993.)

Robust design (RD) is an important methodology for improving the design's manufacturability and for increasing process system stability. Since its introduction to the US industry in 1980, Taguchi's approach to quality engineering and robust design has received much attention from designers, manufacturers, statisticians, and quality professionals.

Essentially, the central idea in robust design is that variations in a process system's performance can inevitably result in poor quality and monetary losses during the system's life cycle. The sources of these variations can directly be classified into the two categories of *controllable* and *uncontrollable* parameters. In a typical design application, factors such as geometric dimensions (sizing) of equipment can easily be controlled by designers. Uncontrollable factors, such as environmental variables, component deterioration or manufacturing imperfections, are also sources of variations having effects that cannot be eliminated, and must especially be considered in designing for availability. Therefore, *RD*'s main function is to reduce a design's potential variation by reducing the *sensitivity* of the design to the *sources* of variation, rather than by controlling these sources. In other words, *RD* reduces potential system response variations by designing appropriate capability model settings for controllable parameters, in order to dampen the effects of hard-to-control variables. Taguchi's methodology for implementing robust design is essentially a four-step procedure that includes not only formulating the design problem but planning, analysing and verifying the design results as well (Taguchi et al. 1989).

A communication mechanism should also allow unsolicited information, such as updates on process capabilities, to be transmitted from manufacturing facilities

to designers. To enable such a dialogue between designers and manufacturers, the following issues must be addressed:

- How is process capability or functional effectiveness represented?
- How are capability models located and acquired by the designer?
- How are capability models mapped into the design space?
- How is information contained in these models applied throughout engineering design?

These issues require some further form of methodology for information exchange, not only between engineering design teams but also between designers and manufacturers. Such a methodology should include an object-oriented architecture that expedites the task of combining *CAD* environments with process manufacturing and/or construction planning, a mechanism for knowledge representation that enables the exchange of design integrity information, and a communication protocol between designers and manufacturers.

Such a methodology addresses the possible and practical application of *artificial intelligence (AI)* modelling techniques, with the inclusion of *knowledge-based expert systems* within a *blackboard model*, in the development of intelligent computer automated methodology for determining the integrity of engineering design. The blackboard model provides for automated continual design reviews of engineering design, including communication protocols and an object-oriented language that allows segregate design groups to remotely exchange collaborative information via the internet (McGuire et al. 1993; Olsen et al. 1995; Pancerella et al. 1995).

On this basis, different engineering design expertise groups, and manufacturing companies specialising in specific engineering systems can concurrently participate in *collaborative design* from around the world, whereby input of design parameters and criteria into a blackboard model provides for automated continual design reviews of the overall engineering design. Such a blackboard model, together with its knowledge-based expert systems, must be suitable both in programming language efficiency and in communication protocols for internet application.

a) Process Capability

In the context of industrial processes, the definition of *process* is "*a series of operations performed to produce a result or product*", and *capability* is defined as "*effective action*". *Process capability* can thus be defined as "*the effective action of a series of operations to produce a result or product*".

A *process capability model* is a mathematical model that compares the *behaviour* of a *process characteristic* to engineering specifications. A *process capability index* is a numerical summary of the model, also called a capability or performance index or ratio, where *capability index* is used as the generic term. A capability index relates design specification limits to a particular process characteristic. The index indicates that the process is capable of producing results that, in all likelihood, will meet or exceed the design's requirements. A capability index is convenient because

it reduces complex information about the process to a single number. Capability indices have several applications, though the use of the indices is driven mostly by monitoring requirements specified by design criteria.

Many design engineers require manufacturers to record capability indices for all the design's process characteristics on a heuristic basis. The indices are used to indicate how well the process may perform. For stable or predictable processes, it is assumed that these indices will also indicate expected future performance. Suppliers or manufacturers may use capability indices for specific system characteristics to establish priorities for improvements after installation. Similarly, the effects of process change can be assessed by comparing capability indices that are calculated before and after the change. However, despite the widespread use of capability indices in industry, and some good review articles (Gunter 1989a, b, c, d), there is much confusion and misunderstanding regarding their interpretation and appropriate use, particularly as a tool for comparing process characteristic to engineering specifications in *designing for availability*.

Process capability in quantified terms is the ratio of the deviation of a process characteristic from the specification limit, divided by a measure of a process characteristic's variability. Process capability can be represented in mathematical terms as (Steiner et al. 1995):

$$\text{Process Capability} = \min\left(\frac{USL-\mu}{3\sigma}\,\frac{\mu-LSL}{3\sigma}\right) \tag{4.17}$$

where:

USL and LSL	are the upper and lower specification limits respectively,
μ and σ	are the mean and standard deviation respectively for measures of the process characteristic of interest.

Calculating the process capability requires knowledge of the process characteristic's mean and standard deviation, μ and σ. These values are usually estimated from trial or test data collected from a pilot process. The two most widely used capability indices, P_{pk} and C_{pk}, are defined as

$$\text{Capability } P_{\text{pk}} = \min\left(\frac{USL-\bar{a}}{3\sigma}\,\frac{\bar{a}-LSL}{3\sigma}\right) \tag{4.18a}$$

where:

USL and LSL	are the upper and lower specification limits respectively,
$\bar{a}$	the overall average, is used to estimate the process mean μ and σ,
σ	is the standard deviation of the process characteristic of interest.

$$\text{Capability } C_{\text{pk}} = \min\left(\frac{USL-\bar{a}}{3\sigma_{Rd}}\,\frac{\bar{a}-LSL}{3\sigma_{Rd}}\right) \tag{4.18b}$$

where:

$\bar{a}$	the overall average, is used to estimate the process mean μ and σ,
σ_{Rd}	is the estimate of the process standard deviation σ,

d is an adjustment factor that is needed to estimate the process standard deviation from the average trial or test data sample range R.

The sample standard deviation is given by the formula

$$\sigma = \sqrt{\sum_{j=1}^{n}\sum_{i=1}^{m}(X_{ij} - \bar{a})^2/(nm-1)} \tag{4.19}$$

where:

m is the total number of subgroups, and n is the subgroup sample size.

Since d is also used in the derivation of control limits for X and R control charts, it is also tabulated in standard references on statistical process control, such as in the QS-9000 SPC manual (Montgomery 1991). Large values of P_{pk} and C_{pk} for several specific process characteristics should correspond to a process that is capable of operating within the design specification limits. The commonly used index P_p and the related index C_p in process design are similar to P_{pk} and C_{pk}. However, P_p and C_p ignore the current estimate of the process characteristic mean, and relate the specification range directly to the process characteristic variation. In effect, P_p and C_p can be considered convenient *conceptual design* measures that suggest how capable the process should be if the process characteristic's mean is centred midway between the specification limits.

The indices P_p and C_p are *not* recommended for process evaluation purposes during the *detail design* phase, since the information they provide to supplement P_{pk} and C_{pk} is independent of data. Histograms of trial or test data collected from a pilot process, usually established during the detail design phase of the engineering design process, are preferable because they also provide other useful process information. However, various important issues relating to the calculation and interpretation of capability indices require closer attention. The design capability of a process is eventually estimated from pilot trial or test data that represent a sample of the envisaged total production.

Clearly, the capability indices P_{pk} and C_{pk} are greatly influenced by the way in which the process data are collected, what is normally called the *process view*. A process view is defined by the time frame and sampling method (sampling frequency, sample size, etc.) used to obtain the pilot process data. Using an appropriate process view is crucial, since different views can lead to very different conclusions. For example, in one view the process may appear stable, while in another the process could appear unstable. To define the process view, the first choice involves the time frame over which the pilot process data will be collected. Often, the time frame is stipulated as a typical *cycle-time* interval. For example, the capability of each selected process characteristic may be measured as a function of the operating time in relation to the process cycle-time.

In other situations, the time frame is restricted to a shorter interval, such as the period needed for the pilot process to produce a specific number of production units.

To obtain a reasonable measure of the process capability, the length of the time frame should be chosen such that it is long enough to reflect all the substantial sources of variation in the process. Defining the sampling method or procedure is also important. The pilot process output should be sampled in such a way that a 'fair' representation is obtained of the process over the chosen time frame. For capability calculations, it is not always necessary for the samples to be collected in subgroups. However, sub-grouping of the pilot process data can also be used to create control charts that may be helpful in understanding a particular process characteristic.

Altering the process view can substantially change the conclusions about the process capability. As a result, specific guidelines regarding the time frame, and the sampling method used to collect the pilot process data necessary for calculating capability, are essential. Another important issue related to the process view is the number of data points used in the estimation. P_{pk} is an estimate of the process capability and, thus, even if the process is unchanged, taking another sample and recalculating the index is unlikely to yield precisely the same result. The amount of uncertainty is based on both the properties of the process and the number of data observations used to calculate the capability index. Larger sample sizes provide more information and, thus, tend to lead to better estimates of the process capability.

A further important aspect in considering process capability as an indicator in designing for availability is *process stability*. A process is considered stable if all the points on its X and R control charts fall within the design control limits, and there are no apparent deviation patterns. The stability of a process is an important property in designing for availability because, if the process design is considered stable, it is likely to also be stable in its installation and in the future, assuming that no major changes occur. Thus, the total output of a stable process is, in some sense, predictable. If the output of a process is considered stable, then the process' capability is predictable, from design to manufacture through to installation and/or construction. On the other hand, if the process output is not deemed to be stable, it might still be possible that over time the process capability index can appear to be stable, depending on the complexity of the process and/or the complex integration of the relevant process systems. The predictability of process capability can be obtained by considering the performance of the process in terms of its process capability over time. If the pilot process capability values exhibit a stable pattern, then there would be some confidence predicting the installed process capability indices, which affects the consequences of using the capability indices of P_{pk} and C_{pk}. If the process is stable, then C_{pk} is approximately equal to P_{pk}, since a stable process has little variability. Thus, if the process is stable, it does not matter much which measure is used (although P_{pk} is preferred). On the other hand, if the process is unstable, there will be substantial variability between the data subgroups, and C_{pk} is thus not equal to P_{pk}. In the case of process instability, C_{pk} will *overestimate* the process capability, since it does not include variability. The same principle applies if the process is unstable and yet predictable. As a result, in all situations, P_{pk} provides a better measure of the process capability than does C_{pk}.

Thus, in the development of intelligent computer automated methodology for determining the integrity of engineering design, particularly through the use of

a blackboard model to provide for automated continual design reviews with respect to *designing for availability*, such reviews inevitably need to include capability models of each process system. Each capability model includes a combination of declarative design rules and constraints, design criteria documentation, and process simulation results. The design rules represent constraints that apply to the design's materials and processes associated with each process capability model, such as process and functional characteristics, geometric symmetries, and expressions constraining the parameters of specific system features.

Process simulations take as input the geometric model of a design or partial design, and return estimates on the process and functional characteristics, or provide a graphical display of the characteristics and their effects. For the purpose of ensuring design for manufacturability, a set of design rules that will ensure easy manufacturability is paramount. Representations of process-derived geometric constraints provide a way of assuring manufacturability while maintaining the necessary distinction between the representation of the design and the description of the processes used to manufacture it. The neutral descriptions of these constraints also enable their use for constraint propagation in qualitative reasoning systems such as *knowledge-based expert systems*.

b) Process Characteristics

Process characteristics include the following measures:

- process capacity,
- process input,
- process throughput,
- process output,
- process quality.

Process capacity: *Capacity* can be defined as "*holding or receiving ability*".

The capacity of an engineering process normally represents a limit on the maximum holding ability of the process. In this context, *process capacity* can be defined as "*the ability of a series of operations to receive and/or hold the result or product inherent to the process*".

Process capacity has thus to do with receiving an input, and a system's ability to hold or retain an operational throughput as a result of delivering an output, and should not be confused with the specific measures of a system's *input*, *throughput* or *output*. Process capacity is the maximum amount of material or product in process. Process capacity decisions are perhaps the most fundamental of all conceptual engineering design considerations. One reason for the importance of capacity decisions relates to the impact on the ability of the process to meet future demands—capacity essentially limits the rate of possible output.

A second reason for the importance of process capacity decisions is the initial cost involved. Capacity is usually a major determinant of a design's manufacturing and installation costs. Another reason for the importance of process capacity

decisions stems from the long-term commitment of resources required in the operation of the installed design, and the fact that once installed, it may be difficult to modify the process without incurring major costs.

The term process capacity generally refers to an upper limit on the rate of *process output*. Even though this seems to be simple enough, there are difficulties in actually measuring process capacity. These difficulties arise because of different interpretations of the term *process capacity*, and problems with identifying suitable measures. Underlying these interpretations, though, is the single fact that process capacity reflects the *availability* of process resources.

There are thus three different definitions of process capacity that are applicable to availability in engineering design:

- *Design capacity* (C_d): the *maximum* ability of a series of operations to receive and/or hold the result or product inherent to the process.
- *Effective capacity* (C_e): the ability of a series of operations to receive and/or hold the result or product inherent to the process, given a specific product mix, production schedule, maintenance, and quality constraints.
- *Rated capacity* (C_r): the *throughput* actually achieved from operational constraints placed upon the ability of a series of operations to receive and/or hold the result or product inherent to the process. *Rated capacity is maximum throughput.*

Measuring process capacity Process capacity can be expressed in terms of *outputs* or *inputs*, though no single capacity measure is universally applicable. Expressing process capacity in terms of *output* measures is the usual choice for line flow processes. However, product mix becomes an issue when the output is not uniform in work content. Expressing process capacity in terms of *input* measures is normally used for flexible flow processes where process output varies in work content, and a measure of total production or units produced becomes meaningless.

Maximum process capacity can be measured in terms of the average output rate and the average utilisation rate expressed as a percentage

$$\text{Maximum Capacity } (C_{\max}) = \frac{\text{Average output rate}}{\text{Average utilisation}/100}\,. \tag{4.20}$$

Process input (I_p): Process input is the quantity or volume of process material that enters the system or equipment over a period of time in accordance with the system's *operational time*. Production input in continuous processes is the quantity or volume of process material that can enter the system or equipment according to its *process capacity*. Maximum input is the *maximum* ability to receive and/or hold the result or product inherent to the process, i.e. *design capacity*.

Process throughput (T_p): Process throughput has to do with quantities of material entering and leaving the system process over a period of processing time. With continuous processes, *throughput* is the quantity of material entering and leaving the process in a *continuous flow*. The material or product in process, *at rated capacity*, is the difference between the *input* and *output* at any specific point in time. The throughput *rate* is equivalent to the rated capacity *per unit of time*. *Process*

throughput rate is indicative of the *capability* of the process to achieve the desired result or *output*. From Little's law (Little 1961), the formula for the relation of throughput, cycle time and work in progress in any production line is given as

$$\text{Production throughput } (T_{\text{prod}}) = \frac{\text{Work in progress}}{\text{Cycle time}} . \tag{4.21}$$

In the context of *discrete* industrial processes, work in progress is synonymous to the material or product in process. Thus

$$\text{Process throughput } (T^{\text{D}}_{\text{proc}}) = \frac{\text{Material in progress}}{\text{Cycle time}} \tag{4.22}$$

where cycle time in discrete industrial processes = processing time + added time due to operational constraints and inspection.

Process throughput of a *continuous* process system can be defined as "*the ratio of a system's material in process over a period of processing time*"

$$\begin{aligned} \text{Process throughput } (T^{\text{C}}_{\text{proc}}) &= \frac{\text{Material in progress}}{\text{Processing time}} \\ &= \text{Rated capacity } (C_{\text{r}}) . \end{aligned} \tag{4.23}$$

Process output (O_{p}): *Output* can be defined as "*the quantity produced or yielded*".

Process output can be defined as "*the quantity of product, or yield of a production process*". Process output has to do with *yield* quantities of product or material from the production process. The relationship between *process throughput* and *process output* is given by the following

$$\begin{aligned} \text{Process output } (O_{\text{p}}) = {} & \text{Process throughput } (T_{\text{p}}) \\ & \times \text{Yield percentage } (Y\%) \end{aligned} \tag{4.24}$$

Utilising the previous formula for *rated capacity* as *maximum throughput*, the relationship between *output* and *yield* in accordance with a process plant's *rated capacity* gives the following

$$\begin{aligned} \text{Process output } (O_{\text{p}}) = {} & \text{Rated capacity } (C_{\text{r}}) \\ & \times \text{Yield percentage } (Y\%) \end{aligned} \tag{4.25}$$

Process or product yield Yield can be defined as "*the amount produced or the output result*".

Product yield in quality terms (without reject product) is the throughput multiplied by the percentage of successful output result (yield percentage)

$$\begin{aligned} \text{Process yield } (Y_{\text{p}}) &= \text{Process throughput } (T_{\text{p}}) \times \text{Yield percentage } (Y\%) \\ &= \text{Process output } (O_{\text{p}}) \end{aligned} \tag{4.26}$$

c) Functional Effectiveness

Functional effectiveness in engineering processes indicates the results produced. It represents functional characteristics of the process, such as process efficiency, utilisation and productivity. Availability in engineering design, particularly in production processes, is often looked upon as a functional characteristic synonymous to *productivity* in that it relates process output to input.

Process effectiveness in itself is an indication of the design's manufactured and/or installed accomplishment against the design's intended capability. Process effectiveness is a ratio of process results (i.e. actual output) to process capability (i.e. design output)

$$\text{Process Effectiveness } (W_{\text{p}}) = \frac{\text{Actual output}}{\text{Design output}} . \tag{4.27}$$

Process efficiency is the ratio of *process output* achieved through the *process throughput* (or, in certain cases, *process input*). In order to understand the concept of *efficiency* correctly, and not confuse it with the concept of *effectiveness*, it is necessary to consider these definitions with regard to related terminology.

Inasmuch as *output* is defined as the quantity produced or yielded, so can *efficiency* be defined as "*the capability of producing or yielding an output quantity*". In fact, it is this *capability of output quantity* that forms the basis of *efficiency measurement*.

Efficiency measurement is the measurement of *productive capability*. Efficiency measurement of engineering processes is thus the measure of the capability of producing or yielding a product. It is the measure of the capability of output quantity. Efficiency measurement of a process, as a ratio, must therefore include *output quantity* compared to some or other production parameter of the equipment in order to reflect its *capability of output quantity.* As this productive capability logically relates directly to *the amount that can be put through a process*, it is conclusive that the production parameter must be *process throughput*. Efficiency measurement of an engineering process is thus a comparison of the *output quantity* to its *process throughput*.

Thus

$$\begin{aligned} \text{Process efficiency } (X_{\text{p}}) &= \frac{\text{Process output}}{\text{Process throughput}} \\ &= \frac{\text{Process throughput} \times \text{Yield percentage}}{\text{Process throughput}} \\ &= \text{Yield percentage } (Y\%) \end{aligned} \tag{4.28}$$

The measure of *efficiency* must not be confused with the measure of *productivity*, which is the ratio of *output* compared to *input*. *Productivity* is the "*ratio of process output to process input*"

$$\begin{aligned} \text{Productivity } (Z) &= \frac{\text{Process output}}{\text{Process input}} \\ &= \frac{\text{Process throughput} \times \text{Yield percentage}}{\text{Process input}} \end{aligned} \tag{4.29}$$

Process utilisation Process utilisation is the ratio of process output to the *constrained* ability to receive and/or hold the result or product inherent to the process (i.e. *rated capacity*)

$$\text{Process utilisation } (U_{\text{p}}) = \frac{\text{Process output}}{\text{Rated capacity}} \tag{4.30}$$

Functional effectiveness in engineering processes represents the functional characteristics of a process, such as efficiency, productivity and utilisation. These characteristics relate process output to throughput, output to input, and output to capacity respectfully. Availability in engineering design is thus considered from the perspective of these functional characteristics, and designing for availability, particularly engineering process availability, considers measurements of process throughput, output, input and capacity.

d) Mathematical Modelling

For each process system, there is a set of performance measures that require particular attention in design. Mathematical models for expressing systems process characteristics, and functional effectiveness for both discrete and continuous process systems involve respectively summation and integration of their conjunct variables over time. These models serve as useful indicators in *designing for availability*, and adequately represent performance measures of each system that can be described in matrix form in a *parameter profile matrix* (Thompson et al. 1998):

Discrete process throughput

$$\left(T_{\text{proc}}^{\text{D}}\right) = \sum_{p=1}^{P} (M/t)p \quad 1 < p < P\,. \tag{4.31}$$

Continuous process throughput

$$\left(T_{\text{proc}}^{\text{C}}\right) = \int_{t}^{T} (M_t/t)\,\mathrm{d}t \quad 0 < t < T \tag{4.32}$$

$$\left(T_{\text{proc}}^{\text{C}}\right)_{\max} = (C_{\text{r}})$$

Process output

$$\left(O_{\text{p}}\right) = \int_{t}^{T} (M_t/t)(Y_t)\,\mathrm{d}t \tag{4.33}$$

$$\left(O_{\text{p}}\right)_{\max} = (C_{\text{r}}) \times (Y\%)$$

Process effectiveness

$$\left(W_{\mathrm{p}}\right) = \int_{t}^{T} \left(M_t/t\right)\left(Y_t/O_{\mathrm{d}}\right)\,\mathrm{d}t \tag{4.34}$$

$$\left(W_{\mathrm{p}}\right)_{\max} = \left(O_{\mathrm{p}}\right)_{\max}/O_{\mathrm{d}}$$

Process efficiency

$$\left(X_{\mathrm{p}}\right) = \int_{t}^{T} \left(M_t/t\right)\left(Y_t/T^{\mathrm{C}}_{\mathrm{proc}}\right)\,\mathrm{d}t \tag{4.35}$$

$$\left(X_{\mathrm{p}}\right) = \left(Y\%\right)$$

Productivity

$$\left(Z\right) = \int_{t}^{T} \left(M_t/t\right)\left(Y_t/I_{\mathrm{p}_t}\right)\,\mathrm{d}t \tag{4.36}$$

$$\left(Z\right) = \frac{\left(O_{\mathrm{p}}\right)}{\left(I_{\mathrm{p}}\right)}$$

Process utilisation

$$\left(U_{\mathrm{p}}\right) = \int_{t}^{T} \left(M_t/t\right)\left(Y_t/C_{\mathrm{r}}\right)\,\mathrm{d}t \tag{4.37}$$

$$\left(U_{\mathrm{p}}\right) = \frac{\left(O_{\mathrm{p}}\right)}{\left(C_{\mathrm{r}}\right)}$$

where:

M_t = material in process in time t
M_t/t = process flow rate or mass-flow rate
I_{p} = process input
Y_t = yield
O_{d} = design output
C_{d} = design capacity
C_{r} = rated capacity
O_{p_t} = process output in time t
$\left(O_{\mathrm{p}}\right)_{\max}$ = maximum process output.

In general continuous flow processes, there are certain governing equations of flow, where the *design process flow rate* or the *mass-flow rate* M_t/t (i.e. throughput, which is a pivotal parameter in the performance measures for expressing systems process characteristics) is the base measure of fundamental fluid flow. The amount of fluid

(or material) flowing through a specified cross section is referred to as the *volumetric flow rate*.

Let $W = M_t/t$ be the total mass-flow rate of fluid flowing through a specified cross section. Then

$$V = W/\rho \quad (\mathrm{m}^3/\mathrm{h}) \tag{4.38}$$

where:

$V =$ volumetric flow rate
$\rho =$ fluid density.

The average linear velocity of flow is the ratio of the volumetric flow rate to the cross-sectional flow area, as given by the following relationship

$$\hat{w} = V/F \quad (\mathrm{m}/\mathrm{h}) \tag{4.39}$$

where:

$\hat{w} =$ average flow velocity
$F =$ cross-sectional flow area.

Mass velocity can be expressed as the average velocity modified by the specific weight of the fluid, which is the fluid's specific gravity

$$G = \hat{w} \cdot \gamma \tag{4.40}$$

where:

$G =$ fluid mass velocity
$\gamma =$ fluid specific gravity.

For a continuous flow process under steady-state conditions, the mass-flow rate M_t/t, or W, must be the same at any section within the process. This is the principle of *mass-flow balance*

$$W_1 = W_2 = W_3 = \text{etc.} \tag{4.41}$$

The mass-flow balance is a *statement of continuity*, which can also be written as

$$F_1 G_1 = F_2 G_2 = F_3 G_3 = \text{etc.} \tag{4.42}$$

where:

$F =$ cross-sectional flow area
$G =$ fluid mass velocity

and:

$\rho =$ fluid density
$\gamma =$ fluid specific gravity
$\rho/\gamma =$ constant.

Without going into the depths of fluid mechanics and hydraulics, which is not relevant to the objectives of this handbook, the nature of general flow regimes needs to be considered in order to address not only the principle of *mass-flow balance* in continuous flow processes but their *total energy balance* as well, so that these measures can be used in determining system performance characteristics that may serve as useful indicators in designing for availability *without having to formulate the specific operational variables of each individual system*. This is best done through *simulation*, which is considered more closely in the next section on analytic development.

There are fundamentally three general flow regimes in continuous flow processes: *laminar flow*, *transition flow* and *turbulent flow*.

The laminar flow regime occurs at relatively low fluid velocities, providing a smooth flow pattern with no or very little mixing of the fluid particles. Transition flow denotes the onset of turbulence. In a turbulent flow regime, fluid velocities are higher, and an unstable pattern within the mass flow is observed in which eddy current forces move at all angles to the axis of normal flow.

The *dependency* of a particular flow regime is denoted by the dimensionless Reynolds number whereby a critical Reynolds number indicates the *transition from one flow regime to another*. For instance, if the Reynolds number for flow in a straight circular pipe is less than 2,100, the flow is laminar. When the Reynolds number exceeds 4,000, the flow is turbulent. Flow between these two critical numbers is transitional.

The mathematical model for the Reynolds number is given by the following relationships

$$Re = W \cdot D/\nu = \hat{w} \cdot D \cdot \rho/\mu = W \cdot D/\mu \tag{4.43}$$

where:

$Re =$ Reynolds number
$W =$ mass-flow rate
$D =$ system or tube (pipe) diameter
$\nu =$ *kinematic* viscosity
$\nu = \mu/\rho$
$\hat{w} =$ average flow velocity
$\rho =$ fluid density
$\mu =$ *dynamic* viscosity.

Specific mathematical models for volumetric flow rates, V, and average flow velocities, $\hat{w}$, for laminar flows in a variety of systems are available in determining the Reynolds number.

In considering the *total energy balance*, the flow energy input of a continuous flow process is the sum of the kinetic energy, E_k, the potential energy, E_p, the volumetric energy, E_v, and the internal energy, E_i. Any disruption in one or another of these energies in the *total energy balance* is an indication of degradation in the *performance* or *operability* of the process and, thus, these are important criteria in its engineering design.

The *availability* of the process or system is concerned with expected *system performance* over a period of expected *operational time*. The prediction of *inherent availability* of systems is based upon a prognosis of *systems performance* and *systems operability* under conditions subject to various *performance criteria*, such as mass-flow balance and total energy balance.

Inclusive of any heat input from heat exchangers, or mechanical work derived from pumping, the total energy balance of a continuous process flow consists of the four energies E_k, E_p, E_v and E_i, whereby the *total energy balance* can be formulated as follows

$$E_{k_1} + E_{p_1} + E_{v_1} + E_{i_1} = E_{k_2} + E_{p_2} + E_{v_2} + E_{i_2} \tag{4.44}$$

The *kinetic energy*, E_k, is a function of the fluid mass and the fluid's linear velocity:

$$E_{k_1} = \hat{w}_1^2/2g\alpha$$
$$E_{k_2} = \hat{w}_2^2/2g\alpha$$

where:

$\alpha =$ correction coefficient and, for turbulent flow, $\alpha = 1$.

The *potential energy*, E_p, is a function of the weight, Z, of the fluid:

$$E_{p_1} = Z_1$$
$$E_{p_2} = Z_2$$

The *volumetric energy*, E_v, under pressure P, is equivalent to the energy required to hold volume v at that pressure:

$$E_{v_1} = P_1 v_1$$
$$E_{v_2} = P_2 v_2$$

The internal energy, E_i, is a thermodynamic property of the flow system, with reference state energies, E_1, E_2, which on the input side is a function of heat input from heat exchangers, H_e, and mechanical work from pumping, M_e, approximated by the enthalpies i_1 and i_2:

$$E_{i_1} = \text{state } E_1 = H_e + M_e$$
$$E_{i_2} = \text{state } E_2$$
$$i_1 = E_1 + P_1 v_1$$
$$i_2 = E_2 + P_2 v_2$$

The *total energy balance* can now be formulated as follows (Cheremisinoff 1984):

$$\hat{w}_1^2/2g\alpha + Z_1 + P_1 v_1 + H_e + M_e = \hat{w}_2^2/2g\alpha + Z_2 + P_2 v_2 + E_2 \tag{4.45}$$
$$\hat{w}_1^2/2g\alpha + Z_1 + H_e + M_e = \hat{w}_2^2/2g\alpha + Z_2 + (i_2 - i_1)$$

e) Sizing Maximum or Design Capacity

The *effective capacities* of multiple system operations or processes within the same engineering design installation are usually different. A *bottleneck* is a process that has the lowest *effective capacity* of any process in the designed installation and, thus, limits total output. Expansion of maximum or design capacity occurs only when bottleneck capacity is increased. However, flexible flow processes may have *floating bottlenecks* due to widely varying workloads on different processes at different times.

The *theory of constraints (TOC)* in *designing for availability* focuses on design alternatives that impede maximum capacity (i.e. bottlenecks), with the objective of maximising total product or materials process flow (Goldratt 1990). Also, the focus on bottlenecks is the means to increasing *throughput* and, consequently, the mass-flow rate of product and materials. The performance of the overall process design is a function of minimum bottleneck operations or processes. *TOC* provides the ability to descriptively characterise the functional relationships responsible for a typical complex process environment. Basically, through the application of *system dynamics (SD)* models, which are developed from *TOC* logic diagrams, insights into the dynamics of design alternatives that impede maximum capacity are obtained. The application of *TOC* in designing for availability involves the following steps:

- Identification of system bottleneck(s).
- Exploitation of the bottleneck(s)
  (i.e. maximising *throughput*).
- Elevating the bottleneck(s)
  (i.e. considering increasing capacity at the bottleneck(s)).

Criteria for sizing design capacity Besides increasing the capacity of system bottlenecks in order to expand design capacity, further criteria for *sizing design capacity* are concerned with predicted process utilisation rates that are close to 100%, indicating the need to increase capacity because of the probability of declining productivity over time (i.e. diminished output against constant input). Process utilisation tends to be higher in capital-intensive processes, where prediction of utilisation between 90 and 100% is not uncommon. In such cases, occurrences of bottlenecks in the total process are inevitable, resulting in the essential application of *TOC* in designing for availability.

A further consideration is *economy of scale*. In designing for availability, this implies not only increasing a design's size or capacity but at the same time attempting to decrease the average unit cost through various options, such as:

- Spreading fixed costs:
  As the system utilisation rate increases, the average unit cost is reduced.
- Reducing manufacturing/construction costs:
  Doubling facility size usually does not double costs.
- Reducing material costs:
  Higher volumes allow for bulk acquisition and handling.

- Exploiting process advantages:
  High volume may justify investment in more efficient technology.
- Increasing inherent availabilities:
  Determining initial system operational characteristics.

In contrast to consideration of *economy of scale* is the need also to consider *diseconomies of scale*, whereby excessive size can bring about complexity and inefficiencies that, in turn, can raise the average unit cost, and result in a non-linear growth of overhead.

4.2.1.3 Inherent Availability (A_i) Modelling with Uncertainty

Under initial conditions of uncertainty, it is feasible to define *system availability* only in terms of operable time and *corrective maintenance*. Availability defined in this manner is termed *inherent availability* (A_i). Under such idealised conditions, standby and delay times associated with scheduled or *preventive maintenance*, as well as administrative and logistics downtime are ignored. Inherent availability is thus useful in determining initial system operational characteristics under specified conditions, such as testing in a contractor's facility, or any other controlled test environment. Likewise, inherent availability becomes a useful term to describe *combined* reliability and maintainability characteristics or to define the one in terms of the other during the early conceptual phase of the engineering design process when, generally, these terms cannot be defined individually and are rather related to system performance.

Since such a definition of availability is easily measured, it is frequently used as a contract-specified requirement. Inherent availability is primarily the concern of the design engineer during the establishment of functional interface with the contractor and manufacturers in the early phases of the engineering design process. Inherent availability looks at availability from a design perspective; thus, reliability and maintainability are considered complementary measures in the inherent availability equation. Inherent availability is in effect a model of reliability and maintainability measures. The inherent availability equation is given as (Eq. 4.46), (DoD 3235.1-H 1982):

$$A_i = \frac{\text{MTBF}}{(\text{MTBF} + \text{MTTR})} \tag{4.46}$$

where:

MTBF is the mean time between failure
MTTR is the mean time to repair.

A_i is the largest availability value that can be achieved because only the times related to operational disruptions due to breakdowns and their repair are considered, whereas downtime associated with planned maintenance as well as administrative and logistics downtime are ignored.

If the expected design reliability measure of mean time between failures (or, more particularly, mean time to breakdowns) is very large compared to the related

mean time to repair (or mean time to replace), then the inherent availability is high. Similarly, if the design maintainability measure of mean time to repair (MTTR) is a minimum, the inherent availability A_i will be a maximum.

It is obvious from the inherent availability equation that if design reliability decreases (i.e. MTBF becomes smaller), then better design maintainability (i.e. shorter MTTR) is needed to achieve the same inherent availability. Conversely, as engineering design reliability increases, design maintainability is not so important in being able to achieve the same inherent availability.

An important design integrity principle is thus obtained:

Trade-offs can be made between reliability and maintainability to achieve the same availability in the engineering design process.

a) The Exponential Function for Inherent Availability

If λ is designated the *failure rate* (1/MTBF) and μ is designated the *repair rate* (1/MTTR), and both rates are exponential, then the *probability density function* (p.d.f.) of a failure at time x is

$$f(x) = \lambda \, \mathrm{e}^{-\lambda x} \, . \tag{4.47}$$

The probability density function that a subsequent repair will be completed at time t, the end of the availability cycle, $t > x$, is

$$f(t-x) = \mu \, \mathrm{e}^{-\mu(t-x)} \, . \tag{4.48}$$

The availability cycle can be construed to have two consecutive periods; the first period is when operation is terminated by a failure, and the second period is when downtime ends with a completed repair. Inherent availability is the ratio of the *average* time for the first period, to the average time for the cycle, which includes operation and downtime. The probability density function of a failure before t, followed by a repair completed at t, is the convolution (accumulated product) of Eqs. (4.47) and (4.48)

$$f(t) = \int_0^t f(x) f(t-x) \, \mathrm{d}t \tag{4.49}$$

$$f(t) = \frac{\lambda \mu \, \mathrm{e}^{-\mu t}}{\mu - \lambda} \left(\mathrm{e}^{-\lambda t} - \mathrm{e}^{-\mu t} \right) \quad \text{with } \mu > \lambda \, .$$

The average period of an *availability cycle* $E(t)$ is

$$E(t) = \int_0^t t f(t) \, \mathrm{d}t \tag{4.50}$$

$$E(t) = \frac{\lambda + \mu}{\lambda \mu} \, .$$

The average period of an *availability cycle* $E(t)$ is expressed in terms of mean time between failure (MTBF) and mean time to repair (MTTR):

$$E(t) = \frac{1}{\lambda} + \frac{1}{\mu}$$
$$E(t) = \text{MTBF} + \text{MTTR}$$

Thus, *inherent availability* A_i is the fraction of the *availability cycle*

$$A_\text{i} = \frac{\text{MTBF}}{(\text{MTBF} + \text{MTTR})} = \frac{1/\lambda}{1/\lambda + 1/\mu} = \frac{\mu}{\lambda + \mu} \tag{4.51}$$

b) Confidence Determination of Inherent Availability Predictions

Equation (4.51) indicates that if both the MTBF and MTTR distributions are exponential, then the inherent availability A_i is a function of the failure rate λ and the repair rate μ. Since both λ and μ can readily be used for Bayesian prior and posterior analysis, random values can be generated in repeated trials in order to simulate a value for A_i. The percentage values of the resulting distribution on A_i are the confidence limits of the inherent availability prediction.

In predicting the value of A_i, the ratio of the mean operating period (MTBF) to that of the availability cycle (MTBF + MTTR) can be established by known or estimated distributions for these values. However, establishing confidence levels on different values of A_i (i.e. quantitative assessment of A_i) can be done only by using known failure and repair data to establish distributions on MTBF and MTTR parameters. For example, if both the time between failures and time to repair are exponential, then the values for MTBF and MTTR can be determined from Bayesian prior distributions, which are functions of the prior data. Beyond such relatively simple analysis, establishing confidence levels on different values of A_i is very difficult.

Thus, *predictions* of A_i are feasible under initial conditions of uncertainty, as with conceptual design, if it is possible to define system availability with respect to estimates of *operable time* and *downtime* due to corrective maintenance. Standby and delay times associated with scheduled or preventive maintenance, as well as administrative and logistics downtime are ignored. A major problem arises, though, when these estimates *cannot be based on obtained data*, and predicting the value of A_i cannot be *quantitative*. However, as indicated in Sect. 3.3.3.3 on reliability evaluation, a statistically acceptable *qualitative* methodology to determine the integrity of engineering design in the situation where data are not available or not meaningful is included in the concept of *information integration technology (IIT)*. The concept of *IIT* includes a combination of methods and tools for collecting, organising and analysing diverse information, and for utilising that information to guide optimal decision-making, based on Bayesian prior and posterior analysis (Booker et al. 2000).

4.2.1.4 Preliminary Maintainability Modelling

Probability theory and statistics have an important role in designing for maintainability, as much as they have in engineering design integrity methodology as a whole. Various probability distributions may be used to quantify repair time data, and even uncertainty of repair times. Where repair time data are not available, including any data representing failure rates or expected time to failure, *qualitative* methods involving *possibility theory* need to be used, similar to the prediction of reliability considered in the previous section. However, in the case of data being available, even censored data, repair time distributions may be identified and the corresponding maintainability function may be obtained. The maintainability function is used to predict the probability that a repair, beginning at time $t = 0$, will be accomplished in a time t. The *maintainability function* $M(t)$, for any distribution, is expressed by the following relationship (Dhillon 1999b):

$$M(t) = \int_0^t f_{\mathrm{r}}(t)\,\mathrm{d}t \tag{4.52}$$

$f_{\mathrm{r}}(t)$ is the probability density function of the maintenance (repair) time.

This maintainability function may be represented by various distribution functions, depending upon the statistical characteristics of the data and the function parameters. The *exponential distribution* is particularly useful in presenting *maintenance times that are random* in duration.

The exponential distribution probability density function is defined by the following relationship

$$f_{\mathrm{r}}(t) = (1/\mathrm{MTTR})\,\mathrm{e}^{-(t/\mathrm{MTTR})} \tag{4.53}$$

where:

t is the variable repair time, and MTTR is the mean time to repair.

By substituting Eq. (4.53) into Eq. (4.52), the following relationship is obtained

$$M(t) = \int_0^t (1/\mathrm{MTTR})\,\mathrm{e}^{-(t/\mathrm{MTTR})}\,\mathrm{d}t \tag{4.54}$$

$$M(t) = 1 - \mathrm{e}^{-(t/\mathrm{MTTR})}$$

$$M(t) = 1 - \mathrm{e}^{-\mu_t}\ .$$

The fundamental parameter is the repair rate, μ, the reciprocal of MTTR, rather than the failure rate, λ, the reciprocal of MTBF. The treatment of 'time to an event' is also reversed, in that the objective should be to make μ as high as possible, so that repairs are completed quickly, and to make λ as low as possible, so that the time between failures as long as possible.

In the maintainability relationship given in Eq. (4.54), let t denote a specified or required 'standard' time to repair. Since t is specified, it is necessary only to evaluate μ. Furthermore, suppose that available data consist of estimates of repair times $t_1, t_2, \ldots, t_{\mathrm{r}}$. The total estimated time, T, on repair status is then

$$T = \sum_{i=1}^{\mathrm{r}} t_i \,. \tag{4.55}$$

Because the repair events are all independent, the joint probability, or likelihood L, of the first r repair times, $t_1, t_2, \ldots, t_r$ is the product of their respective probabilities

$$L = \prod_{i=1}^{r} f_r(t) \,. \tag{4.56}$$

From Eq. (4.53) we get

$$L = \mu \exp\left[-\mu \left(\sum_{i=1}^{r} t_i\right)\right] \,. \tag{4.57}$$

The *maximum-likelihood estimate*, E, is a value μ that maximises the natural logarithm of L

$$\begin{aligned} E &= \ln L \\ E &= r \ln \mu - \mu T \\ \frac{\partial E}{\partial \mu} &= \frac{r}{\mu} - T \,. \end{aligned} \tag{4.58}$$

Setting the derivative to zero, the *maximum-likelihood estimate* of μ is

$$\mu' = \frac{r}{T} \,. \tag{4.59}$$

The *best estimate* $m'(t)$ of the maintainability function, $M(t)$, with standard maintenance time t, is then obtained where $m'(t) = M$, in the case of $0 \leq M < 1$, may be viewed as having a Bayesian prior or posterior distribution with parameters that are valid statistics for r repair actions and T total repair time (Eq. (4.60)). If these estimates *cannot be based on obtained data*, the methodology of *information integration technology (IIT)* is applicable, in which Bayesian prior and posterior analysis is utilised.

$$\begin{aligned} M(t) &= 1 - \mathrm{e}^{-\mu_t} \\ m'(t) &= 1 - \mathrm{e}^{-\mu_t'} = 1 - \mathrm{e}^{-r_t/T} = M \,. \end{aligned} \tag{4.60}$$

4.2.2 Theoretical Overview of Availability and Maintainability Assessment in Preliminary Design

Availability and maintainability assessment attempts to estimate the expected *usage* of equipment over a period of *operational time subject to both planned and unplanned maintenance downtime* or, alternatively, the expected *utilisation* over a specified period of each individual item of equipment at the upper systems levels of the systems breakdown structure. System availability is an important *measure* of *repairable systems*, since it considers both reliability and maintainability, whereas availability and maintainability modelling takes into account both the failure and repair *states* of a system. More specifically, availability and maintainability *assessment* takes into account not only the failure and repair states of a system but downtime due to *preventive maintenance* as well. Availability and maintainability assessment in this context is considered during the *preliminary* or *schematic design* phase of the engineering design process. The most applicable methodology for availability and maintainability assessment in the preliminary design phase includes basic concepts of mathematical modelling such as:

i. Markov modelling for design availability and maintainability
ii. Achieved availability modelling subject to maintenance
iii. Maintainability assessment with maintenance modelling
iv. Maintenance strategy and cost optimisation modelling.

4.2.2.1 Markov Modelling for Design Availability and Maintainability

Markov modelling is a powerful engineering design analysis tool, and it can be used in most cases of *designing for reliability* and *designing for maintainability*. The method is useful in modelling systems, especially large complex systems, with *dependent failure and repair modes*. Markov models are particularly useful to model repairable systems with random failure occurrences (i.e. constant or time-independent *failure rates*) and random repair times (i.e. constant or time-independent *repair rates*). The method becomes unreliable for systems with time-dependent failure and repair rates.

a) The Two-State Markov Model

Several initial assumptions are important when applying Markov modelling to engineering design analysis (Dhillon 1999b):

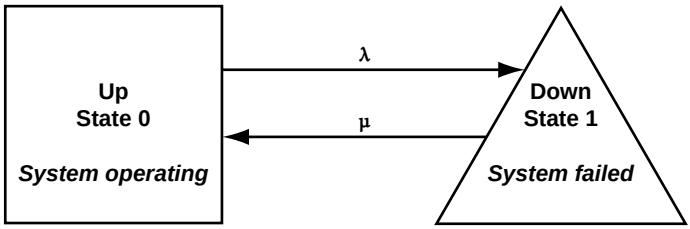


Fig. 4.7 Markov model state space diagram

- All events are independent of each other.
- The probability of transition from the system *operating state* to the system *failed state* (state 0 to state 1) is given by $\lambda\Delta t$, where Δt is a finite time interval, and λ is the constant failure rate, or the transition rate.
- The probability of transition from the system *failed state* to the system *operating state* (state 1 to state 0) is given by $\mu\Delta t$, where Δt is a finite time interval, and μ is the constant repair rate, or the transition rate.
- The probability of more than one transition from one state to another in Δt is very small.

The transition states can be represented in the following diagram (Fig. 4.7).

From Fig. 4.7, the following mathematical model can be derived (Dhillon 1999b):

$$P_0(t+\Delta t) = P_0(t)(1-\lambda_{\mathrm{f}}t) + P_1(t)\mu_{\mathrm{r}}t \tag{4.61}$$

and

$$P_1(t+\Delta t) = P_1(t)(1-\mu_{\mathrm{r}}t) + P_0(t)\lambda_{\mathrm{f}}t\ . \tag{4.62}$$

Status variables and probabilities The various status variables and probabilities of these two equations need to be evaluated:

λ_{f}	is the system constant failure rate,
μ_{r}	is the system constant repair rate,
$P_0(t+\Delta t)$	is the probability that the system is in an operating state 0 at the time $t+\Delta t$,
$P_1(t+\Delta t)$	is the probability that the system is in a failed state 1 at the time $t+\Delta t$,
$P_0(t)$	is the probability that the system is in an operating state 0 at time t,
$P_1(t)$	is the probability that the system is in failed state 1 at time t,
$(1-\lambda_{\mathrm{f}}t)$	is the probability of no failure in time interval t when the system is in state 0,
$(1-\mu_{\mathrm{r}}t)$	is the probability of no repair in time interval t when the system is in state 1,
$\lambda_{\mathrm{f}}t$	is the probability of system failure in time interval t,
$\mu_{\mathrm{r}}t$	is the probability of accomplishing system repair in time interval t.

In the limiting case, Eqs. (4.61) and (4.62) are represented by

$$\lim_{\Delta t\to 0}\frac{P_0(t+\Delta t)-P_0(t)}{\Delta t}=\frac{\mathrm{d}P_0(t)}{\mathrm{d}t}=P_1(t)\mu_\mathrm{r}-P_0(t)\lambda_\mathrm{f} \tag{4.63}$$

$$\lim_{\Delta t\to 0}\frac{P_1(t+\Delta t)-P_0(t)}{\Delta t}=\frac{\mathrm{d}P_1(t)}{\mathrm{d}t}=P_0(t)\lambda_\mathrm{f}-P_1(t)\mu_\mathrm{r} \tag{4.64}$$

In order to solve Eqs. (4.63) and (4.64) at time $t=0$, the values for the following probabilities are: $P_0(0)=0$, and $P_1(0)=0$.

Then

$$P_0(t)=\frac{\mu_\mathrm{r}}{\lambda_\mathrm{f}+\mu_\mathrm{r}}+\frac{\lambda_\mathrm{f}}{\lambda_\mathrm{f}+\mu_\mathrm{r}}\,\mathrm{e}^{-(\lambda_\mathrm{f}+\mu_\mathrm{r})t} \tag{4.65}$$

and

$$P_1(t)=\frac{\lambda_\mathrm{f}}{\lambda_\mathrm{f}+\mu_\mathrm{r}}+\frac{\lambda_\mathrm{f}}{\lambda_\mathrm{f}+\mu_\mathrm{r}}\,\mathrm{e}^{-(\lambda_\mathrm{f}+\mu_\mathrm{r})t}\,. \tag{4.66}$$

Thus, at any point in time t, the system's availability may be represented by the following

$$A(t)=P_0(t) \tag{4.67}$$

and

$$P_0(t)=\frac{\mu_\mathrm{r}}{\lambda_\mathrm{f}+\mu_\mathrm{r}}+\frac{\lambda_\mathrm{f}}{\lambda_\mathrm{f}+\mu_\mathrm{r}}\,\mathrm{e}^{-(\lambda_\mathrm{f}+\mu_\mathrm{r})t}$$

where:

$A(t)=$ the system's availability at a specified time t.

For engineering design *availability assessment*, estimate of availability for the system would be a *steady-state availability*, A_s, where $t\to\infty$. Thus

$$A_\mathrm{s}=\lim_{t\to\infty}A(t) \tag{4.68}$$

and A_s is A(steady state).

Substituting Eq. (4.67) into Eq. (4.68) gives the steady-state availability for the system.

Thus, $A_\mathrm{s}=A$(steady state) is given by

$$A_\mathrm{s}=\lim_{t\to\infty}\left[\mu_\mathrm{r}/\lambda_\mathrm{f}+\mu_\mathrm{r}+\lambda_\mathrm{f}/\lambda_\mathrm{f}+\mu_\mathrm{r}(\mathrm{e}^{-(\lambda_\mathrm{f}+\mu_\mathrm{r})}t)\right] \tag{4.69}$$

$$A_\mathrm{s}=\frac{\mu_\mathrm{r}}{\lambda_\mathrm{f}+\mu_\mathrm{r}}\,.$$

b) Multi-State Markov Models—Method of Supplementary Variables

The components of most systems are assumed to fail with constant failure rates (i.e. failure times are governed by exponential distributions). However, though *repair times* of components are often non-exponentially distributed, they usually have

general distributions (i.e. repair rates of the components are arbitrary functions of time). Multi-component repairable systems with general failure and/or repair time distributions are difficult to analyse mathematically. These systems are known as non-Markovian systems, as the stochastic process is non-Markovian. However, with the inclusion of the method of *supplementary variables*, the Markov process approach provides a sufficient level of analysis that can be used to model complex systems with constant failure rates and non-exponential repair times. Inclusion of sufficient supplementary variables in the specification of the state of the system can make a process Markovian (Dhillon 1983).

To enable the system to be characterised as a Markov system, a mathematical model is constructed with concise definitions of the various states for the system, together with a set of supplementary variables that include the concept of efficiency (or, rather, reduced efficiency) in the state definition of the system. Because the state at time t is an exact description of the circumstances prevailing in the system at that time, the behaviour of the system over the passage of time Δt may be found by determining the state probabilities of the system. A complex system can thus be characterised as a Markov system by employing a set of supplementary variables with which a part of the system's history is included in the state definition of the system. With the inclusion of *supplementary variables*, the Markov model represents a multi-state stochastic system with modes of normal operation and total failure, as well as operation at several different levels of performance (i.e. with reduced efficiency).

The system has thus three operation modes: 'normal operation', 'operation with reduced efficiency' and 'non-operation'. The supplementary variable technique enables a dynamic model of the behaviour of the system to be set up in the form of a set of differential-difference equations with variable coefficients, and respective boundary and initial conditions (Virtanen 1977).

As an illustration of the method of supplementary variables, consider the system transition diagram in Fig. 4.8 (Dhillon 1983).

The diagram represents a complex system that operates partially when some of system's components fail and, if a catastrophic failure occurs, the system in its entirety fails. When the system is operating partially, a repair process is expected to be initiated to restore the system to its fully operational state. However, the system may have a catastrophic failure from the partially operating state. Once the system fails completely, it is expected to be restored to its normal operating state.

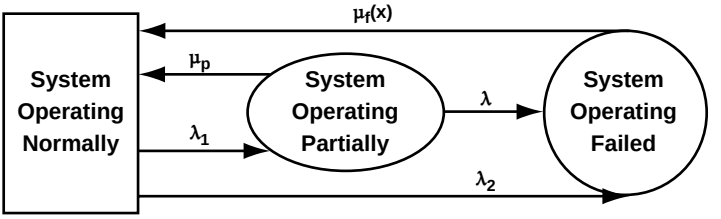


Fig. 4.8 Multi-state system transition

The following assumptions are associated with this multi-state model:

- System failures are statistically independent.
- A partially, or fully failed system is restored to a 'good as new' state.
- System failure rates are constant.
- System component failure times are random.
- The partially failed system repair rate is constant.
- Failed system repair times are arbitrarily distributed.

As with the two-state Markov model, the mathematical expressions for the multi-state Markov model, including supplementary variables indicating partial operation or a reduced efficiency of the system, are given in the following Markov multi-state model equations, according to Fig. 4.8:

$$P_0(t+\Delta t) = P_0(t)(1-\lambda_1\Delta t)(1-\lambda_2\Delta t) + P_1(t)\mu_p\Delta t + \left[\int_0^\infty P_2(x,t)\mu_f(x)\,dx\right]\Delta t \tag{4.70}$$

$$P_1(t+\Delta t) = P_1(t)(1-\lambda_3\Delta t)(1-\mu_p\Delta t) + P_0(t)\lambda_1\Delta t \tag{4.71}$$

$$P_2(x+\Delta t;t+\Delta t) = P_2(x,t)\,[1-\mu_f(x)\Delta t] \tag{4.72}$$

λ_j is the jth constant failure rate of the system with $j=1$ (normal–partial transition), $j=2$ (normal to failed), $j=3$ (partial to failed),

μ_p is the system constant repair rate from the partial operating state 1 to the normal operating state 0,

$\mu_f(x)$ is the repair rate when the system is in the failed state and has the elapsed repair time of x,

$P_0(t+\Delta t)$ is the probability that the system is in an operating state 0 at time $t+\Delta t$,

$P_1(t+\Delta t)$ is the probability that the system is in a partially failed state 1 at time $t+\Delta t$,

$P_2(x+\Delta t;t+\Delta t)$ is the probability that at time t, the system is in a failed state 2 and the elapsed repair time lies in the interval $(x,x+\Delta x)$,

$P_0(t)$ is the probability that the system is in an operating state 0 at time t,

$P_1(t)$ is the probability that the system is in a partially failed state 1 at time t,

$P_2(x,t)$ is the probability that the system is in a failed state 2 after an elapsed repair time of x,

$(1-\lambda_i\Delta t)$ is the probability of no failure in time interval Δt when the system is in state i,

$(1-\mu_p\Delta t)$ is the probability of no repair in time interval Δt when the system is in state 1,

$(1-\mu_f\Delta t)$ is the probability of no repair in time interval Δt when the system is in state 2.

The respective boundary and initial conditions are:

$$P_2(0,t) = \lambda_2 P_0(t) + \lambda_3 P_1(t)$$

and at $t = 0$

$$P_0(0) = 1$$
$$P_2(0) = 0$$
$$P_2(x,0) = 0$$

The differential-difference equations with variable coefficients are

$$\frac{\mathrm{d}P_0(t)}{\mathrm{d}t} + (\lambda_1 + \lambda_2)P_0(t) - P_1(t)\mu_\mathrm{p} = \int_0^\infty P_2(x,t)\mu_\mathrm{f}(x)\,\mathrm{d}x \tag{4.73}$$

$$\frac{\mathrm{d}P_1(t)}{\mathrm{d}t} + (\lambda_3 + \mu_\mathrm{p})P_1(t) - P_0(t)\lambda_1 = 0 \tag{4.74}$$

$$\frac{\partial P_2(x,t)}{\partial x} + \frac{\partial P_2(x,t)}{\partial t} + \mu_\mathrm{f}(x)P_2(x,t) = 0 \tag{4.75}$$

So far, the supplementary variable technique has been used to obtain the model's partial differential-difference equations, or state equations, which describe the behaviour of the system. With the help of Laplace transforms, both transient and steady-state solutions for these state equations may now be found. The Laplace transform of a function is given by the expression

$$E(t) = \int_0^\infty \mathrm{e}^{-St} f(t)\,\mathrm{d}t\ . \tag{4.76}$$

Using Laplace transforms, and initial condition $P_0(0) = 1$, the differential Eqs. (4.73) to (4.75) are transformed into steady-state solutions for these state equations, with the boundary condition of:

$$P_2(0,s) = \lambda_2 P_0(s) + \lambda_3 P_1(s)$$

Then

$$sP_0(s) - 1 + (\lambda_1 + \lambda_2)P_0(s) - P_1(s)\mu_\mathrm{p} = \int_0^\infty P_2(x,s)\mu_\mathrm{f}(x)\,\mathrm{d}x \tag{4.77}$$

and

$$sP_1(s) + (\lambda_3 + \mu_\mathrm{p})P_1(s) - P_0(s)\lambda_1 = 0 \tag{4.78}$$

and

$$\frac{\partial P_2(x,s)}{\partial x} + [s + \mu_\mathrm{f}(x)]P_2(x,s) = 0\ . \tag{4.79}$$

The steady-state values for $P_0(s)$, $P_1(s)$ and $P_2(s)$ can now be found through integrating. The steady-state solutions are independent of the type of waiting time and repair time distributions, and only the *expected values* of these distributions become

apparent. Furthermore, steady state is achieved under general conditions, and the solutions for steady state can be found without any exact knowledge about the distributions of the system (Virtanen 1975).

4.2.2.2 Achieved Availability Modelling Subject to Maintenance

Achieved availability ($A_{\acute{s}}$) is frequently used during development testing and initial production testing when a system or its equipment is not operating in its intended support environment. Excluded are operator before-and-after maintenance checks and standby periods. Achieved availability is much more of a system hardware-oriented measure than is *operational availability*, which considers operating environment factors.

It is, however, dependent on a *preventive maintenance policy*, which can be greatly influenced by non-hardware considerations. The mathematical model for *achieved availability*, according to the USA Department of Defence, is given by the following expression (Eq. 4.80), (Conlon et al. 1982):

$$A_{\acute{s}} = \frac{\mathrm{OT}}{\mathrm{OT} + \mathrm{TCM} + \mathrm{TPM}} \tag{4.80}$$

where:

OT = operating time
TCM = total corrective maintenance
TPM = total preventive maintenance.

An alternative approach to modelling achieved availability is to consider the probability that a system or its equipment, when used under designed conditions in an ideal support environment, will perform according to the specifications formulated during the *preliminary design* phase. The most significant characteristic of achieved availability for both alternatives is that it includes maintenance time (corrective and preventive), and excludes logistic delay times. The mathematical model for achieved availability in this context is given as (Dhillon 1999b):

$$A_{\acute{s}} = \frac{\mathrm{MTBM}}{\mathrm{MTBM} + \mathrm{TCM} + \mathrm{TPM}} \tag{4.81}$$

where:

MTBM is the mean time between maintenance.

This differs from inherent availability, A_i, only in its inclusion of the consideration for *total preventive maintenance*. The measurement base for MTBM must be consistent when calculating achieved availability $A_{\acute{s}}$. MTBM is represented by the

following expression

$$\text{MTBM} = \left[\frac{1}{\text{MTBF}} + \frac{1}{\text{MTBM-LD}} + \frac{1}{\text{MTBPM}}\right] \tag{4.82}$$

where:

MTBF	is the mean time between failures
MTBM-LD	is the mean time between maintenance less logistic delays
MTBPM	is the mean time between preventive maintenance.

The measurement base for MTBF, MTBM–LD and MTBPM must be consistent when calculating the MTBM parameter. Consider further the values TCM and TPM

$$\text{MDT} = \text{TCM} + \text{TPM} \tag{4.83}$$

where:

MDT	=	mean active maintenance downtime
TCM	=	total corrective maintenance
TPM	=	total preventive maintenance

and

$$\text{MDT} = \frac{\sum_{i=1}^{m} CM_i CF_i}{\sum_{i=1}^{m} CF_i} + \frac{\sum_{j=1}^{n} PM_j PF_j}{\sum_{j=1}^{n} PF_j} \tag{4.84}$$

where:

n	=	total corrective tasks performed
m	=	total preventive tasks performed
CM_i	=	elapsed time for corrective task i
PM_j	=	elapsed time for preventive task j
CF_i	=	estimated frequency for task i
PF_j	=	estimated frequency for task j.

4.2.2.3 Maintainability Assessment with Maintenance Modelling

Maintainability and maintenance are closely interrelated, yet they are not the same. Maintainability refers to the measures taken during the design, development and installation of a system or its equipment that will reduce the required maintenance effort, logistics and costs and, thus, also the operational downtime. Maintenance refers to the measures taken to restore and keep the system or its equipment in an operable condition. Maintenance is, in effect, the care of the physical and operational condition of the system or its equipment. Many mathematical models have been developed for both maintainability and for maintenance.

However, maintenance models have mainly been developed to better define and predict certain aspects of maintenance, such as scheduled downtime, scheduled replacement, and optimal warranty periods, for installed systems and equipment.

These models are usually based on certain probability distributions, predominantly the exponential distribution for representing corrective maintenance times, and the lognormal distribution for representing minimum operating times.

a) Impact of Maintenance Assessment on Systems Design

A widely used probability distribution in predicting the impact of designing for maintainability on systems design, based upon defining constraints on the minimum operating time below which no maintenance activity will result in downtime, is the lognormal distribution.

The lognormal distribution probability density function is defined by the following relationship

$$f_{\mathrm{r}}(t) = \frac{1}{(t-\theta)\sigma\sqrt{2\pi}}\,\mathrm{e}^{-\{1/2[\ln(t-\theta)-\beta]\}} \tag{4.85}$$

where:

t = maintenance time
θ = minimum operating time
β = mean time for maintenance
σ = standard deviation of the maintenance times.

An estimate of the *mean time for maintenance*, β, is based on an estimate of the number of shutdowns (i.e. planned downtimes that have an impact on production) that are required over a specific period, such as one year. This is best approached from a calculation of the average of the sum of the natural logarithms of the individually estimated downtimes, where m is the number of shutdowns over a specific period.

The relationship for the *mean time for maintenance*, β, considering the estimated downtimes and the number of shutdowns, is defined as

$$\beta = (\ln t_1 + \ln t_2 + \ln t_3 + \ldots + \ln t_m)/m\,. \tag{4.86}$$

The standard deviation, σ, of the estimated *mean time for maintenance*, β, is given by

$$\sigma = \left[\sum_{i=1}^{m} (\ln t_i - \beta)^2/(m-1)\right]^{1/2}. \tag{4.87}$$

For the lognormal distribution, the equation for the maintainability function $M(t)$ is given as the following expression

$$M(t) = \int_0^{\infty} t\, f_{\mathrm{r}}(t)\,\mathrm{d}t \tag{4.88}$$

$$M(t) = 1/\sigma\sqrt{2\pi}\int_0^{\infty} \mathrm{e}^{-1/2(\ln t-\beta)^2}\,\mathrm{d}t$$

This maintainability function serves primarily as a design parameter in *designing for maintainability*, whereby it defines the expected downtime over a specified period.

The measures used in maintainability analysis, besides the widely used mean time to repair (MTTR), include concepts related mainly to *maintenance*, such as the *expected mean preventive maintenance downtime*, the *median corrective maintenance downtime*, the *expected maximum corrective maintenance downtime* and the *expected mean maintenance downtime*.

b) Maintainability Measures and Maintenance Assessment

The *expected mean preventive maintenance downtime*, T_{pm}, is a useful parameter in design for maintainability, in that it gives an indication of the expected scheduled downtime of a system over its life cycle. The objective of defining the expected mean preventive maintenance downtime is to estimate the impact of a *preventive maintenance program* on the system, whereby the system and its equipment (assemblies and components) are to be kept at a specified design performance level. Such a preventive maintenance program is to affect the point in time at which the equipment wears out or fails, resulting in system downtime.

A carefully planned *preventive maintenance program* can help to reduce system downtime and improve its performance. On the other hand, a poorly established preventive maintenance program can have a negative impact on system operations. The expected mean preventive maintenance downtime, T_{pm}, is expressed by the mathematical model (Dhillon 1999b):

$$T_{\mathrm{pm}} = \frac{\sum_{i=1}^{k}(T_{\mathrm{p}ti})(F_{\mathrm{p}ti})}{\sum_{i=1}^{k}(F_{\mathrm{p}ti})} \tag{4.89}$$

where:

$T_{\mathrm{p}ti} =$ the estimated lapse time for preventive maintenance task i for $i = 1,2,3,\ldots,k$
$F_{\mathrm{p}ti} =$ the estimated frequency of preventive maintenance task i for $i = 1,2,3,\ldots,k$
$k \;\; =$ number of preventive maintenance tasks.

The *median corrective maintenance downtime*, $\tilde{T}_{\mathrm{cm}}$, is a measure of the time within which 50% of all corrective maintenance can be completed. Calculation of the median corrective maintenance downtime depends upon the distribution of the times for corrective maintenance.

For a *lognormal distribution* of repair time, the median corrective maintenance downtime, $\tilde{T}_{\mathrm{cm}}$, is expressed as

$$\tilde{T}_{\mathrm{cm}} = \mathrm{MTTR}/\,\mathrm{e}^{\sigma^2/2} \tag{4.90}$$

$\sigma^2 =$ the variance around the mean value of the natural logarithm of repair times.

For an *exponential distribution* of corrective maintenance repair times, the median corrective maintenance downtime, T_{cm}, is expressed as

$$T_{\mathrm{cm}} = 0.69/\mu \tag{4.91}$$

$\mu =$ the repair rate, which is the reciprocal of MTTR.

The *expected maximum corrective maintenance downtime* T_{cm} is a measure of the time required to complete corrective maintenance repairs at the 90th or 95th percentiles. This implies that, for example, in the case of the 95th percentile, the expected maximum corrective maintenance downtime is the time within which 95% of all corrective maintenance can be completed. It indicates an estimation level of significance where no more than 5% of the expected corrective maintenance will take longer than the expected maximum corrective maintenance downtime. Calculation of the expected maximum corrective maintenance downtime also depends upon the distribution of the times for corrective maintenance.

The expected maximum corrective maintenance downtime with a *lognormal distribution* of corrective maintenance times is expressed as

$$T_{\mathrm{cm}} = \mathrm{antilog}(t_{\mathrm{m}} + k\sigma) \tag{4.92}$$

where:

$t_{\mathrm{m}} =$ the mean of the logarithms of repair times
$k =$ the value 1.28 or 1.65 for the 90th or 95th percentiles
$\sigma =$ the standard deviation of the logarithms of repair times.

The expected maximum corrective maintenance downtime with an *exponential distribution* of corrective maintenance times is expressed as

$$T_{\mathrm{cm}} = 3 \times (\mathrm{MTTR}) \tag{4.93}$$

where:

MTTR = the mean time to repair, given by the following formula.

$$\mathrm{MTTR} = \frac{\sum_{i=1}^{m} \lambda_i T_i}{\sum_{i=1}^{m} \lambda_i} \tag{4.94}$$

$\lambda_i =$ the constant failure rate of item $i = 1, 2, 3, \ldots, m$
$T_i =$ the corrective maintenance or repair time needed to restore item $i = 1, 2, 3, \ldots, m$.

The *expected mean maintenance downtime, MDT*, is the total time needed to restore the system or its equipment to a specified level of performance, and to maintain it at that level of performance. It includes preventive and corrective maintenance times but *not* administrative and logistic delay times. In this regard, it is synonymous with *achieved availability* that includes maintenance time (corrective and preventive) but excludes administrative and logistic delays. After determining T_{pm} and T_{cm}, the

expected mean maintenance downtime, MDT, is given by the following relationship

$$\text{MDT} = T_{\text{pm}} + T_{\text{cm}} \tag{4.95}$$

Substituting Eqs. (4.89) and (4.93) with (4.94) into Eq. (4.95) gives

$$\text{MDT} = \frac{\sum_{i=1}^{k}(T_{\text{p}ti})(F_{\text{p}ti})}{\sum_{i=1}^{k}(F_{\text{p}ti})} + \frac{3\sum_{i=1}^{m}\lambda_i T_i}{\sum_{i=1}^{m}\lambda_i} \tag{4.96}$$

where:

$T_{\text{p}ti} =$ the estimated lapse time for preventive maintenance task i for $i = 1, 2, 3, \ldots, k$
$F_{\text{p}ti} =$ the estimated frequency of preventive maintenance task i for $i = 1, 2, 3, \ldots, k$.

To determine the *expected mean total downtime*, DT, estimates of delays (administration and logistic) need to be added to MDT. These delays are usually estimated as fractions of MDT.

4.2.2.4 Maintenance Strategies and Cost Optimisation Modelling

So far, the interrelationships of maintainability and maintenance have been considered with respect to measures used in maintainability analysis that include maintenance concepts, such as *preventive maintenance*, *corrective maintenance* and *downtime*. In designing for maintainability, it is important to understand the concepts of *maintenance strategies*.

In designing for maintainability, the up-front establishment of cost-effective *maintenance strategies* has a significant impact on the final outcome of the engineering design, particularly in considering built-in-testing (BIT), online fault diagnostics, and the application of condition monitoring. A proper understanding of the basic principles of maintenance thus becomes extremely important (in fact, it becomes essential) in the engineering design process, and includes not only maintenance and production people but design engineers as well. Once the basic principles of maintenance are fully understood, then the more sophisticated and complex aspects essential to cost-effective maintenance strategies can be considered. These aspects include an understanding of condition monitoring, condition measurement, fault diagnostics and predictive maintenance, and how and when they should be carried out in order to effectively care for the physical and operational condition of the system or its equipment.

Designing for maintainability is not only a consideration of the measures taken during the design, development and installation of a system that will reduce the required maintenance effort and, thus, also the operational downtime, as well as logistics and costs, but it is also a provision of the required maintenance strategies that complement these measures in order to ensure the as-designed system performance and related warranty. All these aspects thus need to be carefully considered and placed in their correct perspective for establishing cost-effective maintenance strategies in designing for maintainability.

a) The Basic Principles of Maintenance

Maintenance can be defined as "*the continuous action of caring for the condition of equipment*". By definition, the concept of *condition* has been brought into the understanding of maintenance. Equipment condition is the *operational* and *physical* state of equipment on which the functions of the equipment depends.

In order to understand equipment condition, it thus becomes necessary to understand the concept of equipment *function*. The function of equipment is the *work* and *properties* that the equipment is designed to perform and to have. There are two basic types of equipment functions:

- Operational function
- Physical function.

The *operational functions* can be grouped into primary and secondary functions. The primary operational function of equipment is described by defining what *work* the equipment primarily does. The secondary operational functions of equipment are the other *activities* that the equipment also does. As an example, the primary operational function of a heat exchanger would be to transmit heat through conduction from a hot fluid to a cooler fluid, thereby decreasing the temperature of the hot fluid, and increasing the temperature of the cooler fluid. A secondary function of a heat exchanger is to reduce the occurrence of flash vapour in the liquid line (sometimes called flash gas, arising from a sudden change of the fluid to a vapour).

The *physical functions* of equipment are described by defining the *design configuration* and *physical properties* of the equipment. Referring to the previous example, the most significant physical function of a heat exchanger is the ability to provide efficient heat transfer at high temperature through a heat transfer surface that is large enough to transfer the heat sufficiently, and that is also able to resist expansion stresses that may cause cracks and dangerous leakages.

Thus, the *condition* of equipment as described in the definition of maintenance can now be reviewed. It can be seen that the condition of equipment is directly related to the equipment's functions. There are two types of equipment conditions, related to the functions of the equipment and called the *functional states of condition*. The two types of equipment conditions are:

- Operational condition
- Physical condition.

The *operational condition* of equipment relates to its operational functions, and the *physical condition* of equipment relates to its physical functions.

Maintenance can now be redefined as "*the continuous action of caring for the operational and physical conditions of equipment*".

The next concept to consider in this definition of maintenance is the "*continuous action*". There are predominantly two *actions* in maintenance:

- Corrective action
- Preventive action.

Corrective action, by definition, is "*that action necessary to rectify or set right defects according to a standard*". Corrective action is thus that maintenance work that fixes or repairs equipment after it has failed. *Preventive action*, by definition, is "*that action serving to hinder or stop defects*". Preventive action is thus that maintenance work that prevents or stops defects from occurring in equipment *before it has failed*.

By progressive definition, the concept of *failure* has been brought into the understanding of maintenance action. Thus, in order to fully understand maintenance, it is essential to understand the concept of failure. Equipment failure has already been defined as "*the inability of the equipment to function within its specified limits of performance*". There are thus two descriptions of failure:

- Functional failure
- Potential failure.

Functional failure in equipment is "*the inability of the equipment to carry out the work that it was designed to perform within specified limits of performance*". This inability has qualitative gradation, depending upon the severity of functional failure. There are two degrees of severity in functional failure:

- A complete or *total loss of function*, where the equipment cannot carry out any work that it was designed to perform.
- A *partial loss of function*, where the item is unable to function within specified limits of performance.

Potential failure in equipment is "*the identifiable condition of the equipment, indicating that functional failure can be expected*". Potential failure is a condition or state of condition of the equipment. Functional failure is an occurrence or incident.

The definition of *preventive action* in maintenance can now be reviewed. From the point of view of the two descriptions of failure, preventive action in maintenance is "*that action serving to hinder or to stop functional or potential failures*". Thus, preventive action in maintenance is that action serving to hinder or stop the occurrences of defects in the function of equipment through the detection of an identifiable condition arising in the equipment, indicating that it is unable to carry out the work that it was designed to perform within specified limits of performance.

Maintenance can thus be comprehensively defined as "*the continuous corrective and preventive action of caring for the operational and physical conditions of equipment*".

The different types of maintenance In order to convert the definition of maintenance into practice, it is necessary to define how corrective and preventive action in maintenance is implemented. These actions in maintenance are practically implemented through different *types* of maintenance.

There are three basic types of maintenance:

- Defect maintenance.
- Routine maintenance.
- Preventive maintenance.

Defect maintenance is the *corrective action* in maintenance through fixing or repairing equipment after it has failed.

Routine maintenance is the *preventive action* in maintenance that cares for the *operational condition* of equipment through inspection, adjustment, recording, monitoring, servicing and lubrication, to ensure that the equipment's *operational functions* conform to the required limits of performance.

These routine maintenance activities can be grouped into the following categories that can be *scheduled on a fixed-time interval*:

- Running checks:
  This includes inspections and minor adjustments.
- Monitoring checks:
  This includes data log records and condition monitoring readings.
- Service checks:
  This includes replacement of lubricants and consumable parts.

The concept of routine maintenance is based upon the type of preventive actions that can be routinely carried out or, by definition, performed according to a regular course of procedure on a fixed-time interval basis. Evidently, this type of preventive action can only be directed towards the *operational condition* of equipment.

Preventive maintenance is the *preventive action* in maintenance that strives to reduce the likelihood of failure through the detection of identifiable potential failures in the equipment's *physical condition*, and thus attempts to avoid *functional failure* occurrences. This is done through scheduled checks and inspections of physical condition, fault diagnostics, measurement, scheduled shutdowns for opening and cleaning equipment, scheduled shutdowns for replacing worn components, and scheduled shutdowns for overhauling plant and equipment.

These preventive maintenance activities can be grouped into the following categories that are *scheduled on run-time intervals*:

- Physical checks:
  This includes scheduled checks of physical conditions, and fault diagnostics.
- Measurement checks:
  This includes measurement of physical conditions such as stress cracks, thickness tests, wear tolerances, etc., and scheduled shutdowns for opening and cleaning equipment.
- Replacement shuts:
  This includes scheduled shutdowns for replacement of worn components, and scheduled shutdowns for overhauling plant and equipment.

Thus, the way in which corrective and preventive action in maintenance is practically implemented is through the different *types* of maintenance whereby:

- Defect maintenance is corrective action in restoring equipment to its operational state or repairing physical defects *after it has failed*.
- Routine maintenance is preventive action in caring for the *operational condition* of the equipment *before it has failed*.

- Preventive maintenance is preventive action in caring for the *physical condition* of the equipment *before it has failed*.

Condition monitoring and the concept of predictive maintenance One of the routine maintenance activities included in the category of monitoring checks is *condition monitoring*. Condition monitoring is the assessment of the condition of equipment *whilst it is in operation*.

Consequently, condition monitoring can be regarded as a *routine maintenance* task that cares for the operational condition of equipment. Thus, from an understanding of the condition of equipment, condition monitoring can be properly defined as "*the assessment of the operational condition of equipment whilst it is in operation*".

There are two types of condition monitoring:

- Periodic monitoring
- Continuous monitoring.

Periodic monitoring is the monitoring of equipment *operational condition* according to a regular course of procedure on a periodic fixed-time interval basis. The simplest form of periodic condition monitoring is operational checks of equipment temperatures, vibration or noise by the operator or service technician. The more sophisticated form of periodic condition monitoring is the use of specialised instrumentation to monitor temperature (thermographics, infra-red scanning, etc.), vibration (accellerometers, etc.), noise (ultrasonics) and contamination (lubricant and debris analysis, etc.). *Continuous monitoring* is monitoring of equipment *operational condition* through the employment of electronic signal processing techniques to determine certain equipment operational characteristics (such as vibration of rotating machinery, pump flow, etc.), with the aid of online sensors, onboard or mounted instrumentation (geriometry), and computerisation (supervisory control and data acquisition, SCADA, systems).

The importance of condition monitoring, compared to walk-through inspections, is the essential *trending of accumulated monitoring data*. Through forecasting the trend of an increasing divergence of the operational condition of equipment away from its standard limits of operational performance, predictions can be made concerning gradual degradation of the physical condition of the equipment. This forecasting of diverging trends of the operational performance of equipment and predicting failure is called *predictive maintenance*. The term is not quite correct, as maintenance by definition implies *action*, and the forecasting of operational conditions of equipment to predict failure is not a specific action or work carried out on the equipment itself. The only action that is carried out in condition monitoring is the taking of readings of equipment operational condition—which is a *routine maintenance activity*. The *result* of the prediction of failure can lead to the *action* of scheduled replacement or equipment overhaul—which is a *preventive maintenance activity*.

Condition monitoring, including forecasting trends in the deviation of operational conditions and, thus, predicting the possibility of failure in the physical condition of equipment, forms the *link between routine maintenance and preventive*

maintenance. Condition monitoring is the 'stepping stone' from caring for equipment *operational condition* (routine maintenance), to caring for equipment *physical condition* (preventive maintenance). Thus, condition monitoring is the routine maintenance assessment of the operational condition of equipment that will give an indication for the need for preventive maintenance action.

Condition measurement and the concept of fault diagnostics There are many benefits that can be derived from the ability to anticipate the need for preventive maintenance. The occurrence of failure that results from degradation of the physical condition of equipment takes place in a sequence or cascade of events with each event increasing the probability of a partial loss of function or total loss of function of the equipment. If the rate of deterioration of the physical condition of equipment can be measured before a total loss of function occurs, then preventive maintenance can be systematically planned to avoid such an occurrence of failure. Such a measurement of the rate of deterioration of the physical condition of equipment is called *condition screening*, and incorporates the use of *condition measurement*.

It is essential, in designing for maintainability, to have an understanding of the *patterns* of functional failure of equipment with a physical condition that is deteriorating. Only then can preventive maintenance be carried out. Most engineered installations have scheduled shutdowns for either production or process changes, physical condition inspections, or for general overhauls, in which the opportunity arises for the physical condition of critical components to be examined and tested by *non-destructive test (NDT)* methods of condition measurement.

Condition inspection is the most basic examination of an equipment's physical condition, and can be enhanced by the use of condition measurement methods for the detection and fault diagnostics of cracks, surface wear or defects, deformation, corrosion, thickness reduction, and stress marks due to aged equipment or excessive use. Fault diagnostics is the analysis of the deterioration of the physical condition of equipment to determine the causes and effects of wear, cracks, defects, deformation, corrosion and stress in the equipment.

b) Mathematical Model of Preventive Maintenance Physical Checks

Although condition inspection is the most basic examination of physical condition, it is often disruptive to the continued operation of equipment. However, it usually decreases downtime due to preventive maintenance because it results in fewer breakdowns. Typical mathematical models for calculating the optimum number of physical condition inspections, with resulting minimum preventive maintenance downtime, are of the following format (Dhillon 1999b):

$$T_{\mathrm{pm}} = IT_{\mathrm{id}} + \frac{kT_{\mathrm{bd}}}{I} \tag{4.97}$$

where:

$T_{\mathrm{pm}} =$ preventive maintenance downtime

I = number of physical condition inspections
k = operational constant for a particular system
T_{id} = downtime per physical condition inspection
T_{bd} = downtime due to equipment breakdown.

Taking derivatives of Eq. (4.97) with respect to I gives

$$\frac{\mathrm{d}T_{\mathrm{pm}}}{\mathrm{d}I} = T_{\mathrm{id}} + \frac{kT_{\mathrm{bd}}}{I^2} \,. \tag{4.98}$$

Setting Eq. (4.98) to zero for optimisation, and rearranging the variables:

$$\bar{I} = [kT_{\mathrm{bd}}/T_{\mathrm{id}}]^{1/2}$$

where:

$\bar{I}$ = optimum number of physical condition inspections.

Substituting Eq. (4.98) into Eq. (4.97) yields the optimum downtime due to physical condition inspections that contribute to the preventive maintenance downtime

$$T_{\mathrm{pm}} = 2[kT_{\mathrm{bd}}T_{\mathrm{id}}]^{1/2} \tag{4.99}$$

c) Mathematical Model of Preventive Maintenance Replacement Shuts

Similar to the previous model, the objective of this model is to minimise preventive maintenance downtime as a result of scheduled shutdowns for replacement of worn components. The model represents a constant interval replacement policy. Such a constant interval replacement model implies the following (Elsayed 1996):

- Replacements are carried out at predetermined intervals, irrespective of the age condition of the equipment's components.
- Replacements are made of failed equipment (i.e. unit replacement and repair cycle).

Preventive maintenance downtime, T_{pm}, can be expressed in the form of system downtime (inclusive of the system's equipment) over the length of the preventive maintenance cycle (Jardine 1973):

$$T_{\mathrm{pm}} = \frac{\mathrm{SDT}}{\mathrm{CL}} \tag{4.100}$$

where:

T_{pm} = preventive maintenance downtime
SDT = system and system's equipment downtime
CL = length of the preventive maintenance cycle

and

$$\mathrm{SDT} = T_{\mathrm{pr}} + T_{\mathrm{bd}} \tag{4.101}$$

$$\mathrm{CL} = T_{\mathrm{pr}} + T_C \tag{4.102}$$

where:

T_{pr} = downtime due to equipment replacement
T_{bd} = downtime due to system equipment breakdown
T_C = uptime time interval between replacements.

Preventive maintenance downtime, T_{pm}, over the length of the preventive maintenance cycle can thus be expressed as the comparison of downtime due to equipment replacement plus the downtime due to system equipment breakdowns, to the downtime due to equipment replacement plus the uptime interval between replacements (i.e. the preventive maintenance cycle).

For several replacement tasks over the length of the preventive maintenance cycle, CL, the variables can be expressed as the following

$$T_{\mathrm{pr}} = \sum_{i=1}^{k} (T_{\mathrm{p}ti})(F_{\mathrm{p}ti}) \tag{4.103}$$

$$T_{\mathrm{bd}} = \sum_{i=1}^{m} \lambda_i T_i \tag{4.104}$$

where:

$T_{\mathrm{p}ti}$ = the estimated lapse time for preventive maintenance replacement task i for $i = 1, 2, 3, \ldots, k$
$F_{\mathrm{p}ti}$ = the estimated frequency of preventive maintenance replacement task i for $i = 1, 2, 3, \ldots, k$
λ_i = the constant failure rate of item $i = 1, 2, 3, \ldots, m$
T_i = the corrective maintenance time needed to replace item $i = 1, 2, 3, \ldots, m$.

Inserting Eqs. (4.101) to (4.104) into Eq. (4.100) yields an expression for T_{pm} that can then be optimised in terms of the uptime interval between replacements, T_C, by taking derivatives of Eq. (4.105) with respect to T_C and setting it to zero

$$\begin{aligned} T_{\mathrm{pm}} &= (T_{\mathrm{pr}} + T_{\mathrm{bd}})/(T_{\mathrm{pr}} + T_C) \\ &= \frac{\sum_{i=1}^{k} (T_{\mathrm{p}ti})(F_{\mathrm{p}ti}) + \sum_{i=1}^{m} \lambda_i T_i}{\sum_{i=1}^{k} (T_{\mathrm{p}ti})(F_{\mathrm{p}ti}) + T_C} \end{aligned} \tag{4.105}$$

d) Maintenance Strategy

The term *strategy* is defined as "*an overall plan with a choice of activities to be effectively carried-out*". Maintenance strategy is closely related to the definition of maintenance as well as to the concept of *effective maintenance*.

To be able to understand the concept of *effective maintenance*, it is necessary to first examine the principles underlying the *goal* of maintenance. The *goal* of maintenance is defined as "*that maintenance action necessary to achieve the correct balance between the costs of input resources and the benefits derived from the performance of effective maintenance action*".

Two principles can be discerned from this definition of the goal of maintenance. The first principle is the *correct balance* between the costs of maintenance resources and the benefits of maintenance. This balance can be represented in the form of a ratio:

$$\text{Balance} = \frac{\text{Benefits of maintenance}}{\text{Costs of maintenance}}$$

This ratio can also be rewritten as:

$$\text{Balance} = \frac{\text{Output of maintenance}}{\text{Input of maintenance}} = \frac{\text{Output}}{\text{Input}}$$

This ratio is known as the *productivity ratio*, or the *cost efficiency ratio*. It is the ratio of the amount of maintenance work performed (output) to the total cost expended (input).

Maintenance action is often measured in terms of manpower utilisation and resource costs. This is a measure of *efficient* maintenance. However, the definition of the goal of maintenance describes the correct balance of input to output, derived from the performance of *effective* maintenance action.

The question to be asked then is 'what is *effective* maintenance, and what is the difference between efficient maintenance and effective maintenance?'

Efficient maintenance in simple terms can be described as '*doing the job right*', and effective maintenance in simple terms can be described as '*doing the right job*'.

The second principle in properly understanding the goal of maintenance is that it is not so much a determination of the *amount* of work that is to be carried out that is crucial but, rather, whether the maintenance work that needs to be done is the *right type* of maintenance that will be done at the *right time*. This is effective maintenance.

The definition of maintenance strategy From an understanding of the definitions of maintenance and the goal of maintenance, equipment *maintenance strategy* can be defined as "*the continuous corrective or preventive action for the care of equipment operational and physical condition on which the equipment's functions depend to achieve the necessary technical benefits through the application of defect maintenance, routine maintenance, and preventive maintenance, in an overall plan*". In other words, a maintenance strategy is carrying out the right types of maintenance (scope of work) at the right time (overall plan). A maintenance strategy implies effective maintenance.

An overall maintenance plan, with a choice of the essential types of maintenance activities to be carried out, takes into account the following *design criteria*:

- The operation of the system and its output demand.
- The functions and criticality of the equipment.
- The required level of maintenance service.

The operation of the system and its output demand are variables that relate to process efficiency, utilisation and productivity, all of which represent the functional characteristics of the process. The functions and criticality of the equipment are determined from FMEA and FMECA. The level of maintenance service is based on the required operational and physical conditions of the equipment, as well as on the amount of planning that is required for each type of maintenance to achieve these conditions.

A maintenance strategy, not only in designing for maintainability but in the general context of process engineering design, outlines the best way to develop the most suitable scope of maintenance work or service to be conducted on the proposed engineered installation, within an overall maintenance plan. This is established through taking cognisance of the following:

- What type of maintenance must be done.
- Why each type of maintenance must be done.
- Where each type of maintenance must be done.
- How each type of maintenance must be done.
- When each type of maintenance must be done.
- What technical expertise is required for the work.
- How frequently each type of maintenance must be done.

This maintenance service is developed according to a strategy that includes all or some of the following concepts that need to be adopted for each item of designed equipment. The selection of and/or combination of these concepts will inevitably impact upon the necessary decisions in designing for maintainability:

- Run-to-failure (defect maintenance).
- Fixed-time-interval (routine maintenance).
- Run-time-interval (preventive maintenance).

In simple terms, then, a maintenance strategy is concerned with matching the *best combination* of the various types of maintenance to particular equipment according to the following criteria:

- The operation of the plant and output demand.
- The functions and criticality of the equipment.
- The required operational and physical conditions of the equipment.
- The amount of planning required for each type of maintenance.
- The frequency of each type of maintenance.
- The necessary technical benefits to be achieved.

It is thus the *balanced combination* of the application of the different *types of maintenance* that constitutes a maintenance strategy. However, a question that can justifiably be asked at this point is ‘why is it necessary to have a maintenance strategy?’—it is essential to develop a maintenance strategy for process equipment, particularly

during the engineering design stage, so that the necessary *technical benefits* can be achieved according to the designed *measures of performance*.

Measures of performance Returning to the definition of the *goal* of maintenance as the maintenance action necessary to achieve the correct balance between the costs of input resources and the *benefits* derived from the performance of an effective maintenance action, it is evident that there are specific benefits to be achieved from effective maintenance. As indicated, these benefits are predominantly *technical benefits* and can be achieved through developing a maintenance strategy, particularly during the engineering design stage. However, not all technical benefits are derived from the performance of an effective maintenance action, as the design criteria of *maintainability* refers to measures taken during the design stage *that strive to reduce* the required maintenance action, repair skill levels, logistic costs or support facilities.

The *technical benefits* relating to the engineering design that can be derived from the performance of effective maintenance are the following:

- Properly maintained operational conditions.
- Properly maintained physical conditions.
- Corrective action being carried out on time.
- Preventive action being carried out on time.
- Achieving the designed equipment reliability.
- Achieving the designed equipment availability.
- Achieving the designed equipment maintainability.
- Achieving the required operational safety.

An important question at this point is 'how would one know whether a developed maintenance strategy for a particular engineering design will, in fact, achieve the necessary technical benefits?' The effectiveness of a maintenance strategy developed during the engineering design stage can be determined only through the measures of performance of the benefits that are achieved in the completed engineered installation. These measures of performance are the measures of operational equipment reliability, availability, maintainability and safety (i.e. operational integrity) that need to be compared to the original design benchmark measures. It is evident that the only means of determining whether a maintenance strategy is effective is to establish measures of *design integrity* as a benchmark against which measures of *operational integrity* can be compared. It thus becomes a comparison of engineering design intention against engineering design application during the equipment life cycle, from design through to restoration, rather than single points of measure in the equipment's life.

Maintenance strategy can now be defined as "*the continuous action of caring for equipment condition through a balanced application of preventive maintenance, routine maintenance and defect maintenance, to achieve benefits of reliability, availability, maintainability and safety*".

Up till now, the terms reliability, availability and maintainability have been used as measures of design integrity and operational integrity. It is, however, necessary to define these terms in the context of the basic principles of maintenance—

particularly as performance measures of the results that can be achieved from the application of the different types of maintenance in a maintenance strategy, and to understand which results are achieved from the application of which type of maintenance.

e) Concept of Equipment Reliability in Maintenance Strategy

Reliability of equipment has been defined as "*the probability that equipment will perform a required function, under specific conditions, for a required period of time*". Operational reliability is the probability that equipment will not fail in a given period of operation.

The fundamental indicator of reliability was previously given as the probability that the equipment has operated over a specific period of time, the average of which is the measure of MTBF (mean time between failures). What is significant in the concept of equipment reliability within a maintenance strategy framework is that the *physical condition* of equipment is determined by the MTBF, which is a measure of its reliability. Reliability is thus the most useful performance measure for determining the result of the physical condition of equipment. Furthermore, it was previously stated that *preventive maintenance* is that type of maintenance that cares for the *physical condition* of equipment.

Thus, the physical condition of equipment is maintained through preventive maintenance, and its effect is determined by MTBF, which is the performance measure of the equipment's reliability. The performance measure of reliability determines the physical condition of equipment and the effectiveness of the preventive maintenance being carried out to care for its physical condition. The inherent reliability of equipment is initially established by its physical design and by its quality of manufacture. Design for reliability thus plays an important role in the initial reliability of equipment, the lack of which is often the cause of failures resulting in downtime stoppages.

f) Concept of Equipment Availability in Maintenance Strategy

The availability of equipment has been defined as "*that period of time in which the equipment is in a usable condition*". Availability is the equipment's capability of being used. The measure of operational availability is the relationship of the equipment's *potential usage* over a period of time, where usage is defined as "*the period of time that equipment is being utilized*". Potential usage of equipment is the sum of its actual utilisation and the period of time that the equipment was capable of being used but was not.

The effect of potential usage is determined by the performance measure of the equipment's availability. What is significant in the concept of equipment availability within a maintenance strategy framework is that the *operational condition* of equipment is determined by its potential usage, which is a measure of its availability. Availability is thus the most useful performance measure for determining the result

of the operational condition of equipment. Furthermore, it was previously stated that *routine maintenance* is that type of maintenance that cares for the *operational condition* of equipment. Thus, the operational condition of equipment is maintained through routine maintenance, and its effect is determined by the equipment's potential usage over a period of time, which is the performance measure of the equipment's availability. The performance measure of availability determines the operational condition of equipment and the effectiveness of the routine maintenance being carried out to care for its operational condition. In designing for availability, the inherent availability of equipment is its potential usage in design operational time.

g) Concept of Equipment Maintainability in Maintenance Strategy

The maintainability of equipment has been defined as "*the probability that equipment which has failed can be restored to its required condition within a given period of time*". Operational maintainability is the probability that failed equipment is repaired within a given period of time. The fundamental indicator of maintainability was previously given as the probability of repair within a given period of time, the average of which is MTTR (mean time to repair). What is significant in the concept of equipment maintainability within a maintenance strategy framework is that the *ability* to repair failed equipment within a given period of time is determined by the MTTR, which is a measure of its maintainability. Maintainability is therefore the most useful performance measure for determining the *repairable condition* of equipment. *Defect maintenance* is that maintenance work that fixes or repairs equipment after it has failed. Thus, failed equipment is restored through defect maintenance, and its effect is determined by MTTR, which is the performance measure of the equipment's maintainability. The performance measure of maintainability determines the repairable condition of equipment and the effectiveness of defect maintenance being carried out to restore the equipment to its repaired state within a given period of time. Maintainability is primarily a design parameter, and designing for maintainability defines how long equipment is expected to be down after failure.

h) The Three Principles of a Maintenance Strategy

There are three fundamental principles of a maintenance strategy:

- The effectiveness of preventive maintenance is determined by the technical benefit of reliability, which is the performance measure of the physical condition of equipment.
- The effectiveness of routine maintenance is determined by the technical benefit of availability, which is the performance measure of the operational condition of equipment.
- The effectiveness of defect maintenance is determined by the maintainability of equipment, which is the performance measure of the repairable condition of equipment.

i) Establishing Maintenance Strategies for Engineering Design

From the three fundamental principles of a maintenance strategy, it is evident that all required maintenance work is made up of one or more types of maintenance that accomplish specific technical benefits. As stated previously, it is the *combination* of these different types of maintenance that constitutes a maintenance strategy.

From an engineering design perspective, a maintenance strategy is the establishment of the most effective combination of the different types of maintenance to be carried out on specific equipment in order to achieve the most desired technical benefit from that equipment. This is determined through designing for reliability, availability, maintainability and safety (i.e. designing for engineering integrity—where in this case, the concept of safety is considered as part of designing for reliability). On the other hand, the most effective combination of the different types of maintenance for completed engineered installations (i.e. a maintenance strategy for *operational* systems and equipment) is established through a RAMS (reliability, availability, maintainability and safety) program (DoD 5000.2-R. 1997). The deliverable results are the establishment of operations and maintenance procedures and work instructions in which the different types of maintenance are effectively combined into maintenance strategies for specific equipment. The established maintenance strategies for the effective care of the condition of engineering equipment are taken up in a *RAMS program*.

The RAMS program The *goal* of the RAMS program is to establish policies and strategies for effective care of the condition of engineering systems and equipment through the implementation of various RAMS methods and techniques. The *objectives* of the RAMS program are to:

- Ensure effective care of equipment condition.
- Optimise the technical benefits derived from equipment reliability, availability, maintainability and safety.
- Establish priorities for achieving targeted quality and safety.
- Establish maintenance strategies for carrying out the most applicable and effective types of maintenance and use of appropriate maintenance procedures and work instructions.
- Ensure a correct balance of costs against desired technical benefits.

The immediate *benefits* of the RAMS program are in the establishment of maintenance policies and strategies through an analysis and understanding of the following:

- The systems process, equipment functions, failure modes, failure effects, failure causes and failure consequences, and the criticality of equipment failures resulting in safety hazards, downtime, and consequential damage,
- Identifying equipment conditions and failure characteristics and establishing effective maintenance through the correct combination of the different types of maintenance by prioritising the related technical benefits to be achieved,

- Avoiding consequential damage and establishing the necessary maintenance procedures, work instructions and logistic support for equipment care and product quality,
- Comparing *design* integrity as a benchmark against measures of *operational* integrity.

The benefits achieved through the establishment of maintenance policies and strategies can be summarised in three fundamental principles of a RAMS program, each relating targeted results and design requirements (in sequential order) of safety, reliability, availability and maintainability to the desired technical benefits, performance measures, consequential effects on the designed equipment, and the required types of maintenance.

Principles of a RAMS program in maintenance strategy The *first RAM principle* in a maintenance strategy is the following logical sequence:

Targeted result:

SAFETY

|

Technical benefit:

RELIABILITY

|

Performance measure:

MTBF

|

Effect on equipment:

PHYSICAL CONDITION

|

Type of maintenance:

PREVENTIVE MAINTENANCE

The *second RAM principle* in a maintenance strategy is the following logical sequence:

Targeted result:

UTILISATION

|

Technical benefit:

AVAILABILITY

|

Performance measure:

POTENTIAL USAGE

|

Effect on equipment:

OPERATIONAL CONDITION

|

Type of maintenance:

ROUTINE MAINTENANCE

The *third RAM principle* in a maintenance strategy is the following logical sequence:

Targeted result:

QUALITY

|

Technical benefit:

MAINTAINABILITY

|

Performance measure:

MTTR

|

Effect on equipment:

REPAIRABLE CONDITION

|

Type of maintenance:

DEFECT MAINTENANCE

j) Maintenance Cost Optimisation Modelling

Returning to the definition of the *goal* of maintenance as "*that maintenance action necessary to achieve the correct balance between the costs of input resources and the benefits derived from the performance of effective maintenance action*", an additional principle in the understanding of the goal of maintenance, and of maintenance as a whole, is the concept of "*the correct balance between the costs of input resources and the benefits . . .*".

In a developed maintenance strategy for engineering design, there are two basic types of maintenance costs that relate to the required input resources for effective maintenance:

- Costs arising from corrective maintenance action.
- Costs arising from preventive maintenance action.

Costs arising from corrective maintenance action are the costs of rectifying defects and fixing or repairing equipment. They increase exponentially according to the extent of usage that the equipment will be subject to, and according to the extent of failures resulting in downtime.

The manpower costs of corrective maintenance action are partly due to the time taken to restore the equipment to its expected operational condition within a minimum period of time or disruption to the overall operational process through the application of defect maintenance. Corrective maintenance costs are thus dependent upon the extent of defect maintenance, the effect of which is determined by MTTR, the performance measure of the equipment's maintainability. As noted before, maintainability is primarily a design parameter, and designing for maintainability defines how long equipment is expected to be down after failure, which has a direct impact upon corrective maintenance costs.

Costs arising from preventive maintenance action are the costs of detecting potential failures and avoiding functional failures. They increase linearly according to the age of the equipment and according to the extent of the maintenance schedules resulting in downtime.

The manpower costs of preventive maintenance action, which comprises both scheduled routine maintenance procedures, and scheduled preventive maintenance procedures incur a cost in direct proportion to the amount of routine maintenance being carried out, and to the amount of preventive maintenance being scheduled. Preventive maintenance costs are thus dependent upon the extent of routine and preventive maintenance, the effect of which is determined by potential usage and MTBF respectfully, which are the measures of performance of equipment availability and reliability. The inherent availability of equipment is its potential usage with respect to the operable time established from designing for availability, and the inherent reliability of equipment is initially established by its physical design and quality of manufacture established from designing for reliability.

By far the largest portion of preventive maintenance costs is associated with scheduled shutdowns and overhauls. Shutdowns and overhauls are scheduled according to the expected life of the major critical components in process engineering systems and equipment. In certain types of industries, particularly in refineries, several different types of shutdowns can be scheduled. They are:

- Interim shutdowns for vessel inspections.
- Open and clean shutdowns.
- Annual shutdowns for replacement of worn components.
- General overhauls for plant and equipment refurbishment.

The scheduled frequency and duration of interim shutdowns for vessel inspections, and of open and clean shutdowns can be determined according to a maintenance strategy in which the most suitable scope of preventive maintenance work is already established during the engineering design stage. The extent and duration of annual shutdowns for replacement of worn components can also be determined during the engineering design stage, and depends not only upon the expected useful life of the critical components of the process engineering design (i.e. failure characteristics) but also on the complexity of integrated systems, the level of equipment and/or component redundancy (i.e. process characteristics), as well as their relevant extent of usage.

General overhauls for plant and equipment refurbishment or rebuild are predominantly scheduled on the basis of results obtained, firstly, from condition monitoring carried out either periodically or continually and, secondly, from condition measurement carried out during interim shutdowns for vessel inspections and open and clean shutdowns. In principle, however, it is obvious that the costs of corrective maintenance action as well as the costs of preventive maintenance action can be rationalised or, in fact, reduced according to the balance of defect maintenance with routine maintenance and scheduled preventive maintenance, based upon a particular maintenance strategy. Such a strategy has its developed beginnings during the engineering design stage, and is progressively modified and improved during the life of the plant.

Mathematical model of preventive maintenance replacement costs The optimum operational period between annual shutdowns for replacement of worn components can be determined under a maintenance strategy of periodic replacement, irrespective of the age condition of the equipment's components. According to this strategy, components are replaced at predetermined intervals, CL, typically the length of the preventive maintenance cycle. If a component fails within this preventive maintenance cycle, it is minimally repaired to last for the remaining time of the cycle. Such a minimal repair job, with relatively negligible repair time, implies that the component's failure rate $\lambda(x)$, corresponding to its failure probability density function $f(x)$ at the time of failure x (i.e. the instantaneous failure rate), remains the same as it was before the failure (Kececioglu 1995).

The cost function for the model is expressed as

$$C_{\mathrm{pm}} = \frac{C_{\mathrm{pr}} + C_{\mathrm{mr}} E[\alpha(T_{\mathrm{p}})]}{T_{\mathrm{p}}} \tag{4.106}$$

where:

C_{pm} = preventive maintenance cycle costs
C_{pr} = the cost of preventive replacement
C_{mr} = the cost of minimal repair
$E[\alpha(T_{\mathrm{p}})]$ = the expected number of failures in interval T_{p}

and

$$E[\alpha(T_{\mathrm{p}})] = \int_0^{T_{\mathrm{p}}} \lambda(x)\,\mathrm{d}x \tag{4.107}$$

where

$$\lambda(x) = f(x)/R(x) \tag{4.108}$$

and:

$\lambda(x)$ = the equipment time dependent failure rate
$f(x)$ = the equipment failure probability density function
$R(x)$ = the equipment reliability function.

Substituting Eq. (4.107) into Eq. (4.106) gives the following result

$$C_{\mathrm{pm}} = \frac{C_{\mathrm{pr}} + C_{\mathrm{mr}} \int_0^{T_{\mathrm{p}}} \lambda(x)\,\mathrm{d}x}{T_{\mathrm{p}}} \tag{4.109}$$

In the case where $\lambda(x)$, the equipment time-dependent failure rate, has an exponential failure probability density function, i.e.

$$\lambda(x) = \frac{f(x)}{R(x)} = \frac{\lambda\,\mathrm{e}^{-\lambda x}}{\mathrm{e}^{-\lambda x}} = \lambda$$

differentiating with respect to T_{p} and setting the resultant equal to zero gives the following:

$$\frac{\mathrm{d}C_{\mathrm{pm}}}{\mathrm{d}T_{\mathrm{p}}} = \frac{-C_{\mathrm{pr}} + C_{\mathrm{mr}}\lambda T_{\mathrm{p}}}{T_{\mathrm{p}}^2} = 0\,.$$

The optimum operational period between annual shutdowns for preventive replacement is then

$$T_{\mathrm{p}} = C_{\mathrm{pr}}/C_{\mathrm{mr}}1/\lambda\,. \tag{4.110}$$

Optimal preventive replacement age of components subject to functional failure In many cases, systems and equipment are subject to functional failure, whereby the equipment or a component of the equipment has to be replaced. Where such functional failure is unexpected, it is not unreasonable to assume that a failure replacement is more costly than a preventive replacement. For example, a preventive replacement is planned, and arrangements are made for it to be conducted without unnecessary delays, or the unexpected failure may have caused consequential damage to other components. In order to reduce the number of failures, preventive replacements are made. However, a balance is required between the amount spent on preventive replacements, and the resulting benefits, i.e. reduced failure replacements.

Such a preventive replacement policy, or preventive maintenance strategy, is one where preventive replacements are made according to the 'right' age of the component, and failure replacements are done only when necessary, to minimise the total expected cost of replacing the component over a period of time. In this optimisation approach, when functional failures occur in equipment, failure replacements are made. The time at which preventive replacements are made depends upon the age of the component. The problem is to balance the cost of preventive replacements against their benefits of reduced failure replacements, which is done by determining the optimal preventive replacement age for the component so that the total expected costs are minimised over a period of time.

This is achieved with preventive replacement modelling with the following properties (Vajda 1974):

- C_{p} is the cost of preventive replacement.
- C_{f} is the cost of failure replacement.
- C_{c} is the total expected replacement cost per cycle.

- T_c is the expected cycle length.
- $f(t)$ is the probability density function of failures of the component.

The replacement policy is to perform preventive replacement once the component has reached a specified age, plus failure replacements when necessary, where the specified age is represented by t_p. The objective is to determine the optimal replacement age of the component to minimise the total expected replacement cost over a period of time.

In this problem, there are two possible cycles of operation: one cycle is determined by the component reaching its planned replacement age, t_p, and the other cycle is determined by the component ceasing to operate due to functional failure occurring before the planned replacement time. The total expected replacement cost, $C(t_p)$, over a period of time t_p is given by

$$C(t_p) = \frac{\text{Total expected replacement cost per cycle}}{\text{Expected cycle length}} \qquad (4.111)$$
$$C(t_p) = C_c/T_c$$

where the total expected replacement cost per cycle C_c is given as (the cost of a preventive replacement cycle multiplied by the probability of a preventive replacement) + (the cost of a failure replacement cycle multiplied by the probability of a failure replacement)

$$C_c = C_p R(t_p) + C_f[1 - R(t_p)] \qquad (4.112)$$

where $R(t_p)$ is the reliability of the component succeeding to last over the period of the preventive replacement cycle t_p. $R(t_p)$ is the probability of no failure occurring in the time period t_p, and the expression $[1 - R(t_p)]$ is the probability of failure occurring in the time period t_p, which is the failure density function.

Thus:

$$C_c = [C_p \times \text{Reliability}] + [C_f \times \text{Failure density}] .$$

The expected cycle length T_c is given as (the length of the preventive replacement cycle multiplied by the probability of a preventive replacement) + (the expected length of a failure replacement multiplied by the probability of a failure replacement)

$$T_c = t_p R(t_p) + t_f[1 - R(t_p)] . \qquad (4.113)$$

In this case, t_f is the *mean time to fail (MTTF)* of the component. Here, it is important to take note of the description of MTTF, compared to MTBF, the mean time between failures. The difference between MTTF and MTBF is in their *usage*. MTTF is applied to items that are not repaired but replaced, such as components, whereas MTBF is applied to items that are repaired. Therefore:

$$T_c = [\text{Replacement age} \cdot \text{Reliability}] + [\text{MTTF} \cdot \text{Failure density}]$$

The replacement model relates replacement age t_p to the total expected replacement cost over a period of time, where

$$C(t_p) = \frac{C_p R(t_p) + C_f[1 - R(t_p)]}{t_p R(t_p) + t_f[1 - R(t_p)]} \tag{4.114}$$

$$C(t_p) = \frac{[C_p \cdot \text{Reliability}] + [C_f \cdot \text{Failure density}]}{[\text{Age} \cdot \text{Reliability}] + [\text{MTTF} \cdot \text{Failure density}]} .$$

Thus, the essential integrity measures for determining the total expected replacement cost, $C(t_p)$, over a period of time t_p, in addition to the cost of preventive replacement C_p and the cost of failure replacement C_f are the component (or equipment) *reliability* and *failure density*. Values for the specific costs of C_p and C_f as well as component reliability and the failure density (or 1–reliability), and MTTF must be evaluated in order to determine the minimum total expected replacement cost $C(t_p)$ over the period of time t_p. Preventive replacement age is where $C(t_p)$ is minimum.

Cost of input resource of spares A significant portion of preventive maintenance costs, during ramp-up and the specified warranty period, as well as the remaining life-cycle stages of an engineered installation, is the input resource of spares. Spares for engineered installations can be grouped according to two categories:

- Contract spares
- Maintenance spares.

Contract spares are normally part of the initial procurement of systems and equipment, and are determined by available reliability data from the manufacturer or vendor. The main concern with contract spares is not so much the quantity, or individual cost, but rather their identification. Determination of maintenance spares is achieved through the method of maintenance *spares requirements planning (SRP)*.

SRP can be defined as "*a strategy involving the purchasing, supply, identification, storage and issue of spare parts which improves system maintenance and results in an increase of plant availability*".

SRP is different from *inventory control*. *SRP* is better suited to maintenance spares that have a high-risk component failure and estimated equipment failure rate. *Inventory control* is better suited to maintenance spares with low-risk component failure and estimated stock levels. With *SRP*, the required spares are calculated according to the estimated failure rate of the relevant equipment, and according to the criticality of the equipment with regard to downtime costs.

Inventory control is a resource management system that makes use of calculated order-points, reorder quantities, and forecasts of the stock level at which stock must be replenished as well as the quantity to be ordered. It is evident that SRP considers single items of spare parts for equipment when they are needed, whereas inventory control considers many items to be placed into stock until they are needed.

SRP determines the efficiency level of the availability of spares for maintenance, and thus minimises downtime as well as avoids holding unnecessary spare parts in stock. Inventory control determines the service level of the stores in not being out of

stock with spare parts, and thus also optimises on spares stock levels (Orlicky et al. 1970).

Both SRP as well as inventory control are important to managing spare parts for maintenance, but it is essential to understand that each of these methods are applied to specific types of spares. The types of maintenance spares that are managed through SRP and inventory control are determined from the demand for these spares by the type of maintenance action. There are two types of demand for maintenance spares:

- Dependent demand
- Independent demand.

Dependent demand for maintenance spares relates to the need for the replacement of other components of which the maintenance spare is a part. Dependent demand is based on the systems hierarchy structure of the process or equipment that forms the basis of a bill of spares for a spares requirements planning system. Independent demand for maintenance spares relates to the demand of the maintenance spare on its own, and is not subject to the need for other components or parts. Independent demand is based on forecast usage of the spares that forms the basis of order-points and reorder quantities for an inventory control system. It is evident from these descriptions that different categories of spares can be grouped under the two types of demand. There are several general categories of maintenance spares:

- Consumable materials (materials that are used up through the maintenance action, such as oils, greases, waste cloth, etc.).
- Consumable spares (spares that are used up in the operation of the equipment or process, such as filters, pump impellors, turbine blades, tube bundles in coolers, etc.).
- Replacement spares (parts that become worn through excessive usage or insufficient routine maintenance, or that need to be replaced due to defects, damage or failure. These spares are mostly the parts of components such as bearings, sleeves, liners, etc.).
- Repairable spares (assembled units that are repaired or overhauled through the replacement of parts and then returned to stores (RTS) for later re-issue, such as electric motors, valves, pumps, etc.).
- Critical spares (spares that are kept in stores for insurance against hazardous failures of critical equipment, such as special high-pressure or acid resistant valves, high-voltage electrical parts, etc.).
- Strategic spares (spares that are kept in stores for insurance against high downtime costs due to long ordering lead times, such as special alloy parts, specialised engineered parts, etc.).

There is a further category that is called capital spares, which are not really maintenance spares and consist of assembled units that are very expensive and are usually categorised by very high capital equipment industries such as power generation plants. Most stores in industry make use of an ABC classification system to categorise the types of stock being held but, in many cases, this ABC classification has proved to be inadequate to support effective maintenance strategies.

Dependent demand maintenance spares usually consist of some replacement spares, repairable spares, critical spares and strategic spares that are stocked because of the risk or frequency of failure of the relevant equipment. These spares are controlled through a spares requirements planning (SRP) system. *Preventive maintenance* makes use of dependent demand maintenance spares, and is therefore associated with SRP.

Independent demand maintenance spares usually consist of consumable materials, consumable spares and some replacement spares that need to be stocked irrespective of the frequency of component replacement. These spares are controlled through an order-point and reorder quantity inventory control system. *Routine maintenance* makes use of independent demand maintenance spares, and is thus associated with inventory control.

Because the sort of maintenance spares that are controlled through an SRP system are typically the logistic support spares required for shutdowns and general overhauls (i.e. some replacement spares, repairable spares, critical spares and strategic spares that are stocked because of the risk or frequency of failure of the relevant equipment), SRP is extremely important for the effective application of preventive maintenance, and also for the effective use of contracted maintenance crews during shutdowns and overhauls (Hillestad 1982).

Mathematical modelling of spares requirement Most spares requirements optimisation models assume the *constant failure rate* to be a good approximation for a *constant demand rate*, even if components have non-constant failure rate distributions. Such a failure rate is fundamentally a measure of the intrinsic failure characteristics of a component brought about by usage stress and load over time. However, it is not quite correct to express the demand rate for a spare simply by the intrinsic failure characteristic of a component.

In most cases, the demand for a given spare is the result of a number of factors. Firstly, there may be several different items of equipment that require the same spare. Secondly, there could be several similar parts in each component. Thirdly, there are usually a large number of similar components within each system. Clearly, it is cumbersome to derive the exact spares demand based on the component failure rate. Furthermore, it is somewhat unrealistic to assume a specific failure rate of a component within a complex integration of systems with complex failure processes. At best, the intrinsic failure characteristics of components are determined from quantitative probability distributions of failure data obtained in a somewhat clinical environment under certain operating conditions. As indicated before, the true failure process depends upon many other factors, including, for example, routine and preventive maintenance. It is generally accepted that preventive maintenance affects the failure properties of components, although it is debatable whether the end result is positive or negative from the point of view of equipment *residual life*.

When modelling spares requirements, the foremost criterion to take cognisance of is that the need for spares is determined by a spares *demand*. This demand is formed by and dependent upon several factors, such as (Alfredsson et al. 1999):

- Equipment and/or system utilisation.
- Failure occurrence in the equipment.
- Failure mode of the failed component.
- Failure consequence and severity.
- Number of similar parts or components.
- Frequency of preventive maintenance replacement.

Although seemingly problematic from the perspective of complexity, the multiplicity of similar parts in each component, with usually a large number of similar components within each system, is in fact beneficial in characterising the *demand* for different kinds of spares. It validates the application of classical *limit theory* concerning the maintenance renewal process. This is illustrated by the following theorem (Drenick 1960):

> given N components, indexed by $i = N, K, 1$, of which the failure processes are independent renewal processes, let $F_i(t)$ be the distribution for the time between failures of component i. Furthermore, λ_i is the expected number of renewals per time unit, so that its reciprocal, $1/\lambda_i$, is the expected time between failures of component i.

Let $G_N(t)$ be the distribution of the time between failures across all components. If:

(i) $\lim_{N\to\infty} \lambda_i / \sum_{i=1}^{N} \lambda_i = 0$

(ii) $F_i(t) \leq At^{\sigma}$ and $A > 0$, $\sigma > 0$ as $t \to 0\ \forall i$

then

$$\lim_{N\to\infty} G_N\left(t / \sum_{i=1}^{N} \lambda_i\right) = 1 - \mathrm{e}^{-\lambda t} \quad \text{for } t > 0\,. \tag{4.115}$$

Consequently, Drenick's theorem states that, under the above assumptions, the pooled output will approach a Poisson process as the number of failures increase. Condition (i) is non-restrictive. Condition (ii) is satisfied by all failure distributions commonly used—for example, the Weibull distribution. Thus, when the demand for a spare is the result of several component failure processes (which it normally is), the demand tends to be approximated by a Poisson distribution—that is, the demand rate is constant, irrespective of whether the individual components have arbitrary failure characteristics.

There are only a few quantitative methods available when determining spares requirements. These are identified as analytical methods based on *constant demand rates*, analytical methods based on *renewal theory*, as well as simulation models. Analytical methods based on constant demand rates tend to be the most applicable for spares requirements modelling.

Renewal theory describes component failure by the renewal process that is characterised by a distribution for the time between renewals denoted $F(t)$. If the distribution $F(t) = 1 - \mathrm{e}^{-\lambda t}$, then the renewal process is a Poisson process with rate λ. Hence, the renewal process is usually a generalisation of the Poisson distribution.

However, the renewal process does not include several properties of the Poisson distribution. Most importantly, the result of two independent renewal processes is not a renewal process unless both processes are Poisson processes. Furthermore, the probabilistic split of a renewal process does not yield independent renewal processes.

As indicated previously, when modelling for spares requirements, the demand is ultimately dependent upon several factors. Spares demand is in most circumstances the result of the component failure characteristics. If the component failure is modelled as a renewal process, the spare demand is not a renewal process. In effect, models based on renewal theory have limited applicability in terms of spares optimisation. Such models are limited to a single process—that is, a single system, single component, and single part situation, which is very rare when determining an optimum spares requirements strategy for a real-world engineering design.

Simulation models are generally impractical for spares optimisation (or, in fact, any kind of optimisation). Event-driven simulation can be applied to analyse basically any stochastic system or process. In terms of optimisation, however, it is not applicable. The reason for this is the relatively extensive time required for a single function evaluation. Any optimisation algorithm iteratively evaluates an objective function and/or its derivatives numerous times in order to establish the optimal solution. If each function evaluation takes time, the optimisation algorithm soon becomes impractical. Function evaluation is generally much faster, and optimisation feasible with analytical models based on Poisson demand (constant demand rate). An analytical method for spares requirements based on a Poisson demand, or *constant demand rate*, which is approximated by the *constant failure rate*, can thus be developed (with a sufficient degree of acceptance) as the probability of having a spare when required. Such a probability takes into consideration the constant failure rate of an item (component or part) that is intended to have a spare, the number of items in the equipment and/or system that are intended to have spares (critical items), and the number of items in the system as a whole. The following model can be used to determine the spares requirement quantity (Blanchard et al. 1995):

$$\mathrm{SP} = \sum_{i=0}^{m} [(-1)\ln(\mathrm{e}^{-n\lambda_t})]^i \mathrm{e}^{-n\lambda_t}/i! \tag{4.116}$$

where:

SP = the probability of having a spare when required
m = the number of items in the system as a whole
n = the number of items intended to have spares
t = period of time in which an item is likely to fail
λ = the constant failure rate of an item intended to have a spare.

4.2.3 Theoretical Overview of Availability and Maintainability Evaluation in Detail Design

Availability and maintainability evaluation determines the *measures of time that are subject to equipment failure*, particularly *known values of failure rates and repair rates* for each individual item of equipment at the lower systems levels of the systems breakdown structure. Availability and maintainability evaluation is considered in the *detail design* phase of the engineering design process, with determination of the rates and frequencies that *component* failures occur and are *repaired* over a specified period of time. The most applicable methodology for availability and maintainability evaluation in the detail design phase includes basic concepts of mathematical modelling such as:

i. *Dependability modelling for availability and maintainability*
ii. *Operational availability modelling subject to logistic support*
iii. *Maintainability evaluation and built-in or non-destructive testing*
iv. *Specific application modelling of availability and maintainability*.

Due to the increasing complexity of engineering processes, it is unrealistic to expect that standard specifications covering the operational evaluation of a system are adequate for detail engineering designs. The problem in the specification of the operational process is complexity. Potential deviations from the expected operational behaviour can be caused by unexpected failures in a complex system environment, or by the complex integration of several systems. To challenge the problems of complexity, all possible operational sequences must be considered in an operational specification, essential for modelling a complex system in its expected operational state, or at least according to a predetermined level of abstraction of such an operational state. This form of modelling, which incorporates operational specifications during the detail design phase of the engineering design process, is often termed *operational modelling*. The aim of operational modelling is to determine the operational view of an engineering design, and to integrate it with operational and technical specifications to guarantee model consistency. Various operational models are considered, including a graphical formalism appropriate for modelling concurrent processes, and thus for describing the operational view of complex integrated systems.

4.2.3.1 Dependability Modelling for Design Availability and Maintainability

Dependability is the measure of a system's condition during operation, provided that it is available for operation at the beginning of its application (i.e. *operational availability*, which will be considered in detail in the following section). Dependability can also be described as the probability that a system will accomplish its intended application (or mission), provided that it was available for operation from the beginning (Dhillon 1999b). Dependability models used for the evaluation of performance

of an engineering design are considered from a twofold meaning of the concept of dependability (Zakarian et al. 1997):

- System operational integrity
  (reliability, availability and maintainability).
- System performance
  (dependence on the performance of equipment).

A dependability model that considers the *operational integrity* of a process engineering system, where the system is considered to be operational as long as its functional requirements are satisfied, includes the measures of *operational integrity* (operational reliability R_o, operational availability A_o, and operational maintainability M_o). A dependability model that considers *system performance* includes measures of the *process characteristics*. In other words, a process system is assumed to function properly if it is able to achieve the required level of performance where the *process capability*, as given in Eq. (4.17), exceeds a given lower bound of a particular process characteristic. Careful consideration of these concepts of dependability of a process engineering system during the engineering design stage can definitely improve system dependability.

Dependability D_s, considering system *operational integrity*, is modelled as

$$D_s = M_o(1 - R_o) + A_o(R_o) \tag{4.117}$$

where:

R_o = operational reliability as fraction/percentage
A_o = operational availability as fraction/percentage
M_o = operational maintainability as fraction/percentage.

Expressing system dependability in performance measures for operational reliability, availability and maintainability would include the measures of MTTR and MTBF. In this case, system dependability is the sum of the ratios of system uptime to total cycle time, and system repair time to total downtime.

It is therefore an indication of the fraction of time that a system is *available* in a cycle of system operation and failure, plus the fraction of time that the system is *repairable* when it is down (i.e. the ability of being used when it is up plus the ability of being repaired when it is down). Thus

$$D_s = A_o + \mathrm{MTTR}/\mathrm{MDT} \tag{4.118}$$

In the case where the performance measure of operational availability can be expressed as

$$A_o = \frac{\mathrm{MTBF}}{\mathrm{MTBF} + \mathrm{MDT}} \tag{4.119}$$

where:

MDT = expected mean downtime
MDT = $T_{pm} + T_{cm} + T_{ld}$

where:

T_{pm} = preventive maintenance downtime
T_{cm} = corrective maintenance downtime
T_{ld} = logistics and administrative downtime

then

$$D_{\mathrm{s}} = \frac{\mathrm{MTBF}}{\mathrm{MTBF} + \mathrm{MDT}} + \frac{\mathrm{MTTR}}{\mathrm{MDT}} \,. \tag{4.120}$$

In the case where the expected mean downtime includes only preventive maintenance downtime, the availability performance measure becomes *inherent availability*, and D_{s} is expressed as

$$D_{\mathrm{s}} = \frac{\mathrm{MTBF}}{\mathrm{MTBF} + \mathrm{MTTR}} + \frac{\mathrm{MTTR}}{T_{\mathrm{pm}}} \,. \tag{4.121}$$

4.2.3.2 Operational Availability (A_{o}) Modelling with Logistic Support

Operational availability, unlike *inherent* availability or *achieved* availability, covers all segments of time that the system's equipment is intended to be operational (total time in Fig. 4.1). The same uptime and downtime relationship exists, except that it has been expanded. Uptime now includes operating time plus non-operating (standby) time when the equipment is assumed to be *operable*. Downtime has been expanded to include preventive and corrective maintenance and the associated administrative and logistics lead time. All are normally measured in clock time. This relationship is intended to provide a realistic measure of equipment availability when the equipment has been installed and is functioning in an operational environment. Operational availability is used to support operational testing assessment and life-cycle costing.

Operational availability is the most desirable form of availability to be used in evaluating the operational potential of equipment, and is an important measure of system effectiveness because it relates the system's equipment, logistic support and environment characteristics into one meaningful parameter—an index depicting the state of equipment at the beginning of its operation in an engineered installation. Because it is an effectiveness-related index, operational availability is used as a starting point for nearly all system effectiveness and sizing analyses during the later stages of the engineering design process.

One significant problem associated with evaluating operational availability is that it becomes costly and time-consuming to define all the various parameters, especially during the detail engineering design phase when all equipment (assemblies and components) are being identified. For instance, defining administrative and logistics downtime per equipment per specified period, and total preventive maintenance under normal operational conditions is very difficult and not feasible in many cases. Nevertheless, evaluating operational availability does provide an accepted methodology of relating standard reliability and maintainability characteristics into

a single effectiveness-oriented parameter. As such, it is an essential tool for determining the integrity of engineering design. An important aspect to take note of when evaluating operational availability is that it is affected by equipment usage or *utilisation rate*. The less an item is used in a given period, the higher the operational availability will be.

Therefore, when defining the 'total time' period, it is important to exclude lengthy periods during which little or no system usage is anticipated. One other expression for operational availability is when standby time is assumed to be zero, typical of single stream processes with no equipment redundancy. While maintenance-oriented, this form of operational availability still retains consideration of the same basic time elements. The downtime interval includes corrective and preventive maintenance, as well as administrative and logistics downtime. This form of operational availability would generally prove more useful in support of defining preventive maintenance requirements and logistic support analysis during the detail design phase of the engineering design process. The general mathematical model for operational availability is (Conlon et al. 1982):

$$A_o = \frac{\mathrm{OT} + \mathrm{ST}}{\mathrm{OT} + \mathrm{ST} + \mathrm{TCM} + \mathrm{TPM} + \mathrm{ALDT}} \tag{4.122}$$

where:

OT = operating time
ST = standby time
TCM = total corrective maintenance
TPM = total preventive maintenance
ALDT = administrative and logistics downtime.

Inherent availability looks at availability from a design perspective, whereas operational availability considers *system effectiveness* and the operational potential of equipment, and is used for analysing the sizing of equipment during the later stages of the engineering design process. Thus, more encompassing maintainability measures of mean time between maintenance and mean downtime are used in the operational availability equation. Operational availability is, in effect, a model of maintainability measures in which downtime resulting from both corrective and preventive maintenance is considered. A_o is thus a smaller availability value than A_i. Operational availability can thus be mathematically expressed as

$$A_o = \frac{\mathrm{MTBM}}{(\mathrm{MTBM} + \mathrm{MDT})} \tag{4.123}$$

where:

MTBM = mean time between maintenance
MDT = mean downtime.

The *mean time between maintenance* (MTBM) includes all corrective and preventive actions (compared to MTBF, which accounts for failures—in contrast to the

concept of A_o for dependability in Eq. (4.119)). The *mean downtime* (MDT) includes all time associated with the system being down for corrective maintenance including delays (compared to MTTR, which addresses only repair time), including downtime for preventive maintenance (PM), plus administrative and logistics downtime. Although it is preferred to design equipment for which most PM actions can be performed while the equipment is operating (such as built-in testing, BIT), PM in this context implies a certain downtime.

The uptime and downtime concepts for constant values of availability indicate the relative difficulty of increasing availability at higher percentages, compared to improving availability at lower percentages. This is illustrated by the fact that increasing availability from 99 to 99.9% requires an increase in MTBM by one order of magnitude or a decrease in MDT by one order of magnitude, whereas increasing availability from 85 to 90% requires improving MTBM by less than $1/2$ order of magnitude or decreasing MDT by $3/4$ order of magnitude.

a) General Approach for Evaluating Operational Availability

The operational and maintenance concepts associated with system utilisation must be defined in detail using terminology compatible with all involved in the design of engineered installations. Using these definitions, a time-line availability model is constructed that reflects the availability parameters, as illustrated in Fig. 4.9 (Conlon et al. 1982).

Figure 4.9 displays elements of availability, particularly standby times (ST_W) and (ST_C), which are included in quantitative operational availability.

The up or down status of a system during preventive maintenance must be closely examined because, generally, a portion of the preventive maintenance period may be considered as uptime. Standby time must also be examined closely before determining system up or down status during this period. With the aid of the time-line model, all time elements that represent uptime and downtime are determined. For example, a maintenance strategy may be defined so that the equipment is maintained in a committable or up-state during the performance of preventive maintenance.

Additionally, for multi-mode systems, it will be necessary to determine uptimes and downtimes as a function of each mode. This generally will require the use of a separate time-line model for each identifiable operational mode. Separate time-line

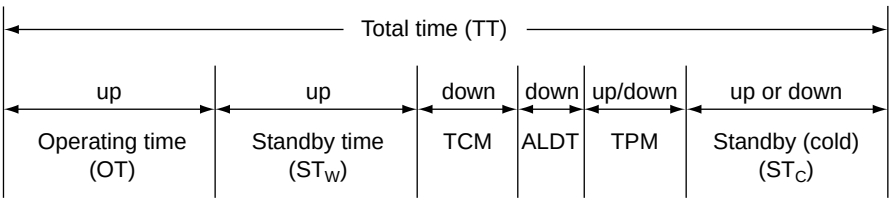


Fig. 4.9 Operational availability time-line model—generalised format (DoD 3235.1-H 1982)

models are generally required to support the availability analyses of systems that experience significantly different continuous, periodic, and surge utilisation rates. Quantitative values for the individual time-line models are determined and coordinated with the engineering design project management baselines. Time elements are computed and availability evaluated, using the definitions of operational availability appropriate for the detail design phase. Availability model status is continually checked and updated as required. The model is updated as the operational, maintenance and logistics support concepts progressively become defined and quantifiable.

b) System Availability Evaluation Considerations

As indicated previously, the quantitative evaluation of availability must be carefully and accurately tailored to each system. However, there are certain general concepts that will apply to different types of process engineering systems, such as *recovery time*. Normally, availability measures imply that every hour has equal value from the viewpoint of operations and maintenance/logistics activities. The operational concept requires the system to function only for selected periods. The remaining time is traditionally referred to as 'off-time' during which no activity is conducted. An alternative to 'off-time' or 'cold standby' is the use of the term 'recovery time'. Recovery time represents an interval of time during which the system may be up or down (Fig. 4.10). Recovery time, RT, does not appear in the operational availability calculation that is based only on the total time period TT. Significantly, corrective maintenance time TCM is found in both TT and RT time intervals.

Corrective maintenance performed during the TT period is maintenance required to keep the system in an operational available status. Corrective maintenance performed during the RT period generally addresses malfunctions that do not result in a downtime status.

The principal advantage of using recovery time analysis is that it can provide a meaningful availability evaluation for systems with operational availability that is predictable, and preventive maintenance that constitutes a significant portion of maintenance time. The recovery time calculation technique concentrates availability calculation during the operational time period, thereby focusing attention on critical uptime and downtime elements.

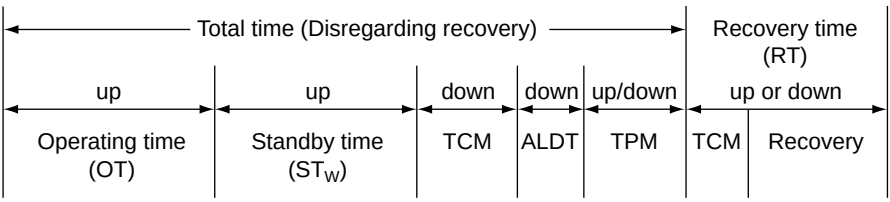


Fig. 4.10 Operational availability time-line model—recovery time format (DoD 3235.1-H 1982)

4.2.3.3 Maintainability Evaluation and Built-In or Non-destructive Testing

Maintainability has been defined as a characteristic of design and installation. It is this inherent characteristic of a completed engineering design that determines the type and amount of maintenance required to restore or retain it in a specified condition. Where maintainability is a design consideration, maintenance is the consequence of the design. It is thus apparent that the ability and need to perform maintenance actions is the underlying consideration when evaluating maintainability. The consideration of maintenance when designing engineering systems is not new. There have been very successful efforts in the development of design for accessibility, built-in testing, etc. What is new is the emphasis on *quantitative assessment* and evaluation that results in a complete change in engineering design philosophy, methodology and management. In the past, design for maximum or optimum reliability and maintainability was emphasised. However, all this resulted in was unknown reliability and maintainability from the design stage through to installation.

New techniques and methods allow design integrity judgment to be quantitatively measured, as in the case of *maintainability evaluation*. Maintainability evaluation is the determination of design considerations and testing, intended to evaluate system maintainability characteristics that are based on quantitative measures or indices. In addition to evaluating these characteristics, maintainability evaluation should also address the impact of physical design features on system maintenance and maintenance action frequency.

There are various mathematical indices used to evaluate system maintainability characteristics. These indices must be composed of measurable quantities, provide effectiveness-oriented data, and must be readily obtainable from applicable development testing, such as the use of *non-destructive testing* (NDT) internal or integrated diagnostic systems, also referred to as *built-in-test* (BIT) or *built-in-test-equipment* (BITE), and applied to pilot systems as well as to the engineered installation. The use of *maintainability evaluation indices* enables engineering designers to evaluate system and/or equipment characteristics as well as logistics and maintenance practices more precisely during the detail design phase.

a) Maintainability Evaluation Indices

Mean time to repair (MTTR) As noted previously, the maintainability measure of *mean time to repair* (MTTR) is the total corrective maintenance downtime accumulated during a specific period, divided by the total number of corrective maintenance actions completed during the same period. MTTR is commonly used as a general equipment maintainability measure, although it can be applied to each maintenance level individually. MTTR considers active corrective maintenance time only. Because the frequency of corrective maintenance actions and the number of man-hours expended are not considered, this index does not provide a good measure of the maintenance burden.

Maximum time to repair (MaxTTR) MaxTTR is the maximum corrective maintenance downtime within which either 90 or 95% (as specified) of all corrective maintenance actions can be accomplished. A MaxTTR requirement is useful in those special cases in which there is a tolerable downtime for the system.

An absolute maximum is ideal but impractical because there will be failures that require exceptionally long repair times. A 95th percentile MaxTTR specification requires that no more than 5% of all corrective maintenance actions take longer than MaxTTR.

Maintenance ratio (MR) MR is the cumulative number of man-hours of maintenance to be expended in direct labour over a given period of time, divided by the expected cumulative number of end-item operating hours. Both corrective and preventive maintenance are included. Man-hours for off-system repair of replaced components, and man-hours for daily operational checks are included for some systems. Particular care must be taken that the operating hour base be clearly defined, such as in the case of power-generating systems, when either system operating hours or power delivery hours can be used. MR is a useful measure to determine the relative maintenance burden associated with a system. It provides a means of comparing systems and is useful in determining the compatibility of a system with the required size of the maintenance organisation.

Mean time between maintenance actions (MTBMA) MTBMA is the mean of the distribution of the time intervals between either corrective maintenance actions, preventive maintenance actions or all maintenance actions. This index is frequently used in availability calculations and in statistically oriented maintenance analyses.

Average number of maintenance man-hours required The average number of maintenance man-hours required at each maintenance level provides a quantitative means of expressing the personnel requirements of the overall maintenance concept. This index also provides a conversion factor from active downtime to labour hours.

Maintainability cost indices Maintainability is a significant factor in the cost of equipment. An increase in maintainability results in a reduction of logistic support costs of engineered installations. A more maintainable system inevitably reduces maintenance times and operating costs, and a more efficient maintenance turnaround reduces downtime. There are many factors of maintainability that contribute to the investment costs of engineered installations. These include a direct effect on system and equipment hardware costs, support equipment, built-in testing, and contract spares.

Off-system maintainability indices The indices MTTR, MaxTTR and MR all specifically exclude off-system maintenance actions. Off-system measures are particularly important if a system's maintenance strategy involves extensive use of modular removal and replacement for workshop repair/overhaul, since this type of concept transfers the maintenance burden to off-system maintenance. As a maintainability evaluation tool for engineered installations, off-system maintainability measures are essential. Without these, it is not possible to evaluate the ability of

off-system repair, and logistics capability, to maintain the engineered installation. However, because of the peculiar nature of these parameters, none are considered in this research, although it is essential to have a complete set of on-system and off-system indices to adequately assess system maintainability and the total maintenance burden.

b) Diagnostic Systems and Built-In Testing

One aspect of maintainability that has received significant attention in recent system designs is the use of *automatic diagnostic systems*. These systems include both internal or integrated diagnostic systems, referred to as *built-in-test* (BIT) or *built-in-test-equipment* (BITE), and external diagnostic systems, referred to as *automatic test equipment* (ATE), or offline test equipment. The following concepts focus on BIT but apply equally to other diagnostic systems.

Need for automatic diagnostic systems—BIT As technology advances continue to increase the capability and complexity of modern engineering processes, particularly in space and military systems, more reliance is being placed on the use of automatic diagnostics as a means of attaining the required level of failure detection capability. The need for BIT is driven by operational availability requirements, which cannot allow for lengthy MTTRs associated with detecting and isolating failure modes in engineering designs, especially in microcircuit technology equipment. Because BIT is applied within a system's function, and at the same functioning speed, it affords the capability to detect and isolate failures that conventional test equipment and techniques cannot provide. A well-designed BIT system can substantially reduce the need for trained field-level maintenance personnel by permitting less skilled personnel to locate failures and channel suspect equipment to centralised workshop repair facilities that are equipped to repair defective equipment.

However, BIT is not a comprehensive solution to all system maintenance requirements but, rather, a necessary tool for maintaining complex integrated systems.

Specifying BIT performance One of the more difficult tasks inherent in the design and development of process engineering systems is the development of realistic and meaningful operational requirements and their subsequent conversion into understandable and achievable contractual specifications. This is equally applicable to BIT, particularly with respect to typical performance measures or figures-of-merit that are used to specify BIT performance.

Typical BIT performance measures, or figures-of-merit

- Percent detection—the percent of all faults or failures that the BIT system must detect.
- Percent isolation—the percent of detected faults or failures that the system must isolate to a specified assembly level.
- Automatic fault isolation capability (AFIC)—the percent detection multiplied by the percent isolation.

- Percent of false alarms—the minimum tolerable percent of indicated faults where, in fact, no failure is found to exist.

For each of the above parameters, there is a considerable span of interpretation. For example, does the percent detection refer to failure modes or to the percentage of all failures that could potentially occur? Furthermore, does the detection capability apply across the failure spectrum, i.e. mechanical systems, instrumentation, connections and software, or is its diagnostic capability applicable only to certain hardware such as electronic systems? Also, to what systems hierarchy level will the BIT system isolate failures?

Early BIT systems were designed to isolate faults at component level. This resulted in BIT systems being as complex as, and frequently less reliable than, the basic system. The current trend is to isolate faults to the sub-system or assembly level based on the BIT system's ability to detect abnormal output signal patterns. Large industry workshop maintenance facilities frequently apply external diagnostic equipment to isolate to the component or part level.

A major engineering design issue (as well as contractual issue) relates to the definition of failure. Should BIT performance be viewed in terms of only BIT addressable failures, which normally exclude system interface components such as exchangers, crossover ducts, pipelines, connectors, cables, etc., and which are usually the failure critical components in complex integrated systems? An important consideration thus relates to exactly what failures BIT can detect. Often, BIT systems operate ineffectively if 80% of detectable failures occur infrequently while the remaining 20% occur with predictable regularity. It therefore becomes important to specify BIT performance measures in relation to overall system availability requirements.

The percent of false alarms is a difficult parameter to specify or to measure accurately because initial fault detection followed by analysis indicating that no fault exists can signify different possible occurrences, such as:

- The BIT system erroneously detected a fault.
- An intermittent out-of-tolerance condition exists.
- A failure exists but cannot be readily reproduced in a maintenance environment.

From a logistic viewpoint, false alarms can often lead to false removals creating unnecessary demands on supply and maintenance systems. A potentially greater concern is the fact that false alarms and removals may create a lack of confidence in the BIT system to the point where maintenance or operations personnel may ignore certain fault detection indications. Under these conditions, the BIT system in particular and the maintenance concept in general can neither mature nor provide the support required to meet design requirements.

The specification of BIT performance must therefore be tailored to the type of system being designed, as well as to the system design criteria. Designing for maintainability must include a comprehensive definition of BIT capability based upon the figures-of-merit presented above.

Characteristics external to BIT There are two important considerations, external to BIT, which must be addressed in the concept of BIT and diagnostics in designing for maintainability. Initially, reliable performance of the designed system determines, to a large extent, the criticality of BIT performance.

If the basic system is designed to be very reliable (in the region of 0.995 and 0.999), a shortfall in the BIT performance may have limited impact on the system's operational utility. Moreover, it is obvious that generally all system faults that can be corrected through maintenance action must initially be detected and isolated. Therefore, design for maintainability requirements such as maintenance methods, tools, manuals, test equipment and personnel required to detect and isolate non-BIT detectable faults can be a major consideration in the detail design phase of engineered installations. BIT is inherently an aspect of design for maintainability.

The following example illustrates the impact of BIT on the overall maintenance effort. It further attempts to illustrate the effect of external factors on BIT performance (DoD 3235.1-H 1982).

Description: a radar installation is composed of five line replaceable units (LRUs) with the following BIT and system performance characteristics:

System:	Five (5) LRUs
MTTR (w/BIT):	2 h (includes failures that have been both detected and isolated)
MTTR (no/BIT):	5 h (includes failures that have been detected but not isolated)
MTBF:	50 operational hours
Period of interest:	2,500 operational hours
BIT specified:	percent detection = 90%
	percent isolation = 90% (to the LRU level)
	false alarm rate = 5% (of all BIT indications)

In this example of a sophisticated military engineered installation, a relatively high-capability BIT system has been specified, where industrial installations with BIT would be less rigorously specified. Upon cursory examination, this extensive BIT coverage would appear to require minimal additional maintenance. The problem is to determine what total corrective maintenance time would be required for 2,500 operating hours. Thus:

- How many total failures could be expected?
  2,500 total hours at 50 MTBF = 50 failures
- How many of these failures (on average) will BIT detect?
  50 failures × 90% = 45 BIT detected failures
- How many detected failures on average will be isolated to an LRU?
  45 detected failures × 90% isolation = 40 failures
- What is the automatic fault isolation capability (AFIC)?
  % detection × % isolation (LRU) = AFIC
  0.9 × 0.9 = 0.81 = 81%
- How many false alarm indications are expected to occur during the 2,500 operational hours?
  Total BIT indications (I_{BIT}) = true failure detections + false alarms

I_{BIT} = (BIT detection rate) × (total failures) + (false alarm rate) × (total BIT indications)
$I_{BIT} = (0.90) \times (50) + (0.05) \times (I_{BIT})$
$(1 - 0.05)\, I_{BIT} = 45$
$I_{BIT} = 47.36$
and:
False alarms = total BIT indications − rue indications
False alarms = 47.36 − 45
$= 2.36 \approx 2$

With this information, the total *corrective maintenance* time can now be calculated (DoD 3235.1-H 1982):

- What is the total corrective maintenance time (on average) required to repair the detected/isolated failures?
  TC (w/BIT) = 40 failures × 2 h (MTTR w/BIT)
  = 80 h
- What is the total corrective maintenance time (on average) required to repair the remaining no/BIT detected/isolated failures?
  TC (no/BIT) = 10 failures × 5 h (MTTR no/BIT)
  = 50 h
- If it is assumed that no/BIT maintenance time is required to sort out false alarm indications, what total no/BIT corrective maintenance time is required for the 2,500 flying hour period?
  TC (no/BIT) = no/BIT repair time + false alarm maintenance time
  $= (10) \times (5) + (2) \times (5) = 60$ h
- What is the total corrective maintenance time TC (total) anticipated during the 2,500 hours?
  TC (total) = BIT maintenance + no/BIT maintenance
  $= 80 + 60 = 140$ h

Thus, even with a relatively high AFIC of 81%, the no/BIT-oriented corrective maintenance represents 43% of the total anticipated corrective maintenance hours. Furthermore, the impact of scheduled/preventive maintenance has not been considered. This additional maintenance is generally not associated with BIT.

The information presented in this example is greatly simplified in that it is assumed that the BIT AFIC (% detection × % isolation) will be 81%. If the AFIC is 81%, then 57% of the maintenance effort will be oriented towards BIT detected/isolated failures. If the true AFIC is found to be lower, it will be necessary to re-evaluate the overall effectiveness of the maintenance strategy and logistics program, as well as total system effectiveness (DoD 3235.1-H 1982).

c) Basic System and BIT Concurrent Design and Evaluation Considerations

In designing for maintainability, the difficulty involved in the design and evaluation of BIT that must perform in accordance with specific basic system specifications

and design criteria is a problem of concurrent design. The development and evaluation of BIT and fault diagnostics has traditionally followed basic system engineering design. The argument usually presented is that the basic system has to be designed and evaluated before determining what the BIT is intended to test. This argument has some basis, in fact, but there are significant drawbacks associated with lengthy design schedule differentials between the system's design and BIT design and testing. For example, design considerations relating to a *systems breakdown structuring (SBS)*, such as partitioning and sub-system/assembly/component configuration, determine to a large extent the required BIT design. BIT design is also driven by the essential prediction of various system failure modes in an FMEA, which BIT is expected to address. Consequently, the two design efforts cannot be conducted in isolation from one another, and must therefore be concurrent.

Determination of basic system failure modes and frequency of occurrence The design of BIT is based upon two assumptions regarding the integrity of the basic engineering design: first, accurate identification of failure modes and effects (FMEA) and, second, correct estimation of the frequency of occurrence of the failure modes.

If either of these assumptions is proven incorrect by test or operational experience, the resultant BIT performance is likely to be inadequate or, at least, less effective than anticipated. The following two situations, based on the previous example, will illustrate the impact of FMEA and of the frequency of occurrence of the failure modes on a maintenance strategy (i.e. preventive versus corrective maintenance):

Situation 1:

An unforeseen failure mode is observed in the radar installation every 250 operational hours. What impact does this have on the no/BIT maintenance?

$$\begin{aligned} \text{New failures} &= 2{,}500\,\text{h} \times 1 \text{ failure per } 250\,\text{h} \\ &= 10 \text{ failures (new)} \\ \text{TC (no/BIT)}_{\text{new}} &= 10 \times 5 \text{ hours/failure} \\ &= 50\,\text{h} \end{aligned}$$

Thus, total maintenance hours will be:

$$\begin{aligned} \text{TC (total)} &= 80 + 60 + 50 \\ &= 190\,\text{h} \end{aligned}$$

Total no/BIT maintenance will be:

$$\begin{aligned} \text{TC (no/BIT)}_{\text{total}} &= 60 + 50 \\ &= 110\,\text{h} \end{aligned}$$

$\text{TC (no/BIT)}_{\text{total}}$ represents 58% of total maintenance.

For the BIT detected/isolated maintenance:

$$\begin{aligned}\text{TC (w/BIT)} &= 80\,\text{h} \\ &= 42\%\ \text{of total (190 h)}\end{aligned}$$

TC (w/BIT) represents 42% of total maintenance.

It is evident that the discovery of one unforeseen, no/BIT detectable failure has a relatively significant impact on the comparable magnitude of the two maintenance percentages.

Previous estimate:

$$\begin{aligned}\text{TC (w/BIT)} &= 57\% \\ \text{TC (no/BIT)} &= 43\%\end{aligned}$$

Current estimate:

$$\begin{aligned}\text{TC (w/BIT)} &= 42\% = 26\%\ \text{decrease} \\ \text{TC (no/BIT)} &= 58\% = 35\%\ \text{increase}\end{aligned}$$

Situation 2:

One of the original BIT detectable failures is predicted to have a very low frequency of occurrence. BIT detection for this failure was considered unnecessary, and was therefore not included in the original BIT design to detect 90% of the failures. It is now found that the failure occurs five times as often as expected. This is a realistic situation, and one that directly impacts upon the no/BIT maintenance hours.

d) Evaluation of BIT Systems

The test and evaluation of BIT systems and the prediction of BIT performance present some controversy. BIT systems are hardware and software logic networks designed to detect the presence of an unwanted signal, or the absence of a desired signal, each representing a failure mode. Each failure mode is detected by a specific logic-network. While the same network may be designed to detect a specific failure in several components, there is no assurance that the logic is correct until verified by testing. It is possible to validate BIT performance using statistical techniques, assuming a sufficiently large, representative sample of failures is available. Unlike typical reliability evaluation, though, which has been established over the past five decades, BIT testing and BIT system design represent less established technologies and are only recently beginning to receive increased attention. This limited attention has resulted in the lack of gathering an adequate representative database needed to support accurate and defendable estimates of BIT performance. A certain lack of confidence in BIT performance evaluation has therefore resulted because of these circumstances. Since it is not economically feasible to wait for an engineering

system to experience random failures, failures are induced through *synthetic fault insertion*. These faults are generally selected from a list of possible faults, all of which are presumed to be detectable. The faults are synthetically inserted, and BIT detects and isolates, for example, 93% of these. This does not mean that the BIT system is a 93% AFIC BIT system, because the data are not a representative random sample of the entire failure population and, therefore, cannot be used to make statistically valid predictions of future performance. While synthetic fault insertion has certain recognised limitations in predicting operational BIT performance, it is a valuable methodology in designing for maintainability during the preliminary and detail engineering design phases. Also, fault insertion can be used to simulate random failures that may occur but cannot be detected. These include effects of poor operation or maintenance.

Because of the lack of adequately established BIT technologies, requiring use of fault insertion, there are normally insufficient data available to support accurate estimations of BIT performance. It generally requires several years of operational exposure to develop an adequate database to support a BIT performance analysis. Current trends support early reliability testing during design and development, to facilitate identification of failure modes and timely incorporation of design improvements. These pilot tests provide a database to support preliminary estimates of system reliability. What is most frequently overlooked is that these data, after minimal screening, could also be used to monitor, verify and upgrade BIT performance, to support preliminary estimates of system maintainability—assuming that the BIT system is functional at the appropriate stage in the basic system's design and development. This action requires a disciplined approach towards the use of BIT in failure detection early in the system's life cycle that has not been prevalent in previous engineering design projects (DoD 3235.1-H 1982).

In summary, there is an essential requirement to evaluate BIT performance during the system design and development stages, inclusive of initial operational test and evaluation (IOT&E). This includes combining random failure detection data with data from pilot plant tests and fault insertion trials. Early emphasis on BIT design will generally result in accelerated BIT system establishment and more accurate early projections of BIT performance. BIT evaluation should be actively pursued throughout the ramp-up/operational stages, to assure that the necessary software and hardware changes are incorporated.

4.2.3.4 Specific Application Modelling of Availability and Maintainability

When considering a system that is not only in one of the two standard states of operability, i.e. an up-state (the system is capable of full operational performance) or a down-state (the system is totally inoperable and under repair); but may also perform its function at one or more levels of *reduced efficiency*, the conventional concepts of system integrity are found to be unsuitable and inadequate. The integrity of the system remains unresolved (there exist situations when the system is

neither fully operable nor fully inoperable, so that reliability and availability cannot be discretely determined), or it gets a value that contradicts empirical observation.

If operation with reduced efficiency is regarded as normal, too high a value for system integrity (reliability and availability) is obtained, whereas if a reduction in efficiency is regarded as *not* achieving total operability, too low a value for system integrity is obtained.

a) Equivalent Availability (EA)

The concept of *equivalent availability* affords a means of determining system integrity when the system is operating with *reduced efficiency* and is neither fully operable nor fully inoperable. From the definition of *operational availability* given previously, the general measure of *availability* of a system as a ratio is a comparison of the system's *usable time* or *operational time*, to a total given period or cycle time

$$\text{Availability} = \frac{\text{Operational Time}}{\text{Time Period}} \,. \tag{4.124}$$

To be able to relate system operation with *reduced efficiency* to an integrity measure such as system *availability* (specifically to the concept of *equivalent availability*), it is necessary to first review the relationships of the various process functional characteristics with one another, such as maximum capacity, rated capacity, efficiency, utilisation and availability.

Thus, referring back to Eq. (4.28), the efficiency measurement of an engineering process is a comparison of the *process output* quantity to its *process throughput*

$$\text{Process efficiency } (X_\text{p}) = \frac{\text{Process output}}{\text{Process throughput}} \,. \tag{4.125}$$

According to Eq. (4.30), process utilisation is the ratio of process output to the *constrained* ability to receive and/or hold the result or product inherent to the process (i.e. *rated capacity*)

$$\text{Process utilisation } (U_\text{p}) = \frac{\text{Process output}}{\text{Rated Capacity}} \,. \tag{4.126}$$

The *maximum* ability to receive and/or hold the result of the process, or product inherent to the process, is expressed as maximum process capacity or *design capacity*. According to Eq. (4.20), this is defined in terms of the *average output rate* and the *average utilisation rate* expressed as a percentage

$$\text{Max. Capacity } (C_\text{max}) = \frac{\text{Average Output Rate}}{\text{Average Utilisation}/100} \,. \tag{4.127}$$

Furthermore, *rated capacity* (C_r) is maximum throughput. It is the *throughput* actually achieved from operational constraints placed upon the ability of a series of

operations to receive and/or hold the result or product inherent to the process. Referring back to Eq. (4.23), we have

$$\text{Rated capacity } (C_\text{r}) = \frac{\text{Material in process}}{\text{Processing time}} = \text{Process throughput } \left(T^\text{C}_\text{proc}\right)_\text{max} \tag{4.128}$$

Maximum dependable capacity is achieved when a process system is operating at 100% *utilisation* or at *maximum efficiency* for a given *operational time*.

A system's *maximum dependable capacity* is equivalent to *process output* at 100% *utilisation*.

Thus

$$\text{Output (100\% utilisation)} = \text{Max. dependable capacity} \tag{4.129}$$

The *operational time* during which a system is achieving a *process output* that is equivalent to its *maximum dependable capacity* is termed the *equivalent operational time*.

Equivalent operational time is defined as "*that operational time during which a system achieves process output which is equivalent to its maximum dependable capacity*"

$$\text{Equiv. Operational time} = \text{Process Operational time} \times \frac{\text{Process output}}{\text{Max. Dependable Capacity}}\,. \tag{4.130}$$

If

$$\text{Process output (100\% utilisation)} = \text{Max. dependable capacity}$$

then

$$\text{Equiv. operational time} = \text{Process operational time}\,.$$

From Eq. (4.123), the general measure of *availability* of a system (or equipment) as a ratio is a comparison of the system's *operational time* to a total given period. Similarly, the quantifiable measure of *equivalent availability* of a system is a comparison of the system's *equivalent operational time* to a total given period. The system's process operational time is equal to the equivalent operational time when its process output (at 100% utilisation) is equal to the maximum dependable capacity or, alternatively, when its process output is equal to the rated capacity (and rated capacity = maximum dependable capacity).

From Eqs. (4.14) and (4.15), the difference between process utilisation, U_p, and process efficiency, X_p, is the difference between a system's rated capacity and process throughput respectively. From Eqs. (4.126) and (4.128), rated capacity C_r is equivalent to maximum throughput. Thus, at 100% process utilisation, a system's rated capacity is equal to maximum process throughput, and 100% process utilisation is equivalent to maximum efficiency.

Equivalent availability can be defined as "*the comparison of the equipment's equivalent operational time to a total given period, during which a system achieves*

process output that is equivalent to its maximum dependable capacity" (Nelson 1981). Thus

$$\text{Equivalent Availability} = \frac{\text{Equivalent Operational Time}}{\text{Time Period}} \tag{4.131}$$

$$\text{EA} = \frac{\sum(\text{ET}_\text{o})}{T}$$

$$\text{Equivalent Availability} = \frac{\text{Operational Time}}{\text{Time Period}} \times \frac{\text{Process output}}{\text{MDC}} \tag{4.132}$$

$$\text{EA} = \frac{\sum[(T_\text{o}) \cdot n(\text{MDC})]}{T \cdot \text{MDC}}$$

where:

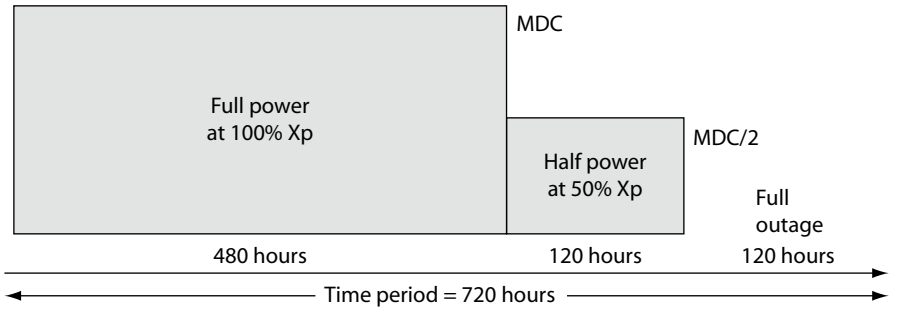
$\text{MDC} =$ maximum dependable capacity
$n =$ fraction of process output.

Thus

$$\text{Equivalent Availability (EA) at 100\% utilisation or max. efficiency} = \frac{\text{Operational Time}}{\text{Time Period}}\,.$$

The measure of *equivalent availability* can be graphically illustrated in the following example. A power generator is estimated to be in operation for 480 h at maximum dependable capacity. Thereafter, its output is estimated to diminish (derate) with an efficiency reduction of 50% for 120 h, after which the generator will be in full outage for 120 h. What is the expected *availability* of the generating plant over the 30-day cycle?

What is the generator's expected *equivalent availability* during this cycle?



Measure of *equivalent availability* of a power generator

$$\text{Expected Availability (A)} = \frac{\text{Operational Time}}{\text{Time Period}} = \frac{(\sum T_\text{o})}{T}$$

$$= (480 + 120)/720$$

$$= 0.83 \text{ or } 83\%$$

$$\begin{aligned}\text{Equiv. Availability (EA)} &= \frac{\text{Operational Time}}{\text{Time Period}} \times \frac{\text{Process Output}}{\text{MDC}} \\ &= \frac{\sum[(T_o) \cdot n(\text{MDC})]}{T \cdot \text{MDC}} \\ &= \frac{[480 \times (1)] + [120 \times (0.5)]}{720 \times (1)} \\ &= 0.75 \text{ or } 75\%\end{aligned}$$

where:

$$\begin{aligned}\text{Total time period} &= 720\,\text{h} \\ \text{Operational time} &= (480 + 120) = 600\,\text{h} \\ \text{MDC} &= \text{maximum dependable capacity} \\ \text{MDC} &= 1 \times (\text{constant representing capacity, } C) \\ \text{Process output} &= [0.75/(600/720)][(1) \times C] \\ \text{Process output} &= 0.9C \\ \text{Process output} &= 90\% \text{ of MDC.}\end{aligned}$$

b) Equivalent Maintainability Measures of Downtime and Outage

It is necessary to consider *mean downtime (MDT)* compared to the *mean time to repair (MTTR)*. There is frequently confusion between the two and it is important to understand the difference.

Downtime, or *outage*, is the period during which equipment is in the *failed state.* Downtime may commence before *repair*, as indicated in Fig. 4.11 (Smith 1981). This may be due to a significant time lapse from the onset of the downtime period up till when the actual repair, or corrective action, commences.

Repair time may often involve *checks* or *alignments* that may extend beyond the downtime period. From the diagram, it can be seen that the combination of *downtime* plus *repair time* includes aspects such as realisation time, access time, diagnosis time, spare parts procurement, replacement time, check time and alignment time. *MDT* is thus the *mean* of all the *time periods* that include *realisation, access, diagnosis, spares acquisition and replacement or repair*.

A comparison of *downtime* and *repair time* is given in Fig. 4.11.

According to the American Military Standard (MIL-STD-721B), a *failure* is defined as "*the inability of an item to function within its specified limits of performance*". Furthermore, the definition of *function* was given as "*the work that an item is designed to perform*", and *functional failure* was defined as "*the inability of an item to carry-out the work that it is designed to perform within specified limits of performance*".

From these definitions, it is evident that there are two degrees of severity of functional failure:

- A *complete loss of function*, where the item cannot carry out any of the work that it was designed to perform.

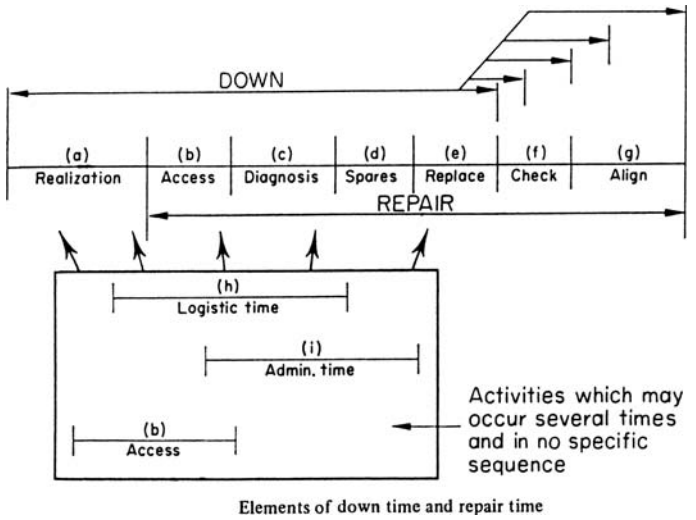


Fig. 4.11 A comparison of downtime and repair time (Smith 1981)

- A *partial loss of function*, where the item is unable to function within specified limits of performance.

In addition, equipment *condition* was defined as "*the state of an item on which its function depends*" and, as described before, the state of an item on which its function depends can be both an *operational* as well as a *physical condition*.

An important principle in determining the integrity of engineering design can thus be discerned relating to the expected *condition* and the required condition assessment (such as BIT) of the designed item:

> *An item's operational condition is related to the state of its operational function or working performance, and its physical condition is related to the state of its physical function or design properties.*

Equipment in a *failed state* is thus equipment that has an *operational* or *physical condition* that is in such a state that it is unable to carry out the work that it is designed to perform within specified limits of performance. Thus, two levels of severity of a *failed state* are implied:

- Where the item cannot carry out any of the work that it was designed to perform, i.e. a *total loss of function*.
- Where the item is unable to function within specified limits of performance, i.e. a *partial loss of function*.

Downtime, or outage, which has been described as the period during which equipment is in the *failed state*, has by implication two levels of severity, whereby the term *downtime* is indicative of the period during which equipment cannot carry out any of the work that it was designed to perform, and the term *outage* is indicative of

the period during which equipment is unable either to carry out any of the work that it was designed to perform or to function within specified limits of performance.

Downtime can be defined as "*the period during which an equipment's operational or physical condition is in such a state that it is unable to carry-out the work that it is designed to perform*".

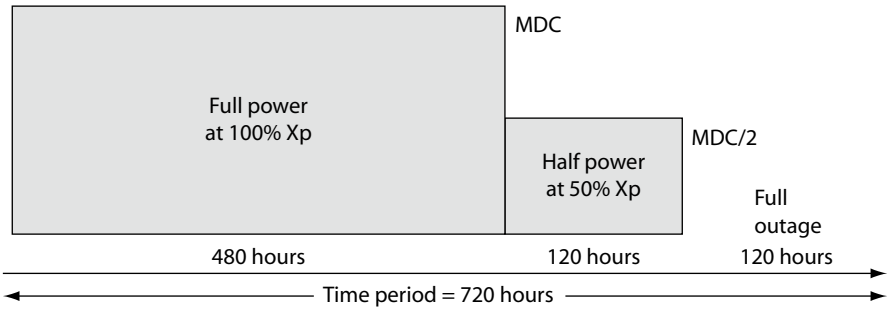
Outage can be defined as "*the period during which an equipment's operational or physical condition is in such a state that it is unable to carry-out the work that it is designed to perform within specified limits of performance*".

It is clear that the term *outage* encompasses *both a total loss of function and a partial loss of function*, whereas the term *downtime* constitutes a *total loss of function*. Thus, the concept of *full outage* is indicative of a *total loss of function*, and the concept of *partial outage* is indicative of a *partial loss of function*, whereas *downtime* is indicative of a *total loss of function* only. The concepts of full outage and partial outage are significant in determining the *equivalent mean time to outage* and the *equivalent mean time to restore*.

The equivalent mean time to outage (EM) Equivalent mean time to outage can be defined as "*the comparison of the equipment's operational time, to the number of full and partial outages over a specific period*"

$$\text{Equivalent Mean Time to Outage (EM)} = \frac{\text{Operational Time}}{\text{Full and Partial Outages}} . \qquad (4.133)$$

The measure of *equivalent mean time to outage* can be illustrated using the previous example. As indicated, the power generator is estimated to be in operation for 480 h at maximum dependable capacity, MDC. Thereafter, its output is estimated to derate, with a production efficiency reduction of 50% for 120 h, after which it will be in full outage for 120 h. What is the expected *equivalent mean time to outage* of the generator over a 30-day cycle?



Measure of *equivalent mean time to outage* of a power generator

$$\text{EM} = \frac{\sum(T_o)}{N} = \frac{480 + 120}{2} = 300 \text{ h} . \qquad (4.134)$$

The significance of the concepts of *full outage* and *partial outage*, being indicative of a total and a partial loss of function of individual systems, is that it enables the determination of the equivalent mean time to outage of *complex integrations of systems*, and of the effect that this complexity would have on the availability of engineered installations as a whole.

The equivalent mean time to restore (ER) It has previously been shown that the restoration of a failed item to an operational effective condition is normally when *repair action*, or *corrective action* in *maintenance* is performed in accordance with prescribed standard procedures. The item's operational effective condition in this context is also considered to be the item's *repairable condition*.

Mean time to repair (MTTR) in relation to equivalent mean time to restore (ER) The *repairable condition* of equipment is determined by the *mean time to repair* (MTTR), which is a measure of its *maintainability*

$$\begin{aligned} \text{MTTR} &= \text{Mean Time To Repair} \\ &= \frac{\sum(\lambda R)}{\sum(\lambda)} \end{aligned} \tag{4.135}$$

where:

$\lambda =$ failure rate of components
$R =$ repair time of components (h).

In contrast to the *mean time to repair* (MTTR), which includes the rate of failure at *component* level, the concept of *equivalent mean time to restore* (ER) takes into consideration the *equivalent lost time* in outages at *system* level, measured against the number of full and partial outages. This is best understood by defining *equivalent lost time*.

Equivalent operational time was previously defined as "*that operational time during which a system achieves process output which is equivalent to its maximum dependable capacity*".

In contrast, *equivalent lost time* is defined as "*that outage time during which a system loses process output, compared to the process output which is equivalent to the maximum dependable capacity that could have been attained if no outages had occurred*".

Furthermore, it was previously shown that the *maximum dependable capacity* (MDC) is reached when the system is operating at maximum *efficiency* or, expressed as a percentage, when the system is operating at 100% *utilisation* for a given *operational time*, i.e. *process output* at 100% *utilisation* is equivalent to the system's *maximum dependable capacity*

$$\begin{aligned} \text{Equivalent Lost Time} &= \frac{\text{Lost Output} \times \text{Operational Time}}{\text{Production Output at MDC}} \\ \text{ELT} &= \frac{\sum[n(\text{MDC}) \cdot T_o]}{\text{MDC}} \end{aligned} \tag{4.136}$$

where:

$n =$ fraction of process output.

Equivalent mean time to restore (ER) can be defined as "*the ratio of equivalent lost time in outages, to the number of full and partial outages over a specific period*".

If the definition of equivalent lost time is included, then *equivalent mean time to restore* can further be defined as "the ratio of that outage time during which a system loses process output compared to the process output which is equivalent to the maximum dependable capacity that could have been attained if no outages had occurred, to the number of full and partial outages over a specific period". Thus

$$\text{Equivalent Mean Time to Restore} = \frac{\text{Equivalent Lost Time}}{\text{No. of Full and Partial Outages}}$$

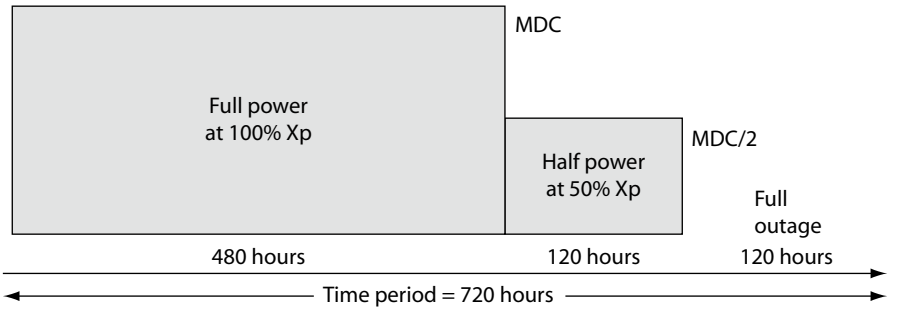
$$\text{ER} = \frac{\text{ELT}}{N} \tag{4.137}$$

$$\text{ER} = \frac{\sum[n(\text{MDC}) \cdot T_\text{o}]}{\text{MDC} \cdot N} \tag{4.138}$$

where:

n = fraction of process output
N = number of full and partial outages
T_o = outage time equal to lost operational time.

The measure of *equivalent mean time to restore* can be illustrated using the previous example. As indicated, the power generator is estimated to be in operation for 480 h at maximum dependable capacity. Thereafter, its output is estimated to diminish (derate), with a production efficiency reduction of 50% for 120 h, after which the plant will be in full outage for 120 h. What is the expected *equivalent mean time to restore* of the generating plant over the 30-day cycle?



Measure of *equivalent mean time to restore* of a power generator

$$\begin{aligned}\text{ER} &= \frac{\sum[n(\text{MDC})\cdot T_\text{o}]}{\text{MDC}\cdot N}\\ &= \frac{[0.5(\text{MDC})\times 120]+[1(\text{MDC})\times 120]}{\text{MDC}\times 2}\\ &= 90\text{ h}\,.\end{aligned}$$

c) Outage Measurement with the Ratio of ER Over EM

Outage measurement includes the concepts of *full outage* and *partial outage* in determining the ratio of the equivalent mean time to restore (ER) and the equivalent mean time to outage (EM). The significance of the ratio of equivalent mean time to outage (EM) over the equivalent mean time to restore (ER) is that it gives the measure of system *unavailability*, U.

In considering unavailability (U), the ratio of ER over EM is evaluated at system level where

$$\begin{aligned}\text{ER} &= \frac{\sum[n(\text{MDC})\cdot T_\text{o}]}{\text{MDC}\cdot N} \qquad (4.139)\\ \text{EM} &= \frac{\sum(T_\text{o})}{N}\\ \frac{\text{ER}}{\text{EM}} &= \frac{\sum[n(\text{MDC})\cdot T_\text{o}]}{\text{MDC}\cdot N}\cdot\frac{N}{\sum(T_\text{o})}\\ \frac{\text{ER}}{\text{EM}} &= \frac{\sum[n(\text{MDC})\cdot T_\text{o}]}{\text{MDC}\cdot\sum(T_\text{o})}\,.\end{aligned}$$

Expected availability (A), or the general measure of *availability* of a system as a ratio, was formulated as a comparison of the system's *usable time* or *operational time*, to a total given period or cycle time

$$A = \frac{(\sum T_\text{o})}{T}\,. \qquad (4.140)$$

If the ratio of ER over EM is multiplied by the availability of a particular system (A system) over a period T, the result is the sum of full and partial outages over the period T, or system unavailability, U

$$\begin{aligned}(A)\text{system}\cdot\frac{\text{ER}}{\text{EM}} &= \frac{\sum[n(\text{MDC})\cdot T_\text{o}]}{\text{MDC}\cdot\sum(T_\text{o})}\cdot\frac{(\sum T_\text{o})}{T} \qquad (4.141)\\ &= \frac{\sum[n(\text{MDC})\cdot T_\text{o}]}{\text{MDC}\cdot T}\\ &= \text{Unavailability }(U)\text{ system.}\end{aligned}$$

Thus, equivalent availability (EA) is equal to the ratio of the equivalent mean time to restore (ER) and the equivalent mean time to outage (EM), multiplied by the expected availability (A) over the period T.

Thus, the formula for equivalent availability (EA) can be given as:

$$\begin{aligned} \mathrm{EA} &= \frac{\sum[n(\mathrm{MDC}) \cdot T_\mathrm{o}]}{\mathrm{MDC} \cdot T} \\ \frac{\mathrm{ER}}{\mathrm{EM}} \cdot A &= \frac{\sum[n(\mathrm{MDC}) \cdot T_\mathrm{o}]}{\mathrm{MDC} \cdot \sum(T_\mathrm{o})} \cdot \frac{(\sum T_\mathrm{o})}{T} \\ \frac{\mathrm{ER}}{\mathrm{EM}} \cdot A &= \frac{\sum[n(\mathrm{MDC}) \cdot T_\mathrm{o}]}{\mathrm{MDC} \cdot T} \\ &= \mathrm{EA} \end{aligned}$$

So far, the equivalent mean time to outage (EM) and the equivalent mean time to restore (ER) have been considered from the point of view of outages at system level. However, the concepts of *full outage* being indicative of a *total loss of system function*, and *partial outage* being indicative of a *partial loss of system function*, and their significance in determining EM and ER make it possible to consider outages of individual systems within a complex integration of many systems, as well as the effect that an outage of an individual system would have on the availability of the systems as a whole. In other words, the effect of reducing EM and ER in a single system within a complex integration of systems can be determined from an evaluation of the *changes* in the *equivalent availability* of the systems (engineered installation) as a whole.

The effect of single system improvement on installation equivalent availability
The extent of the complexity of integration of individual systems in an engineered installation relative to the installation's hierarchical levels can be determined from the relationship of equivalent availability (EA) and unavailability (U) for the individual systems, and installation as a whole

$$\mathrm{EA\ system} = \frac{\mathrm{ER}}{\mathrm{EM}} \cdot A\ \mathrm{system} = \frac{\mathrm{ER}}{\mathrm{EM}} \cdot \frac{(\sum T_\mathrm{o})}{T} = U\ \mathrm{system} \qquad (4.142a)$$

$$\mathrm{EA\ install.} = \frac{\mathrm{ER}}{\mathrm{EM}} \cdot A\ \mathrm{install.} = \frac{\mathrm{ER}}{\mathrm{EM}} \cdot \frac{(\sum T_\mathrm{o})}{T} = U\ \mathrm{install.} \qquad (4.142b)$$

In this case, the ratio ER/EM would be the ratio of the equivalent mean time to restore (ER) over the equivalent mean time to outage (EM) of the *individual systems* that are included in the installation. If the installation (or process plant) had only one inherent system in its hierarchical structure, then the relationship given above would be adequate. Thus, the effect of improvement in this system's ER/EM ratio on the *equivalent availability* of the installation that consisted of only the one inherent system in its hierarchical structure can be evaluated. Based on outage data of the system over a period T, the *baseline* ER/EM ratio of the system can be determined. Similarly, improvement in the system's outage would give a new or *future* value for the system's ER/EM ratio, represented as:

$$\frac{\mathrm{ER}}{\mathrm{EM\ baseline}} \text{ and } \frac{\mathrm{ER}}{\mathrm{EM\ future}}\ .$$

The *change* in the equivalent availability (A) of the engineered installation, which consists of only the one inherent system in its hierarchical structure, can be formulated as

$$\Delta\text{EA install.} = \left[\frac{\text{ER}}{\text{EM baseline}}\frac{-\text{ER}}{\text{EM future}}\right]\cdot\frac{T_\text{o}\text{ install.}}{T}\,. \tag{4.143}$$

If the engineered installation consists of several integrated systems, then the ratio ER/EM would need to be modified to the following

$$\Delta\text{EA install.} = \sum_{j=1}^{q}\left[\frac{\text{ER}_j}{\text{EM}_j}\cdot A\text{ install.}\right] \tag{4.144}$$

$$\Delta\text{EA install.} = \sum_{j=1}^{q}\left[\frac{\text{ER}_j}{\text{EM}_j}\cdot\frac{T_\text{o}\text{ install.}}{T}\right]$$

where:

q = number of systems in the installation
ER_j = equivalent mean time to restore of system j
EM_j = equivalent mean time to outage of system j
T_o = operational time of the installation
T = evaluation period.

The effect of multiple system improvement on installation equivalent availability The *change* in the equivalent availability (A) of the engineered installation, which consists of multiple systems in its hierarchical structure, can now be formulated as

$$\Delta\text{EA install.} = \left[\sum_{j=1}^{q}\frac{\text{ER}_j}{\text{EM}_j\text{ baseline}} - \sum_{k=1}^{r}\frac{\text{ER}_k}{\text{EM}_k\text{ future}}\right]\cdot\frac{T_\text{o}\text{ install.}}{T} \tag{4.145}$$

where:

q = number of systems in the engineered installation
ER_j baseline = equivalent mean time to restore of system j
EM_j baseline = equivalent mean time to outage of system j
r = number of improved systems in the installation
ER_k future = equivalent mean time to restore of system k
EM_k future = equivalent mean time to outage of system k
T_o = operational time of the engineered installation
T = evaluation period.

This *change* in the equivalent availability (A) of the engineered installation, as a result of an improvement in the performance of multiple systems in the installation's hierarchical structure, offers an analytic approach in determining which systems are critical in complex integrations of process systems. This is done by determining the optimal *change* in the equivalent availability of the engineered installation

through an iterative process of marginally improving the performance of each system, through improvements in the equivalent mean time to restore of system *k*, and the equivalent mean time to outage of system *k*. The method is, however, computationally cumbersome without the use of algorithmic techniques such as *genetic algorithms* and/or *neural networks*.

Another, perhaps simpler approach to determining the effects of *change* in the equivalent availability of the engineered installation, and determining which systems are critical in complex integrations of process systems, is through the methodology of *systems engineering analysis*. This approach is considered in detail in Sect. 4.3.3.

As an example, consider a simple power-generating plant that is a multiple integrated system consisting of three major systems, namely #1 turbine, #2 turbine and a boiler, as illustrated in Fig. 4.12 below.

Statistical probabilities can easily be calculated to determine whether the plant would be up (producing power) or down (outage). In reality, the plant could operate at intermediate levels of rated capacity, or output, depending on the nature of the

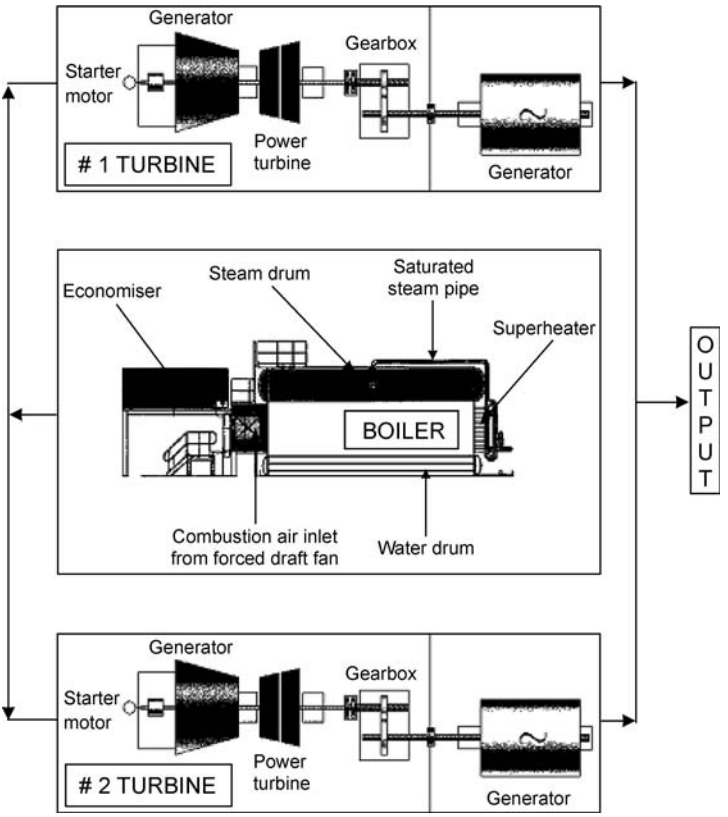


Fig. 4.12 Example of a simple power-generating plant

Table 4.1 Double turbine/boiler generating plant state matrix

State	Boiler	#1	#2	Capacity
1	Up	Up	Up	100%
2	Up	Up	Down	50%
3	Up	Down	Up	50%
4	Down	Up	Up	0%
5	Down	Up	Down	0%
6	Down	Down	Up	0%
7	Down	Down	Down	0%
8	Up	Down	Down	0%

outages of each of the three systems. This notion of plant *state* is indicated in Table 4.1, in which the outages are regarded as *full outages*, and no *partial outages* are considered.

Referring back to Eq. (4.20), *maximum process capacity* was measured in terms of the average output rate and the average utilisation rate expressed as a percentage

$$\text{Maximum Capacity } (C_{\max}) = \frac{\text{Average Output Rate}}{\text{Average Utilisation}/100} \tag{4.146}$$

$$\text{Average Output Rate} = (C_{\max}) \cdot \text{Average Utilisation}\,.$$

A system's *maximum dependable capacity* (MDC) was defined in Eq. (4.129) as being equivalent to *process output* at 100% *utilisation*. Thus

$$\text{MDC} = \text{Output (100\% utilisation)} \tag{4.147}$$

The plant's average output rate can now be determined where individual system outages are regarded to be *full outages*, and no *partial outages* are taken into consideration. The plant is in state 1 if all the sub-systems are operating and output is based on 100% utilisation (i.e. MDC). Seven other states are defined in Table 4.1, which is called a *state matrix*.

However, to calculate the expected or average process output rate of the *plant* (expressed as a percentage of maximum output at maximum design capacity), the percentage capacity for each state (at 100% utilisation) is multiplied by the *availabilities* of each integrated system.

Thus:

Average plant output rate with full outages only $= \Sigma$ *(capacity of plant state at 100% utilisation of systems that are operational* $\times$ *availability of each integrated system).*

As an example: what would be the expected or average output of the plant if the estimated boiler availability is 0.95 and the estimated turbine generator availabilities are 0.9 each?

$$\begin{aligned}\text{Average output rate} &= [1 \times (0.95 \times 0.9 \times 0.9)] \\ &\quad + [0.5 \times (0.95 \times 0.9 \times 0.1)] \\ &\quad + [0.5 \times (0.95 \times 0.1 \times 0.9)] \\ &= 0.7695 + 0.04275 + 0.04275 \\ &= 0.855 \\ &= 85.5\%\end{aligned}$$

It must be noted that this average output rate is expressed as a percentage of the possible output that can be achieved at maximum design capacity and 100% utilisation. It is thus an expression of the output capability of the plant as a whole, depending on the percentage capacity for each state of the plant, as a result of the states of each individual system, multiplied by its *availability*.

The above example of the simple three-system plant is insightful as an introduction to a plant having several states in which outages are regarded as *full outages*. If, however, the system outages are such that, over a specific period of time T, the systems could experience *full outages* as well as *partial outages* that are limited to 50% of system output, then a table similar to the previous *state matrix* can be developed.

To calculate the expected or average output rate of the plant over an operating period T, the percentage capacity (% of MDC) for each state during the shift period T is multiplied by the *equivalent availability* of each system. The *partial state matrix* given in Table 4.2 presents the *possible* states of each system over a period of time T. These states are either 100% down (full outage) or 50% down (partial outage).

Table 4.2 Double turbine/boiler generating plant partial state matrix

Period T	Period T	In period T	In period T	% MDC
1	Up	Up	Up	100%
2	Up	Up	Down 50%	75%
3	Up	Up	Down 100%	50%
4	Up	Down 50%	Up	75%
5	Up	Down 100%	Up	50%
6	Down 50%	Up	Up	50%
7	Down 100%	Up	Up	0%
8	Down 50%	Up	Down 50%	50%
9	Down 50%	Up	Down 100%	50%
10	Down 50%	Down 50%	Up	50%
11	Down 50%	Down 100%	Up	50%
12	Down 50%	Down 50%	Down 50%	50%
13	Down 50%	Down 50%	Down 100%	25%
14	Down 50%	Down 50%	Down 50%	25%
15	Down 50%	Down 100%	Down 100%	0%
16	Down 100%	Down 100%	Down 100%	0%

If, for simplicity, the likelihood and duration of each state in the above partial state matrix table is considered to be equal, and each state could occur only once in the operating period T, then according to Eq. (4.141) given above, the *equivalent availability* of each system can be calculated as:

$$\mathrm{EA} = \frac{\sum[(T_\mathrm{o}) \cdot n(\mathrm{MDC})]}{T \cdot \mathrm{MDC}} \, .$$

In this case:

T_o = operational period for each state
T = total operating period or cycle
$n(\mathrm{MDC})$ = capacity as % of MDC for each *system*
n = fraction of process output
MDC = maximum demand capacity.

Thus, the *equivalent availability* for the boiler in period T (i.e. working *down* the second column of Table 4.2), and using Eq. (4.141), can be calculated as:

$$\begin{aligned}\mathrm{EA}_\mathrm{Boiler} &= \frac{(1.5)[5\times 1+0.5+0+8\times 0.5+0]}{24\times 1} \\ &= \frac{(1.5)9.5}{24} \\ &= 0.59375 = 59.375\% \, .\end{aligned}$$

Similarly, the *equivalent availability* for #1 turbine and #2 turbine can also be calculated.

The *equivalent availability* for the #1 turbine in period T:
(i.e. working *down* the third column of Table 4.2)

$$\begin{aligned}\mathrm{EA}_{\#1\ \mathrm{Turbine}} &= \frac{(1.5)[7\times 1+4\times 0.5+5\times 0]}{24\times 1} \\ &= \frac{(1.5)9}{24} \\ &= 0.5625 = 56.25\% \, .\end{aligned}$$

The *equivalent availability* for the #2 turbine in period T:
(i.e. working *down* the fourth column of Table 4.2)

$$\begin{aligned}\mathrm{EA}_{\#2\ \mathrm{Turbine}} &= \frac{(1.5)[7\times 1+4\times 0.5+5\times 0]}{24\times 1} \\ &= \frac{(1.5)9}{24} \\ &= 0.5625 = 56.25\% \, .\end{aligned}$$

With the system *equivalent availabilities* for the boiler, #1 turbine and #2 turbine now calculated from all the possible partial states (up, 50% down, or 100% down), what would be the expected or average plant output rate when the equivalent availability for the boiler is 59.375% and the equivalent availability for the turbine generators are 56.25% each?

Taking into consideration states with reduced utilisation as a result of *partial outages*, the expected or average plant output rate is calculated as:

Average plant output rate with partial outages $= \Sigma$ *(capacity of plant state at full and partial utilisation of systems that are operational* $\times$ *availability of each integrated system)*

$$
\begin{aligned}
&= [1.0 \times (0.59375 \times 0.5625 \times 0.5625)] + [0.75 \times (0.59375 \times 0.5625 \times 0.5625)] \\
&\quad + [0.5 \times (0.59375 \times 0.5625 \times 0.4375)] + [0.75 \times (0.59375 \times 0.5625 \times 0.5625)] \\
&\quad + [0.5 \times (0.59375 \times 0.4375 \times 0.5625)] + [0.5 \times (0.59375 \times 0.5625 \times 0.5625)] \\
&\quad + [0.5 \times (0.59375 \times 0.5625 \times 0.5625)] + [0.5 \times (0.59375 \times 0.5625 \times 0.4375)] \\
&\quad + [0.5 \times (0.59375 \times 0.5625 \times 0.5625)] + [0.5 \times (0.59375 \times 0.4375 \times 0.5625)] \\
&\quad + [0.5 \times (0.59375 \times 0.5625 \times 0.5625)] + [0.25 \times (0.59375 \times 0.5625 \times 0.4375)] \\
&\quad + [0.25 \times (0.59375 \times 0.4375 \times 0.5625)] \\
&= 0.18787 + (2 \times 0.14090) + (4 \times 0.09393) + (4 \times 0.07306) + (2 \times 0.04697) \\
&= 0.18787 + 0.2818 + 0.37572 + 0.29224 + 0.09394 \\
&= 85.5\%
\end{aligned}
$$

The expected or average plant output rate, taking into consideration states with reduced utilisation as a result of *partial outages*, is 85.5%.

4.3 Analytic Development of Availability and Maintainability in Engineering Design

Several techniques are identified for availability and maintainability prediction, assessment and evaluation, in the conceptual, preliminary and detail design phases respectively. As with the analytic development of reliability and performance of Sect. 3.3, only certain of the availability and maintainability techniques have been considered for further development. This approach is adopted on the basis of the transformational capabilities of the techniques in developing intelligent computer automated methodology using *optimisation algorithms (OA)*. These optimisation algorithms should ideally be suitable for application in *artificial intelligence-based (AIB) modelling*, in which development of *knowledge-based expert systems* within a *blackboard model* can be applied in determining the integrity of engineering design. Furthermore, the *AIB* model must be suited to applied *concurrent engineering design* in an integrated *collaborative design* environment in which automated continual design reviews may be conducted during the engineering design process by remotely located design groups communicating via the internet.

4.3.1 Analytic Development of Availability and Maintainability Prediction in Conceptual Design

A technique selected for further development as a tool for *availability and maintainability prediction* in determining the integrity of engineering design during the *conceptual design* phase is modelling based on *measures of system performance*. This technique has already been considered in part for reliability prediction in Sect. 3.3.1, and needs to be expanded to include prediction of reliability *as well as* inherent availability and maintainability. System *performance analysis* through the technique of *simulation modelling* is also considered, specifically for prediction of system characteristics that affect system availability. Furthermore, the technique of *robust design* is selected for its preferred application in *decision-making* about engineering design integrity, particularly in considering the various uncertainties involved in system performance simulation modelling.

Monte Carlo simulation is used to propagate these uncertainties in the application of simulation models in a *collaborative engineering design* environment. The techniques selected for availability and maintainability prediction in the conceptual design phase are thus considered under the following topics:

i. System performance measures and limits of capability
ii. System performance analysis and simulation modelling
iii. Uncertainty in system performance simulation modelling.

4.3.1.1 System Performance Measures and Limits of Capability

Referring back to Sect. 3.3.1, it was pointed out that, instead of using actual performance values such as temperatures, pressures, etc., it is more meaningful to use the *proximity* of the actual performance value to the *limit of capability* of the item of equipment. In engineering design review, the proximity of performance to a limit closely relates to a measure of the item's *safety margin*, which could indicate the need for design changes or selecting alternate systems. Non-dimensional numerical values for system performance may be obtained by determining the limits of capability, $C_{\max}$ and $C_{\min}$, with respect to the performance parameters for system integrity (i.e. reliability, availability, maintainability and safety). The nominal range of integrity values for which the system is designed (i.e. 95 to 98% reliability at 80 to 85% availability) must also be specified. Thus, a set of data points are obtained for each item of equipment with respect to the relevant performance parameters, to be entered into a *parameter performance matrix*.

a) Performance Parameters for System Integrity

For predicting system availability, the performance measures for reliability and maintainability are estimated, and the *inherent availability* determined from these

measures. As indicated previously, system reliability can be predicted by estimating the mean time between failures (MTBF), and maintainability performance can be predicted by estimating the mean time to repair (MTTR).

Inherent availability, A_i, can then be predicted according to:

$$A_i = \frac{\text{MTBF}}{(\text{MTBF} + \text{MTTR})} \tag{4.148}$$

In the case of reliability and maintainability, there are no *operating* limits of capability but, instead, a prediction of engineering design performance relating to MTBF and MTTR. Data points for the parameter performance matrix can be obtained through *expert judgement* of system reliability by estimating the mean time between failures (MTBF), and of maintainability performance by estimating the mean time to repair (MTTR) of critical failures (Booker et al. 2000). (Refer to Sect. 3.3.3.4 dealing with expert judgement as data.)

A *reliability* data point x_{ij} can be generated from the predicted MTBF(R), a maximum acceptable MTBF($R_{\max}$), and a minimum acceptable MTBF($R_{\min}$), where:

$$x_{ij} = \frac{(R - R_{\min}) \times 10}{R_{\max} - R_{\min}} \tag{4.149}$$

Similarly, a maintainability data point x_{ij} can be generated from the predicted MTTR(M), a minimum acceptable MTTR($M_{\min}$), and a maximum acceptable MTTR($M_{\max}$), where:

$$x_{ij} = \frac{(M_{\max} - M) \times 10}{M_{\max} - M_{\min}} \tag{4.150}$$

b) Analysis of the Parameter Profile Matrix

The performance measures of a system can be described in matrix form in a *parameter profile matrix* (Thompson et al. 1998). The matrix is compiled containing data points relating to all salient parameters that describe a system's performance. The rows and columns of the matrix can be analysed in order to predict the characteristics of the designed system. Figure 3.22 is reproduced below as Fig. 4.13 with a change to the column heading from *process systems* to *equipment items*, as a single system is being considered.

Consider one row of the matrix. Each data point x_{ij} refers to a single performance parameter; looking along a row reveals whether the system design is consistently good with respect to this parameter for all the system's equipment items, or whether there is a variable performance. For a system with a large number of equipment items (in other words, a high level of complexity), a good system design should have a high mean and a low standard deviation of x_{ij} scores for each parameter. These scores are calculated as indicated in Figs. 3.23, 3.24 and 3.25 of Chap. 3, Sect. 3.3.1. Furthermore, a *parameter performance index* (PPI) that constitutes an

	Equipment items					
	$\mathbf{x}_{11}$	$\mathbf{x}_{12}$	$\mathbf{x}_{13}$	$\mathbf{x}_{14}$	...	$\mathbf{x}_{1i}$
	$\mathbf{x}_{21}$	$\mathbf{x}_{22}$	$\mathbf{x}_{23}$	$\mathbf{x}_{24}$	...	$\mathbf{x}_{2i}$
Performance	$\mathbf{x}_{31}$	$\mathbf{x}_{32}$	$\mathbf{x}_{33}$	$\mathbf{x}_{34}$	...	$\mathbf{x}_{3i}$
parameters	$\mathbf{x}_{41}$	$\mathbf{x}_{42}$	$\mathbf{x}_{43}$	$\mathbf{x}_{44}$	...	$\mathbf{x}_{4i}$
	...	...	...	...	...	...
	$\mathbf{x}_{j1}$	$\mathbf{x}_{j2}$	$\mathbf{x}_{j3}$	$\mathbf{x}_{j4}$	...	$\mathbf{x}_{ji}$

Fig. 4.13 Parameter profile matrix

analysis of the *rows* of the parameter profile matrix can be calculated(Thompson et al. 1998):

$$\text{PPI} = n\left(\sum_{j=1}^{n} 1/c_{ij}\right)^{-1} \tag{4.151}$$

where n is the number of design alternatives.

The inverse method of calculation of an overall score is advantageous when the range of scores is 0 to 10, as it highlights low scores, whereas a straightforward addition of scores may not reveal a low score if there are high scores in the group. However, the inverse calculation method is less sensitive to error than a multiplication method. The numerical value of PPI lies in the range 0 to 10, no matter how many data points are included in the calculation. Thus, a comparison can be made to judge whether there is acceptable overall performance with respect to all the parameters, or whether the system design is weak in any respect—particularly concerning the parameters of reliability, inherent availability, and maintainability.

A similar calculation to the *parameter performance index* can be made for each *column* of the parameter profile matrix, whereby an equipment or *device performance index* (DPI) is calculated as (Thompson et al. 1998):

$$\text{DPI} = m\left(\sum_{j=1}^{m} 1/c_{ij}\right)^{-1} \tag{4.152}$$

where m is the number of performance parameters relating to the equipment item of column j.

A comparison of DPIs reveals those equipment items that are contributing less to the overall performance of the system. For an individual equipment item, a good design is a high mean value of the x_{ij} scores with a low standard deviation. This system performance prediction method is intended for complex systems comprising many sub-systems. Overall system performance can be quite efficiently determined, as a wide range of system performance parameters can be included in the PPI and DPI indices. However, in the case of reliability, inherent availability, and maintainability, only the two parameters MTTR and MTBF are included for prediction, as indicated in Eq. (4.148). From an engineering design integrity point of view, the method collates relevant design integrity data (i.e. reliability, inherent

availability, and maintainability) obtained from expert judgement predictions, and compares these with design performance criteria, requirements and expectations.

c) The Design Checklist

There are many *qualitative* factors that influence reliability, inherent availability, and maintainability. Some are closely related to operability, and there are no clear demarcation lines. In order to expand design integrity prediction, a study of the many factors that influence these parameters must be carried out. An initial list is first derived in the form of a *design checklist*. The results of previous research into reliability, availability and maintainability problems are carefully considered when devising the checklist questions (McKinney et al. 1989).

The checklist is intended for general application, and includes factors affecting design operability. In many cases, there will be questions that do not apply to the design being reviewed, which are then omitted.

The questions can be presented to the analyst in the form of a specific *knowledge-based expert system* within an *artificial intelligence-based (AIB) blackboard model* for design review during the design process. Results are presented in a format that enables design review teams to collaboratively make reasonable design judgements. This is important, in view of the fact that one design team may not carry out the complete design of a particular system as a result of multidisciplinary engineering design requirements, and the design review teams may not all be grouped at one location, prompting the need for *collaborative engineering design*. This scenario is considered later in greater detail in accounting for various uncertainties involved in system performance simulation modelling for engineering design, utilising the *robust design* technique. Furthermore, knowledge-based expert systems within AIB blackboard models are given in Sect. 3.4, Sect. 4.4, and Sect. 5.4.

A typical example of a checklist question set, extending from conceptual to schematic design, is the following:

Question set Is pressure release and drainage (including purging and venting) provided?

Are purge points considered? If there are no purge points, then this may mean drainage via some or other means that could increase exposure to maintenance personnel requiring the need for protection.

i. Purge points not present, requiring some other means 0
ii. Purge points present but accessibility will be poor 1
iii. Purge points present and accessibility will be good 2.

A series of questions is posed for each design, and each answer is given a score 0, 1 or 2. The total score is obtained by the summation of the scores, calculated as a percentage of the total of all the relevant questions. Therefore, questions that do not apply are removed from the analysis. The objective is to obtain overall design integrity ratings for the process and for each device, in order to highlight weak design integrity considerations. A high percentage score indicates good performance

where the scores complement the MTTR and MTBF calculations. Where there is a mismatch—for example, a high estimated MTBF but a low reliability score, or a high estimated MTTR but low maintainability score—then further design investigation is required.

d) Integrity Prediction of Common Items of Equipment

The prediction method is intended for those process systems that comprise many well-known items (as the design is still in its conceptual phase). It could be expected that certain items of equipment may exhibit common maintainability requirements. In order to save time in data estimates, typical maintenance requirements for common devices are prepared as data sheets.

Thus, if a centrifugal pump is selected, then a common data sheet would be available for this item. The data sheet can be reviewed and accepted as it is, or it can be edited to reflect certain particular circumstances, or the checklist can be completed from a blank form to compile a new set of data for that item. In addition to the responses to questions for each individual item, a response to each particular question regarding total systems integration may be considered for all relevant items. For example, in the case of maintainability, the question might refer to the ability to detect a critical failure condition. If the response to this question is estimated at 60%, then it would suggest that 40% of all items for which the question is relevant would remain undetected. It is thus possible to review the responses to questions across all the integrated systems, to gain an understanding of the integrity of the conceptual design as a whole.

e) Design Reviews of Performance Parameters for System Integrity

Design review practices can take many forms. At the lowest level, they consist of an examination of drawings before manufacture begins. More comprehensive design reviews include a review at different phases of the engineering design process: the specification (design requirements), conceptual design, schematic or preliminary design, and detail design. There are particular techniques that the design review team implement at these different phases, according to their suitability. The method proposed here is intended for use when very little detail of the equipment is known. In particular, it is intended for use during the conceptual design phase in preparation for the follow-up schematic design phase when systems are synthesised using manufactured equipment. Many engineered installations are designed in this way. Design reviews can be quantitative or qualitative. The advantage of *quantitative* reviews is that they present clearly a justification for a decision that a particular design is either satisfactory or unsatisfactory with respect to essential performance specifications.

Therefore, if it is possible, there are advantages to a quantitative review as early as possible in the engineering design process. A design review should add value

to each design. Design calculation checks are taken care of by all good, traditional design practices; however, a good design review will be repaid by reduced commissioning and start-up problems and better ramp-up operations. The design review method proposed here seeks to provide a quantitative evaluation that adds value to engineering design by integrating process performance parameters such as mass flows, pressures and temperatures with reliability, availability and maintainability. Performance data required are the same as those used to specify the equipment. Therefore, there is no requirement for extra data in excess of those that would be available for process systems that comprise many well-known items. The only additional requirement is a value judgement of acceptable and unacceptable MTBF and MTTR. These data are then compiled into a parameter profile matrix using data points derived from the proximity of a required operating point to the performance limit of the equipment. As indicated previously, the use of the proximity of the nominal design performance to a limit of equipment capability is similar to the concept of a safety margin. Similarly, the estimated MTBF and MTTR data points reflect the closeness of predicted performance to expectations. Having compiled the matrix, analysis can be performed on particular variables for all items of equipment, or on all variables for a particular item of equipment, yielding PPI and DPI indices respectively.

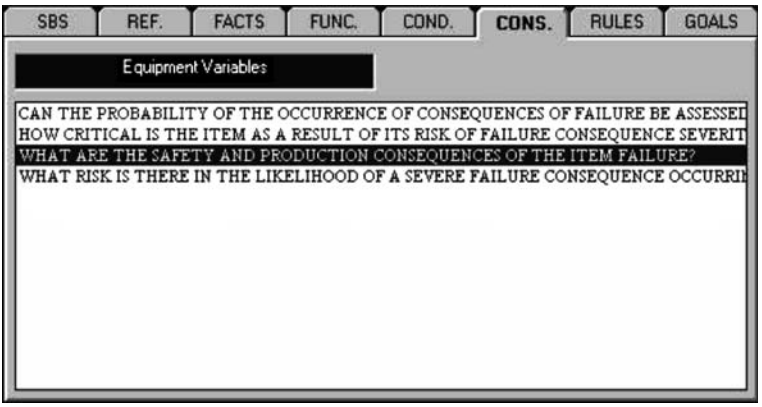
On a large engineering design project, data can be generated by several design teams, compiled and analysed using the *AIB blackboard model* for automated design reviews throughout the engineering design process. MTBF and MTTR expectations can be varied in a sensitivity analysis. The computer automated methodology can highlight matrix cells with low scores and pick out performance variables and equipment that show poor performance. Therefore, the data handling and calculation aspects of the design verification do not impose excessive requirements. The flexibility of the approach, and the method of data point derivation are especially useful in process engineering enhancement projects. Inevitably, there are instances when engineered installations are subject to modifications either during construction or even after ramp-up (e.g. there may be advantages in processing at different pressures and/or temperatures), necessitating review of the equipment performance data after design completion. Implications of changes to temperature and pressure requirements can be readily explored, since the parameter profile analysis method will immediately identify when the performance of an item is in close proximity to a limit.

Furthermore, engineered installations may be required to process materials that are different to those that they were originally designed to process. As the equipment data are already available in the *AIB blackboard model*, all that is needed is to input new process data for further analysis. The time taken to set up the equipment database during a post-design review will be justified, since the data can be used throughout the life of the design. Reliability, inherent availability, and maintainability are included in the parameter profile matrix evaluation by estimating MTBF and MTTR times, and then calculating the inherent availability. In order to expand on these important variables and to explore the various factors that influence reliability, inherent availability, and maintainability, checklists are developed that may be used

in different ways, either on their own or to complement the parameter profile analysis. In a design review, many of the questions may be answered to obtain a view of a system's integrity, or to obtain an overall view of the integrity of the integrated systems design. Another use for the checklists would be in a process enhancement study whereby an existing engineered installation is audited to identify precisely the improvements required.

f) Reliability and Maintainability Checklists

The checklists are designed for use by engineering designers in the case of a design review exercise, or by process engineers in the case of required post-design process enhancements. Although the question sets listed in the *AIB blackboard models* presented in Sects. 3.4, 4.4 and 5.4 are somewhat different from the example checklists for reliability and maintainability, the relevant principles and intended use remain the same. A segment of an AIB blackboard model Expert System tab page is given below, showing a particular question set.



The following question sets indicate the general content of the checklists (Thompson et al. 1998).

Reliability checklist

Q1. Is the component a single unit, active redundant or stand-by redundant?
Q2. Are the demands on the equipment short, medium or long pulses?
Q3. Does the duty cycle involve any thermal, mechanical or pressure shocks?
Q4. Is the pH of the process fluid high or low?
Q5. Is the component used for high, medium or low flow rates?
Q6. Is the component in a physical situation where external corrosive attack can occur from: open weather, other machinery being submerged?
Q7. Are solids present in the process fluid?
Q8. Are gases present?
Q9. Is the fluid of a viscosity that is likely to increase loads on the equipment?

Q10. Are sharp bends, causing forceful impingement, present?
Q11. Are stagnant zones present that may hold the process fluid after flushing?
Q12. How complex is the equipment?
Q13. Are alignment/adjustment procedures needed on installation/replacement?
Q14. Is any special equipment required to make the adjustments?
Q15. Do components have many state changes (e.g. opening/closing of valves)?
Q16. Is the equipment novel in design or application?
Q17. Do components have arduous sealing duties?
Q18. Are special materials used?

Maintainability checklist

Q1. Will catastrophic failure be evident in the control room?
Q2. Will degraded failure be evident from the control room?
Q3. Time period of degraded failure detection?
Q4. Does maintenance require protective clothing due to hazardous substances or hot equipment, or does the equipment need time to cool down?
Q5. How easy is it to isolate equipment?
Q6. What method of isolation is required?
Q7. What area of plant needs to be isolated?
Q8. Is pressure release and drainage (including purging and venting) provided?
Q9. Is electrical isolation of equipment required?
Q10. Is scaffolding required for maintenance?
Q11. Can scaffolding be erected by maintenance personnel or by contractors?
Q12. Is there adequate space to build scaffolding?
Q13. Is there adequate space to manoeuvre while maintenance is taking place?
Q14. How is the equipment lifted?
Q15. Whatever lifting equipment is used, are there any problems foreseen?
Q16. Does other equipment need to be removed before access can be gained?
Q17. Is visual access to the fault good enough to carry out maintenance?
Q18. Is the physical access good enough to carry out maintenance?

4.3.1.2 System Performance Analysis and Simulation Modelling

Section 3.3.1.2 considered system performance within the context of *designing for availability*, which can be perceived as the combination of:

- a system's *process capability* (with regard to the process characteristics of *capacity*, *input*, *throughput*, *output* and *quality*);
- a system's *functional effectiveness* (with regard to the functional characteristics of *efficiency* and *utilisation*);
- a system's *operational condition* (with regard to operational measures such as *temperatures*, *pressures*, *flows*, etc.).

All these characteristics may serve as useful indicators in designing for availability whereby system performance *simulation modelling* is generally considered the

most appropriate methodology for predicting their integrated–interactive values. In this case, simulation modelling has been found to be an effective tool for analysing a large quantity of interrelated and compound variables in predicting a complex system design's process capability, functional effectiveness and operational condition. Simulation modelling has been applied in determining the performance of complex integrated systems design in Sect. 4.4.

System performance analysis is concerned with the study of the behaviour of a system in terms of its *measurable characteristics*. System performance analysis techniques can be applied in determining the performance characteristics of proposed designs, and to identify those areas of the design where performance problems may be experienced. It is focused on determining how systems behave under certain conditions, and can be used to compare different system designs to evaluate their relative merits in terms of achieving the required design criteria. However, questions relating to assurance of the integrity of a proposed design are not always included in the scope of system performance analysis. A design that is acceptable from a performance-related viewpoint may be unacceptable from an integrity point of view; similarly, a design that meets integrity requirements such as reliability and safety may not be acceptable from a system performance standpoint. System performance analysis is a multidisciplinary field, covering many areas. Among these are parameter performance matrices, evolutionary operation, experimental design, queuing theory, modelling techniques and dynamic simulation.

System performance analysis in engineering design is concerned with some of these modelling techniques, in particular simulation modelling and its application to the study of the performance of systems based on process characteristics that affect system availability. In most engineering systems, there are a significant number of performance characteristics and technical constraints involved in their design. When the interactions between all of the characteristics and constraints are considered, it becomes clear that these interactions are usually numerous and complex. The behaviour of the whole system cannot easily be predicted by the application of relatively simple algorithms, as might be expected for less complex systems based on a few process characteristics. In complex process engineering designs, it is often not totally obvious where the bottlenecks may occur, and what the determining factors behind system performance might be. Thus, the principle underlying the development of system performance models is that by capturing the essential real-world behaviour of a system in a mathematical or simulation model, valuable insight can be gained into its critical behaviour. Once a model of a system has been developed, verified and validated, it is possible to experiment with the model and to determine what the limiting factors in system performance are. This would then lead to possible modifications of the system's design to improve the performance measure of concern.

Development of a model would allow performance characteristics such as sizing, capacity, mass and energy balances, and functional response issues to be addressed at an early stage of an engineering process system's life cycle. In this way, potential performance problems are already identified at the conceptual phase of engineering

design, and designed out of the system prior to firming up design configurations and system specifications in the preliminary or schematic design phase. Without this approach, there is a real danger that the actual bottlenecks of the installed system will not be identified. In the absence of the evidence that a system performance model may provide, it is quite likely that significant amounts of resources could be spent later in 'improving' inherent items of the installed system that have been found to constrain its performance.

System performance modelling provides a relatively inexpensive way of exploring the performance implications of different system design configurations. Although the effort involved in a major modelling project should not be underestimated, the potential savings that can be made from avoiding redesign and/or rework when a system fails to meet its performance objectives will more than justify the cost.

Thus, from an engineering design perspective, it becomes essential not only to understand the dynamic behaviour of complex or integrated systems, in addition to formulating their expected performance characteristics, but also to ensure that the design meets both the performance objectives as well as the necessary integrity constraints.

a) Types of System Performance Models

System performance models can be broadly classified as either *analytic models* or *simulation models*. Analytic models rely on formulae to represent the behaviour of system components. If such formulae exist, then their solution is likely to be fairly concise. However, in many cases formulae do not exist or are valid only under restrictive conditions. Historically, analytic models have yielded only average behaviour patterns, and have not given insight into the likely distribution of expected values. The use of analytic techniques to find underlying distributions in the case of uncertainty in predicting essential process characteristics has extended the range of engineering design problems that may be solved (Law et al. 1991).

For design problems that can be solved using these techniques, analytic models are ideal. However, the *integration* of analytic models representing individual systems, each with process characteristics and performance constraints, is not trivial. To obtain maximum benefit, these models must link together common process characteristics and related system performance constraints, such that they provide an accurate representation of the design's intended integration of systems. In many cases, it will not be possible to solve the analytic model to find the appropriate distribution of expected values. Mean-value predictions will be limited, since a much larger number of factors affect the behaviour of a complex integration of systems. In such cases, *system performance simulation modelling* is most appropriate (Emshoff et al. 1970).

b) System Simulation Modelling

There are two main types of simulation modelling, specifically:

- *Continuous-time simulation model*
- *Discrete-event simulation model*.

In the first type of model, *continuous-time simulation model*, time-related activity is perceived to be continuous. This type of simulation is appropriate for continuous engineering process situations such as modelling the concentrations of chemicals in a reactor vessel. These concentrations will vary smoothly with time (at a fine enough timescale) and, at each instant of time, the reaction will be proceeding at a certain rate.

In the second type of model, *discrete-event simulation model*, time-related events can be distinguished as fundamental entities and, from a modelling perspective, no points in time other than those at which events happen need to be considered. This type of model is well suited to modelling production systems or industrial processes where not only the events are discrete entities but they can take on discrete probability distributions (Shannon 1975; Bulgren 1982).

Simulation models attempt to derive the overall behaviour of a system either by representing the behaviour of each component of the system separately, and specifying how the components interact with each other, or by representing the behaviour of the system as a whole and specifying how the process characteristics interact with each other. Thus, variables of a simulation model may change in any of four ways (Emshoff et al. 1970):

- In a discrete manner at any point in time.
- In a continuous manner at any point in time.
- In a discrete manner only at certain points in time.
- In a continuous manner only at discrete points in time.

In engineering design, it is common albeit not correct to use the term ‘discrete’ to describe a system with constant periodic time steps, where the term refers to the time interval and not to discrete events during the time interval. For *discrete system simulations*, input is introduced into the model as a set of discrete items arriving either randomly or at specified intervals. The individual components then react accordingly, and the overall behaviour of the model can be measured (Bulgren 1982). Conversely, for *continuous system simulations* (or process modelling), a smooth flow of homogenous values is described, analogous to a constant stream of fluid passing through a pipe. The volume may increase or decrease but the flow is continuous. Changes in process characteristic values (i.e. inputs, throughputs, outputs, etc.) are based directly on changes in time, and time can change in equal increments. These values reflect the state of the performance of the modelled system at any particular time, which advances evenly from one time step to the next (Diamond 1997).

Although simulation models are used to predict the behaviour of the system(s) being modelled, their behaviour must be interpreted statistically. This necessitates either many different runs or extended run periods of the model of a given system, depending on the type of simulation modelling applied, to obtain a valid sample of the behaviour that the system is likely to exhibit. Compared to the use of analytic models, developing and interpreting system performance simulation models is a slow process but, nevertheless, definitely much cheaper than experimenting with real-world systems after they have been designed and installed (Law et al. 1991).

As stated previously, in producing a simulation model of a system design, the intent is to determine how that system will behave under various conditions. The structure of the simulation model must therefore monitor, and be sensitive to, the behaviour of the system arising from the interaction of a potentially large number of system items (i.e. sub-systems and assemblies), and/or the interaction of a wide range of variable performance characteristics (i.e. inputs, throughputs and outputs—or, in modelling terms, exogenous, status and endogenous variables respectively). It is thus best to adopt a holistic approach, considering all of the components and processes involved at a high systems hierarchy level. This means that the preferred application of system performance simulation modelling is at the conceptual engineering design phase, with further modelling refinements as the design progresses into the schematic or preliminary design phase. However, under a given set of conditions, a system will most likely be constrained by one particular item or a single performance characteristic—although this may vary depending on the set conditions. It is therefore essential to represent within the model as many of the items and/or performance characteristics in the system as possible, so that potential bottleneck effects can be determined. System items that are not close to being a bottleneck can be represented simply, since the fine detail of their behaviour is not likely to change much. At the conceptual design phase, all system items are represented simply so that some information can already be gleaned as to where potential bottlenecks might exist. The critical areas can then be refined to gain further insight into these bottlenecks. Clearly, if a system item or performance characteristic is *not* represented in the model, it can never be construed to be a constraint on the behaviour of the system. This somewhat undermines the benefit of developing a simulation model at the conceptual design phase, and also reduces its perceived usefulness. If the system's item or characteristic *is* represented, however, the model can be used to investigate how changes in the assumptions made about the item or characteristic affect the overall behaviour of the model, and the system.

The balance between detail and scope of system performance simulation modelling is evident—if the model has wide scope, then it can be extended only to a shallow depth in a given time; conversely, if the same effort is put into a narrow scope, then a greater depth of available modelling detail can be added. The aim of a system performance simulation modelling study should therefore be to initially identify uncertainties surrounding broad characteristics of the system's performance, and then to find those items that could place constraints on system behaviour.

4.3.1.3 Uncertainty in System Performance Simulation Modelling

In considering the various uncertainties involved in system performance simulation modelling for engineering design, the *robust design* technique is a preferred application in decision-making for design integrity. It is generally recognised that there will always be uncertainties in the design of any engineering system. This is due to variations in the performance characteristics not only of the individual system but in the complex integration of multiple systems as well. Besides possible algorithmic errors related to computer simulation model implementation, two general sources contribute to uncertainty in simulation model predictions of performance characteristics in engineering system designs (Du et al. 1999b):

- *External uncertainty*:
  External uncertainty comes from the variability in model prediction arising from alternative model variables (including both design parameters and design variables). It is also termed 'input parameter uncertainty'. Examples include the variability associated with process characteristics of *capacity*, *input*, *throughput* and *quality*, functional characteristics of *efficiency* and *utilisation*, operational conditions, material properties, and physical dimensions of constituent parts.
- *Internal uncertainty*:
  This type of uncertainty has two sources. One is due to the limited information in estimating the characteristics of model parameters for a given, fixed model structure, which is called 'model parameter uncertainty', and another type is in the model structure itself, including uncertainty in the validity of the assumptions underlying the model, referred to as 'model structure uncertainty'.

A critical issue in simulation modelling of an engineering design that comprises a complex integration of systems is that the effect of the uncertainties of one system's performance characteristics may propagate to another through linking model variables, resulting in the overall systems output having an accumulated effect of the individual uncertainties. A practical problem in large-scale systems design is that multidisciplinary groups often use predictive tools of varying accuracy to determine if the design options meet the design requirements, especially when performing impact analyses of proposed changes from other groups (Du et al. 1999b).

The inevitable use of multidisciplinary groups in large-scale systems design necessitates the application of *collaborative engineering design* as well as a careful study of the effect of various uncertainties as a part of design requirements tracking and design coordination. Two primary issues concerning uncertainty in simulation modelling of an engineering design that comprises a complex integration of systems, and thus an integration of multidisciplinary design teams, are:

- How should the effect of uncertainties be propagated across the systems?
- How should the effect of uncertainties be mitigated to make sound decisions?

Techniques for *uncertainty analysis* include the *statistical approach* and the worst-case analysis or *extreme condition approach* (Du et al. 1999c).

The *statistical approach* relies heavily on the use of data sampling to generate cumulative distribution functions (c.d.f.) of system outputs. Monte Carlo simulation, a commonly used random simulation-based approach, becomes expensive in simulations of complex integrations of systems (Hoover et al. 1989).

Reduced sampling techniques, such as the Latin hypercube sampling technique (Box et al. 1978) and Taguchi's orthogonal arrays technique (Phadke 1989), are used to improve computational efficiency, though they are not commonly applied in commercial simulation programs.

The *extreme condition approach* is to derive the range of system performance characteristics, such as process input, throughput or output, in terms of a range of uncertainties, by either sub-optimisations, first-order Taylor expansion or interval analysis (Chen et al. 1995).

Use of the statistical approach as well as of the extreme condition approach has been restricted to propagating the effect of *external uncertainty* only, prompting the need to accommodate more generic representations of *both* external and internal uncertainties. Furthermore, there are few examples associated with how to mitigate the effect of both the external and internal uncertainties in system performance simulation modelling of complex engineering designs. Relatively recent developments in design techniques have generated methods that can reduce the impact of potential variations by manipulating controllable design variables.

Taguchi's robust design is one such approach that emphasises reduction of performance variation through reducing sensitivity to the sources of variation (Phadke 1989). Robust design has also been used at the system level to reduce the performance variation caused by process characteristic deviations. The concept of robust design has been used to mitigate performance variations due to various sources of uncertainties in simulation-based design (Suri et al. 1999).

An integrated methodology for propagating and managing the effect of uncertainties is proposed. Two approaches, namely the extreme condition approach and the statistical approach, are simultaneously developed to propagate the effect of both external uncertainty and internal uncertainty across a design system comprising interrelated sub-system analyses. An uncertainty mitigation strategy based on the principles of *robust design* is proposed. A simplistic simulation model is used to explain the proposed methodology. The principles of the proposed methods can be easily extended to more complicated, multidisciplinary design problems.

a) Propagation of the Effect of Uncertainties

A simulation-based design model is used to explain the proposed methodology. The principles of the methodology are generic and valid for other categories of relationships between the system models. The design model consists of a chain of two simulation programs (assuming they are from two different disciplines) that are connected to each other through linking variables, as illustrated in Fig. 4.14 (Du et al. 1999c).

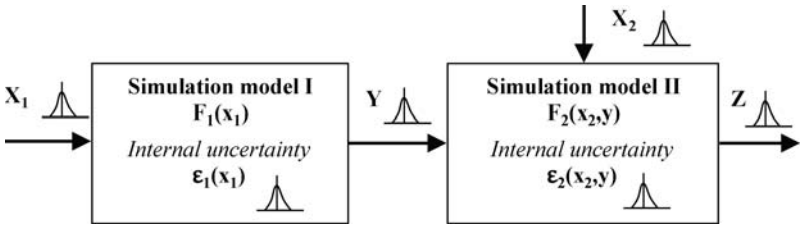


Fig. 4.14 Simulation-based design model from two different disciplines (Du et al. 1999c)

The linking variables are represented by the vector y. The input to the simulation model I is the vector of the design variable x_1 with uncertainty (external uncertainties describe by a range Δx_1, or certain distributions).

Due to the *external uncertainty* and the *internal uncertainty*, which is modelled as $\varepsilon_1(x_1)$ in simulation model I, the output vector of model I, which is given by the expression

$$y = F_1(x_1) + \varepsilon_1(x_1)$$

will have deviations Δy or described by distributions.

For simulation model II, the inputs are the linking variable vector y and the design variable vector x_2. Because of the deviations existing in x_2 and y, and the *internal uncertainty* $\varepsilon_2(x_2 y)$, associated with simulation II, the final output vector, given by the expression

$$z = F_2(x_2 y) + \varepsilon_2(x_2 y)$$

will also have deviations Δz or described by distributions.

For simulation model I, the output expression for y consists of the simulation model $F_1(x_1)$ and the corresponding error model of the internal uncertainty, $\varepsilon_1(x_1)$. For simulation model II, the inputs are the linking variable y and the design variable x_2. The output expression for z consists of the simulation model $F_2(x_2 y)$ and the corresponding error model of the internal uncertainty, $\varepsilon_2(x_2 y)$. The output vector z often represents system performance parameters that are used to model the design objectives and constraints. Because of the deviations existing in x_2 and y, and the internal uncertainty $\varepsilon_2(x_2 y)$, the final output z will also have deviations. The question is how to propagate the effect of various types of uncertainties across a simulation chain with interrelated simulation programs. Two approaches are proposed, first the extreme condition approach and, second, the statistical approach (Du et al. 1999c).

b) Extreme Condition Approach for Uncertainty Analysis

The extreme condition approach is developed to obtain the interval of extremes of the final output from a chain of simulation models. The term *extreme* is defined as

"*the minimum or the maximum value of the end performance (final output) corresponding to the given ranges of internal and external uncertainties*".

With this approach, the external uncertainties are characterised by the intervals $[x_1 - \Delta x_1, x_1 + \Delta x_1]$ and $[x_2 - \Delta x_2, x_2 + \Delta x_2]$. Optimisations are used to find the maximum and minimum (extremes) of the outputs from simulation model I, and simulation model II. The flowchart of the proposed procedure is illustrated in Fig. 4.15.

The steps to obtain the output z, $z_{\min}$, $z_{\max}$ are given as (Du et al. 1999c):

i) Given a set of nominal values x_1 and range Δx_1 for simulation model I, minimise (maximise) $F_1(x_1)$ and $\varepsilon_1(x_1)$ by selecting values from $[x_1 - \Delta x_1, x_1 + \Delta x_1]$ to obtain the values $F_{1\,\min}(x_1)$, $F_{1\,\max}(x_1)$, and $\varepsilon_{1\,\min}(x_1)$, $\varepsilon_{1\,\max}(x_1)$.

ii) The optimisation model is:
Given: the nominal value of x_1 and the range Δx_1
Subject to: $x_1 - \Delta x_1 \le x_1 \le x_1 + \Delta x_1$
Optimise: minimise $F_1(x_1)$ to obtain $F_{1\,\min}(x_1)$
maximise $F_1(x_1)$ to obtain $F_{1\,\max}(x_1)$.

iii) Obtain the extreme values of internal uncertainty $\varepsilon_{1\,\min}(x_1)$ and $\varepsilon_{1\,\max}(x_1)$ over the range $[x_1 - \Delta x_1, x_1 + \Delta x_1]$.

iv) Obtain the interval $[y_{\min}, y_{\max}]$ using:
$y_{\min} = F_{1\,\min}(x_1) + \varepsilon_{1\,\min}(x_1)$
$y_{\max} = F_{1\,\max}(x_1) + \varepsilon_{1\,\max}(x_1)$.

v) Given a set of nominal values x_2 and range Δx_2, for simulation model II, minimise (maximise) $F_2(x_2)$ and $\varepsilon_2(x_2)$ by selecting values from $[x_2 - \Delta x_2, x_2 + \Delta x_2]$ to obtain the values $F_{2\,\min}(x_2)$, $F_{2\,\max}(x_2)$, and $\varepsilon_{2\,\min}(x_2)$, $\varepsilon_{2\,\max}(x_2)$.

vi) The optimisation model is:
Given: the nominal value of x_2 and the range Δx_2

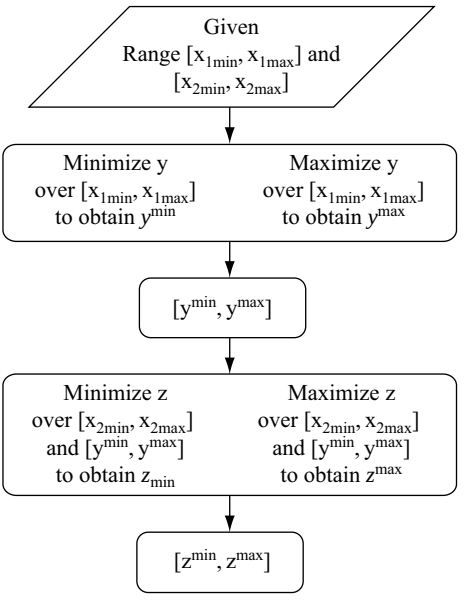


Fig. 4.15 Flowchart for the extreme condition approach for uncertainty analysis (Du et al. 1999c)

Subject to: $x_2 - \Delta x_2 \leq x_2 \leq x_2 + \Delta x_2$
Optimise: minimise $F_2(x_2)$ to obtain $F_{2\ \min}(x_2)$
maximise $F_2(x_2)$ to obtain $F_{2\ \max}(x_2)$.

vii) Obtain the extreme values of internal uncertainty $\varepsilon_{2\ \min}(x_2)$ and $\varepsilon_{2\ \max}(x_2)$ over the range $[x_2 - \Delta x_2, x_2 + \Delta x_2]$.

viii) Obtain the interval $[y_{\min}, y_{\max}]$ using:
$z_{\min} = F_{2\ \min}(x_2) + \varepsilon_{2\ \min}(x_2)$
$z_{\max} = F_{2\ \max}(x_2) + \varepsilon_{2\ \max}(x_2)$.

Based on the computed interval $[z_{\min}, z_{\max}]$, the *nominal value* of z is calculated as:

$$\dot{z} = \frac{[z_{\min} + z_{\max}]}{2} \tag{4.153}$$

The *deviation* of z can be calculated as:

$$\Delta z = [z_{\min} - z_{\max}] \tag{4.154}$$

The *nominal value* and *deviation* of a system output is based on given system input intervals.

The extreme condition approach identifies the interval of a system output based on the given intervals of the system inputs. It is applicable to the situation in which both the external uncertainties in x_1 and x_2 are expressed by ranges. Illustrated in Fig. 4.15 is the flowchart of the proposed procedure of using optimisations to find the maximum and minimum (extremes) of outputs from simulation model I and simulation model II, for the simulation-based design model from two different design disciplines given in Fig. 4.14. It depicts the procedure used to obtain the range of outputs z, $z_{\min}$, $z_{\max}$, as considered in steps i) to viii) above.

c) The Statistical Approach for Uncertainty Analysis

The statistical approach is developed to estimate cumulative distribution functions (c.d.f.) and probability density functions (p.d.f.), or population parameters (for example mean and variance) of the final outputs from a chain of simulation models. It is assumed that x_1 and x_2, and the internal uncertainty, $\varepsilon_1(x_1)$ and $\varepsilon_2(x_2 y)$, follow certain probabilistic distributions that may be obtained by field or experimental data, or information of similar existing processes, or by judgements by engineering experience.

Since the distribution parameters (i.e. mean and variance) of the uncertainty values $\varepsilon_1(x_1)$ and $\varepsilon_2(x_2 y)$ are functions of x_1, x_2 and y, the final distributions of $\varepsilon_1(x_1)$ and $\varepsilon_2(x_2 y)$ are the accumulated effects of both the uncertainty in the error model and the uncertainty of the external parameters such as x_1, x_2 and y.

Monte Carlo simulation methods are used to propagate the effect of uncertainties through the simulation chain. A flowchart of the Monte Carlo simulation procedure is given in Fig. 4.16 (Law et al. 1991).

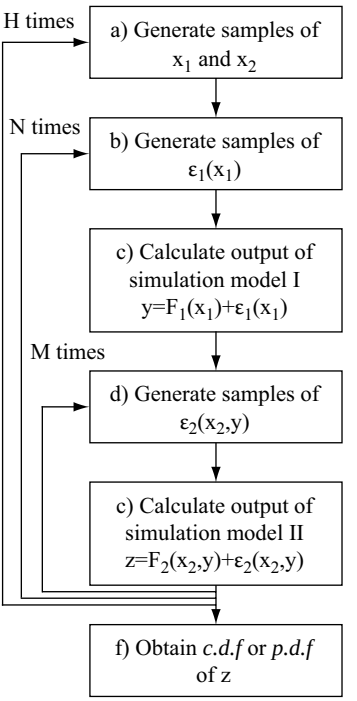


Fig. 4.16 Flowchart of the Monte Carlo simulation procedure (Law et al. 1991)

The Monte Carlo simulation approach generates statistical estimates of the system output based on the given distributions of the inputs and error models. This gives more information than does the extreme condition approach, by which only the best and worst performance are estimated. Because the statistical approach is based on Monte Carlo simulation, it often requires a large number of simulations. More effective sampling techniques such as the Latin hypercube and fractional factorial design can be used to reduce the amount of simulations (Hicks 1993).

The Monte Carlo simulation procedure is as follows (Law et al. 1991):

i) Generate H samples of x_1 and x_2 as simulation inputs based on distribution functions.
ii) For the given x_1, calculate the distribution parameters of the internal uncertainty $\varepsilon_1(x_1)$ for simulation model I, and generate N samples of the internal uncertainty ε_1 for simulation model I based on the distribution function.
iii) Evaluate the corresponding output $y = F_1(x_1) + \varepsilon_1(x_1)$ for simulation model I.
iv) For each y, calculate the distribution parameters of the internal uncertainty $\varepsilon_2(x_2 y)$ of simulation model II, and generate M samples of the internal uncertainty ε_2 based on the distribution function.
v) Evaluate the corresponding output $z = F_2(x_2) + \varepsilon_2(x_2)$ for simulation model II.
vi) Calculate the *mean value* μ_z, the *standard deviation* σ_z or the c.d.f. and p.d.f. of z based on $H \times M \times N$ samples of z.

d) Mitigating the Effect of Uncertainty

To assist designers to make reliable design decisions under uncertainties, the proposed techniques of propagating the effect of uncertainties is integrated with the multidisciplinary optimisation approach based on the principles of *robust design*, i.e. by extending the quality engineering concept to the mitigation of the effects of both external and internal uncertainties. From the viewpoint of robust design, the goal is to make the system (or product) least sensitive to the potential variations without eliminating the sources of uncertainty (Taguchi 1993).

The same concept is used here to reduce the impact of both external and internal uncertainties associated with the simulation programs. The robust optimisation objectives are achieved by simultaneously optimising mean performance and reducing performance variation, subject to the constraints brought about by their deviations. Taguchi's robust design has been used in the past for mitigating the effect of parameter uncertainty, which is similar to the external uncertainty considered here.

This concept is extended to mitigate the effect of model structure uncertainty (internal uncertainty). For the *extreme condition approach*, the robust design model can be formulated as:

Given: (4.155)
Parameter and model uncertainties (ranges)

Find:
Robust design decisions (x)

Subject to:
System constraints: $g_{\text{worst}}(x) \leq 0$

Objectives:
i) Optimise the mean of system attributes: $a(x)$
ii) Minimise the deviation of system attributes: $\Delta a(x)$.

In the above model, $g_{\text{worst}}(x)$ is the maximum constraint function estimated by the worst case of constraint function $g(x)$, and a is the objective vector. Both $g(x)$ and $a(x)$ are the subsets of system output vector z. The mean and deviation of the system outputs can be obtained by the extreme condition approach as introduced earlier. This constitutes the necessary multiple objectives in robust design (i.e. both the mean and the deviation of the system are expected to be minimised with the assumption that optimising the mean of a system attribute can always be transformed into a minimisation problem). The general form of the objective can be expressed as:

$$\min[ax, \Delta a(x)] \tag{4.156}$$

Many existing approaches can be used to solve this multi-objective robust optimisation problem. In the above model, the worst-case analysis is used to formulate the constraints. The worst-case analysis assumes that all fluctuations may occur simultaneously in the worst possible combination (Parkinson et al. 1993). The effect of

variations on a function is estimated using a first-order Taylor's series as follows:

$$\Delta g(x) = \sum_f \left| \frac{\partial g(x)\Delta(x)}{\partial x_1} \right| \tag{4.157}$$

where $\Delta g(x)$ represents the variation transmitted to constraint $g(x)$ for a worst-case analysis.

The design feasibility expressed in Eq. (4.155) can be formulated by increasing the value of the mean $g(x)$ by the functional variation $\Delta g(x)$:

$$g_{\text{worst}}(x) = \Delta g(x) + \sum_f \left| \frac{\partial g(x)\Delta x}{\partial x_1} \right| \tag{4.158}$$

For the *statistical approach* to estimate the performance distribution, the robust model can be formulated as:

Given: (4.159)
Parameter and model uncertainties (distributions)

Find:
Robust design decisions x

Subject to:
System constraints: $P[g(x) \leq 0] \geq P$ limit

Objectives:
i) Optimise the mean of system attributes $a(x)$: $\mu_a(x)$
ii) Minimise the standard deviation of system attributes $a(x)$: $\sigma_a(x)$.

$\mu_a(x)$ and $\sigma_a(x)$ are the estimates of the mean and variance of the system outputs respectively. The constraints in the above model are expressed by the probabilistic formulation. $P[g(x) \leq 0]$ is the probability of constraint satisfaction, and it should be greater than or equal to the defined probability limit P_{limit}.

Because it is computationally expensive to evaluate the probability of constraint satisfaction, alternative formulations—for example, the *moment matching method*—are used in practice to evaluate the constraints. With the moment matching method, $g(x)$ is assumed to follow a normal distribution (Parkinson et al. 1993).

The constraint in Eq. (4.159) is (Parkinson et al. 1993):

$$\mu_a(x) + k\sigma_a(x) \leq 0 \tag{4.160}$$

where k is the constant for the probability of constraint satisfaction.

For example, $k = 1$ stands for the probability $\approx$0.8413 and $k = 2$ stands for probability $\approx$0.9772.

Based on the previous considerations, the strategy that integrates the propagation and mitigation of the effect of uncertainties is summarised in Fig. 4.17. Module A is the uncertainty quantification module that represents the first stage in the integrated methodology. Module B is the propagation module. In this module, either the extreme condition approach or the statistical approach is used to identify the range or to estimate the population parameters of system performance under the influence of

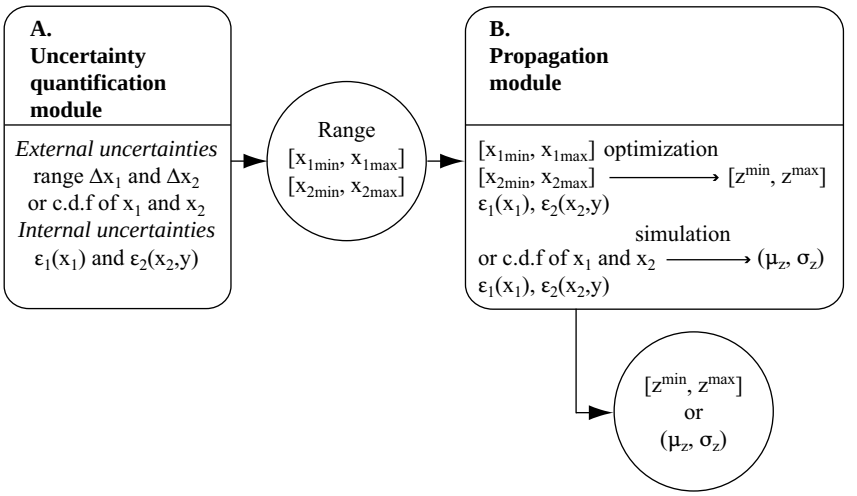


Fig. 4.17 Propagation and mitigation strategy of the effect of uncertainties (Parkinson et al. 1993)

both internal and external uncertainties. The performance ranges or estimated population parameters are then used to mitigate the effect of uncertainties. The basis for controlling the effect of uncertainties is the *robust design* approach formulated in Eqs. (4.155) and (4.159). The process to manage the effect of uncertainty is iterative and involves repeated uncertainty analysis until a robust optimal solution is obtained.

4.3.2 Analytic Development of Availability and Maintainability Assessment in Preliminary Design

Techniques selected for further development of *availability and maintainability assessment* to determine the integrity of engineering design in the *preliminary* or *schematic design* phase of the engineering design process include the application of *Petri nets (PN)*. The techniques selected are considered under the following topics:

i. Maximising design availability using Petri net models
ii. Designing for availability using Petri net modelling.

Designing for availability with preventive maintenance Analytic assessment of large complex process systems has increasingly become an integral part of the engineering design process, particularly in designing for availability and maintainability—and even more so with the inclusion of complex interactions, such as preventive maintenance on system availability. Preventive maintenance is considered as one of the key factors to increasing system reliability, availability and productivity,

and to reducing production costs. The importance of the inclusion of maintenance in engineering design has led to an increased sophistication in mathematical models required to analyse its impact on complex system behaviour (Lam et al. 1994).

A quantitative example of designing for availability with the inclusion of preventive maintenance is developed. The designed system starts in a working state, but ages with time and eventually fails if no preventive maintenance action is carried out. Preventive maintenance is performed at fixed intervals from the start-up of the system in the operational state. The preventive maintenance activity takes an exponentially distributed amount of time and is in the form of component renewal that is assumed will allow for full system performance. The preventive maintenance interval is thus a critical design parameter. If the interval approaches zero, the system is always under maintenance and availability drops to zero. Conversely, if the interval becomes too large, the beneficial effect of the preventive maintenance action becomes negligible. The goal of the example is to develop an analytic expression for the steady-state behaviour of a complex system using *Petri net (PN)* methodology, and to determine the optimal design maintenance interval that will maximise system availability.

4.3.2.1 Maximising Design Availability Using Petri Net Models

Petri net models have only recently gained widespread acceptance—they provide a graphical language ideally suited to modern CAD environments that can be concise in their specification; they provide a natural way to present complex logical interactions among integrated systems, or process activities within a system; and they are closer to a designer's intuition about what a complex systems model should look like (Peterson 1981; Murata 1989). Many structural and stochastic extensions have been proposed in the application of Petri nets to increase their modelling power and their capability to represent large, complex integrated systems. The most up-to-date and valuable source of references for the theoretical development and application of Petri net models is the series of international workshops known as Petri Nets and Performance Models (PNPM), initiated in Italy in 1985, and which moved to the USA, Japan, Australia and France in the following decades.

a) Petri Net Theory

Petri nets have been used as mathematical, graphical tools for modelling and analysing systems showing dynamic behaviours characterised by synchronous and distributed operation, as well as non-determinism (Peterson 1981). A basic Petri net structure consists of *places* and *transitions* interconnected by directed *arcs*. Places are denoted by circles and represent *conditions*, while transitions are denoted by bars and represent *events*. The directed arcs in a Petri net represent flow of control where the occurrence of events is controlled by a set of conditions.

In addition to its graphical structure, a Petri net is effectively used to simulate the dynamic behaviour of a modelled system in terms of states, or markings, and their changes during model execution. A *marking* is an assignment of *tokens* to the places, where a token denotes that the corresponding condition is true. Thus, the marking of places describes the current state of the Petri net in terms of the conditions that are true and those that are false.

The translation of a flowchart to a Petri net is illustrated in Fig. 4.18 where the nodes of the flowchart are replaced by transitions in the Petri net, and the arcs are replaced by places.

The Petri net execution changes the number and location of tokens according to a rule of transition enabling and firing (Murata 1989) where:

- a transition t is enabled if each input place p is marked with the tokens $w(p,t)$, where $w(p,t)$ is the weight of the arc from p to t;
- an enabled transition may or may not fire depending on whether or not the event actually takes place;
- an enabled transition t is fired by removing $w(p,t)$ tokens from each input place p and adding $w(t,p)$ tokens to each output place p.

Petri nets represent a powerful paradigm, useful for modelling complex systems in the context of systems performance, in designing for availability subject to preventive maintenance strategies that include complex interactions such as component renewal. Such interactions are time-related and dependent upon component age and estimated residual life. However, original Petri nets did not carry any notion of time. Thus, in order to make the technique useful for quantitative systems analysis in engineering design, a variety of time extensions have been proposed in the literature. The distinguishing features of these time extensions are whether the duration of the events should be modelled by deterministic or random variables, and whether the time is associated with process functions, or transitional events. Petri nets in which the timing is stochastic are referred to as *stochastic Petri nets (SPN)*, and the most common assumption is that time is assigned to the duration of transitional events. The time evolution of an SPN is expressed as a stochastic process, and referred to

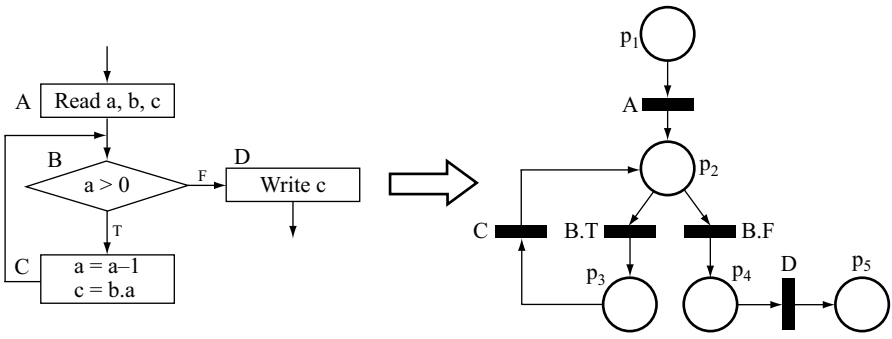


Fig. 4.18 Translation of a flowchart to a Petri net (Peterson 1981)

as its *marking process*. SPN can be used to automatically generate the underlying marking process, which can then be analysed to yield results in terms of the original Petri net model.

This is a case where a *user-level* representation of complex systems, typically in the form of simulation models (such as the *process equipment models (PEMs)* developed in Sect. 4.4), is translated into an analytic representation that is processed and the results referred back to the user-level representation.

The most common assumption in the literature is to assign to the PN transitions an exponentially distributed *firing time* (i.e. start to completion time of an activity), so that the resulting marking process is a *continuous-time Markov chain* (*CTMC*; Molloy 1982). Almost all the PN-based tools are based on this assumption. In principle, simple equations can be derived for both transient and steady-state analysis of CTMCs. However, practical limitations arise from the fact that the state space (i.e. the composition of different states of a system and its transition interaction of moving from state i to state j, including the probability of such a transition) grows much faster than the number of components in the system being modelled. The use of an exponentially distributed firing time has been regarded as a restriction in the application of PN models, as there are many engineering processes with times to occurrence that are not exponentially distributed. The hypothesis of exponential distributions in these cases results in the construction of models that give only a qualitative, rather than quantitative analysis of real systems. The existence of deterministic or other non-exponentially distributed events in engineering processes, such as start-up delays and pre-planned downtimes in real-time systems, gives rise to stochastic models that are non-Markovian in nature. In recent years, a considerable effort has been devoted to improving the PN methodology in order to deal with generally distributed events in real-time systems. However, the inclusion of non-exponential distributions affects the associated marking process (in that some or other retained memory of past events would then be required), and further specification is needed at the PN level in order to uniquely define how the marking process is conditioned on past history (Ciardo et al. 1994).

b) Definition of the Basic Petri Net Model

A marked PN is a tuple $\mathrm{PN} = (P, T, I, O, M)$ (Peterson 1981), where:

- $P = \{p_1, p_2, \ldots, p_n\}$ is the set of *places* (drawn as circles);
- $T = \{t_1, t_2, \ldots, t_n\}$ is the set of *transitions* (drawn as bars);
- I and O are the *input* and the *output functions* respectively;
- $M = \{m_1, m_2, \ldots, m_n\}$ is the *marking* of the PN.

The generic entry m_i is the number of *tokens* (drawn as black dots) in place p_i, in marking M. The initial marking is M_0. The input function I provides the multiplicities of the *input arcs* from functions to transitions; the output function O provides the multiplicities of the *output arcs* from transitions to functions. Input and output arcs have an arrowhead on their destination. A transition is enabled in a marking if

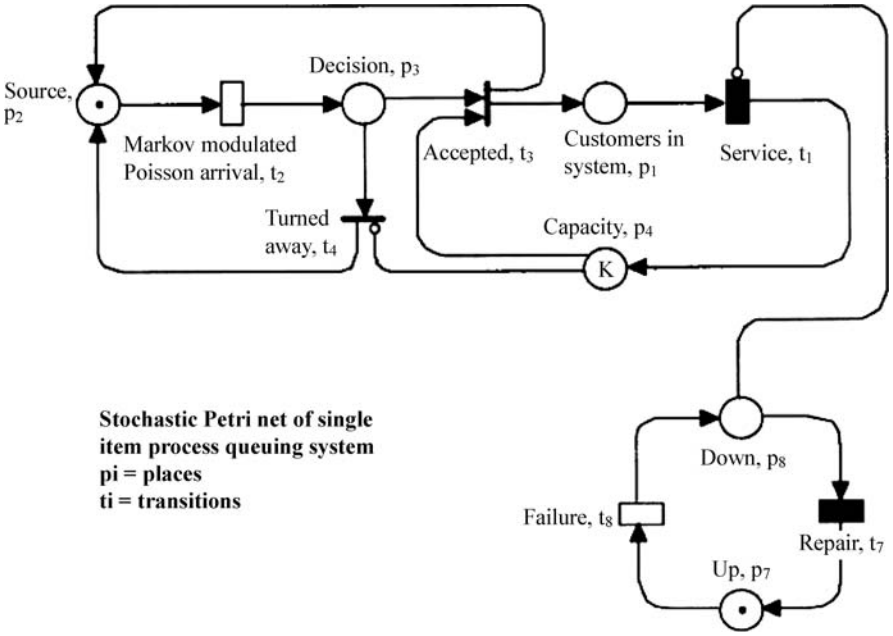


Fig. 4.19 Typical graphical representation of a Petri net (Lindemann et al. 1999)

each of its input places contains at least as many tokens as the multiplicity of the input function I. An enabled transition fires by removing as many tokens as the multiplicity of the input function f from each input place, and adding as many tokens as the multiplicity of the output function O to each output place. A marking M' is said to be directly reachable from M when it is generated from M by firing a single enabled transition t_k. The reachability set $R(M_0)$ is the set of all the markings that can be generated from an initial marking M_0 by repeated application of the above rule. The enabling of a transition corresponds to the starting of an activity, while the firing corresponds to the completion of an activity. Thus, the firing of a transition causes a previously enabled transition to become disabled. PNs can be used to capture the behaviour of many real-world situations, such as the typical PN given in Fig. 4.19 below (Lindemann et al. 1999):

Structural extensions Various structural extensions have been proposed in the literature to increase either the class of problems that can be represented, or the ability and the ease with which real systems can be modelled. The modelling power of a PN is the ability of the PN formalism to represent classes of problems. Modelling convenience is defined as the practical ability to represent a given behaviour in a simpler, more compact or more natural way. Decision power is defined to be the set of properties that can be analysed. Increasing the modelling convenience decreases the decision power. Thus, each possible extension to the basic formalism requires an in-depth evaluation of its effect upon modelling convenience and decision power (Peterson 1981).

Some extensions have proven so effective that they are now considered part of the standard PN definition. They are:

- Inhibitor arcs
- Transition priorities
- Marking-dependent arc multiplicity.

Inhibitor arcs connect a place to a transition and are drawn with a small circle on their destination. An inhibitor arc from a place p_i to a transition t_k disables t_k when p_i is not empty. It is possible to use an arc multiplicity extension together with inhibitor arcs. In this case, a transition t_k is disabled whenever place p_i contains at least as many tokens as the multiplicity of the inhibitor arc. The number of tokens in an inhibitor input is not affected by a firing operation.

Transition priorities are integer numbers assigned to the transitions. A transition is enabled in a marking if and only if no higher priority transitions are enabled. If this extension is introduced, some markings of the original PN may no longer be reachable.

Marking-dependent arc multiplicity was introduced with the intent to model situations in which the number of tokens to be transferred along the arcs (or to enable a transition) depends upon the system state. Arcs with marking-dependent multiplicity are indicated by a Z on the arc, and allow simpler and more compact PNs than would otherwise be possible. In many practical problems, their use can reduce the complexity of the PN model (Ciardo 1994).

c) Definition of Stochastic Petri Nets

The most common way to include time into a PN is to associate the time duration with the activities that induce state changes (i.e. transitions). The duration of each activity is represented by a non-negative random variable with a known cumulative distribution function (c.d.f.).

Let $\Gamma = (\gamma_1, \gamma_2, \ldots, \gamma_{nt})$ be the set of the *nt* random variables associated with the *nt* transitions, then the set of their c.d.f. is:

$$G = [G_1(t), G_2(t), \ldots, G_{nt}(t)] \tag{4.161}$$

When a waiting time γ_k is associated with a transition t_k, the transition becomes enabled according to the rules of the untimed PN, but it can fire only after a time equal to γ_k has elapsed. This time between the enabling and the firing is referred to as the *firing time*.

Let $\{M(t), t \geq 0\}$ be the marking process, $M(t)$ representing the marking reached by the PN at time t. The following analysis is restricted to SPNs in which the random firing times have continuous c.d.f. with infinite support, i.e. $(0, \infty)$. With this assumption, the marking process $M(t)$ is a continuous-time, discrete-state, stochastic process with a state space that is isomorphic to the reachability graph of the untimed PN (i.e. the one looks exactly the same as the other).

Given a marking in which more than one transition with the same priority level (if priority is used) is enabled, the *firing policy* determines the transition that will fire next. There are thus two possible alternatives (Ajmone Marsan et al. 1995):

- Race policy: the transition of which the firing time elapses first is assumed to be the one that will fire next.
- Pre-selection policy: the next transition to fire is chosen according to an externally specified probability mass function independent of their firing times.

By far the most common firing policy for timed transitions is the race policy i). The pre-selection policy ii) is commonly used for immediate transitions, which are introduced for the first time into Markovian SPN (Ajmone Marsan et al. 1995).

Once the firing policy is defined, the *execution policy* must be specified. The execution policy consists of a set of specifications for uniquely defining the stochastic process, $\{M(t)\}$, under1ying an SPN. There are two elements that characterise the execution policy: a criterion to keep memory of the past history of the process (the memory policy), and an indicator of the re-sampling status of the firing time. The memory policy defines how the process is conditioned upon the past. An age variable a_g associated with the timed transition t_g keeps track of the time for which the transition has been enabled. A timed transition fires as soon as the memory variable a_g reaches the value of the firing time γ_g. In the activity period of a transition, the age variable is *not* 0.

The random firing time γ_g of a transition t_g can be sampled at a time instant prior to the beginning of an activity period. To keep track of the re-sampling condition of the random firing time associated with a timed transition, a binary indicator variable r_g that is equal to 1 is assigned to each timed transition t_g when the firing time is to be sampled, and equal to 0 when the firing time is not to be sampled. Reference is made to r_g as the *re-sampling indicator variable*. Hence, in general, the (continuous) memory of a transition t_g is captured by the tuple (a_g, r_g). At any time period t, transition t_g has memory (its firing process depends on the past) if either a_g or r_g is different from zero.

At the entrance to a marking, the remaining firing time (rft) has the value $\text{rft} = \gamma_g - a_g$, and is computed for each enabled transition given its currently sampled firing time γ_g and the age variable a_g. According to the race policy, the next marking is determined by the minimal of the rfts. The following execution policies can now be defined.

Execution policies A timed transition t_g can be:

- Pre-emptive repeat different (prd): if both the age variable a_g and the re-sampling indicator r_g are reset each time t_g is disabled or it fires.
- Pre-emptive resume (prs): if both the age variable a_g and the re-sampling indicator r_g are reset only when t_g fires.
- Pre-emptive repeat identical (pri): if the age variable a_g is reset each time t_g is disabled or fires but the re-sampling indicator r_g is reset only when t_g fires.

Transition t_g is prd—each time a prd transition is disabled or it fires, its memory variable a_g is reset and its indicator re-sampling variable r_g is set to 0 (the firing time must be re-sampled from the same distribution when t_g becomes re-enabled).

Transition t_g is prs—when t_g is disabled, its associated age variable a_g is not reset but maintains its constant value until t_g is re-enabled whereby $t_g = 1$. At each successive enabling point, a_g restarts from the previously retained value. When t_g fires, both a_g and r_g are reset so that the firing time must be re-sampled at the successive enabling point (γ_2). The memory of t_g is reset only when the transition fires.

Transition t_g is pri—under this policy, each time t_g is disabled, its age variable a_g is reset but its indicator re-sampling variable r_g remains equal to 1, and the firing time value γ_1 remains active, so that in the next enabling period an identical firing will result. The same value is maintained over different enabling periods up to the firing of t_g. Only when t_g fires are both a_g and r_g reset, and the firing time is re-sampled (γ_2). Hence, also in this case, the memory is lost only upon firing of t_g.

If the firing time is exponentially distributed, both the prd and prs policies behave in the same way. However, the pri policy does not have the property of no memory. Thus, the marking process of an SPN with only exponentially distributed firing times is not a continuous-time Markov chain (CTMC) if at least a single non-exclusively enabled transition exists with assigned pri policy.

If the firing time is deterministic, both the prd and pri policies behave in the same way (that is, re-sampling a deterministic variable always provides an identical value). The memory of the global marking process is considered as the superposition of the individual memories of the transitions. In general, the marking process $\{M(t)\}$ underlying an SPN is not analytically tractable (i.e. easily manageable) unless some restrictions are imposed (Ciardo et al. 1994).

Note that a simulation approach for the prd and the prs cases, based on very similar assumptions, has been adopted in the application simulation modelling of Sect. 4.4.

d) Definition of Markovian Stochastic Petri Nets (MSPN)

When all the random variables γ_k associated with the PN transitions are exponentially distributed, and the execution policy is *not* pri, the dynamic behaviour of the PN is mapped into a *continuous-time Markov chain (CTMC)* with state space isomorphic to the reachability graph of the untimed PN. This restriction is the most popular, and is usually referred to simply as MSPN or GSPN (Molloy 1982).

In order to completely specify the model, the set $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_{nt})$ of the *nt* firing rates assigned to the *nt* transitions is included. A usual convention in the graphical representation is to indicate transitions with exponentially distributed firing times by means of empty rectangles, and transitions with non-exponentially distributed firing times by means of filled rectangles, as illustrated in Fig. 4.20.

Modelling real systems often involves the presence of activities or actions (such as preventive maintenance activities) of which the duration is short or even negligible, with respect to the timescale of the process (especially continuous engineering

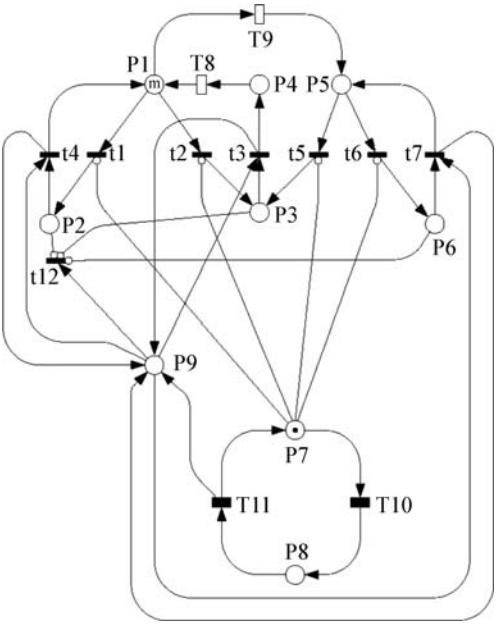


Fig. 4.20 Illustrative example of an MSPN for a fault-tolerant process system (Ajmone Marsan et al. 1995)

processes). Hence, it is desirable to associate an exponentially distributed firing time only with those transitions that are believed to have the largest impact on the system operation. The starting assumption in the MSPN model is that transitions are partitioned into two different classes, namely *immediate transitions* and *timed transitions* (Ajmone Marsan et al. 1995).

Immediate transitions fire in zero time once they are enabled, and have priority over timed transitions. Timed transitions fire after an exponentially distributed firing time (these are called EXP transitions). In the graphical representation of MSPN, immediate transitions are drawn as thin bars. Markings enabling immediate transitions are passed through in zero time and are called *vanishing states*. Markings enabling no immediate transitions are called *tangible states*. Since the process spends zero time in the vanishing states, they do not contribute to the dynamic behaviour of the system, and a procedure can be developed to eliminate these from the final Markov chain. With the partition of PN-transitions into a timed and an immediate class, a greater flexibility of modelling is achieved without increasing the dimensions of the final tangible state space from which the process measures are computed. An illustrative example of an MSPN is given in Fig. 4.20.

Dealing with large complex systems MSPNs can provide a compact representation of very large systems. This is reflected in an exponential growth of the reachable markings as a function of the primitive elements in the MSPN (places and transitions), and as a function of the number of tokens in the initial marking.

This exponential growth of the state space has often been recognised as a severe limitation in the use of the PN methodology to deal with real-life applications, and

a significant effort has been devoted to overcome or to alleviate this problem (Molloy 1982). Since Markovian-SPNs are based on the solution of a CTMC, all the techniques that have been explored to handle very large Markov chains can profitably be utilised in connection with MSPNs. When dealing with large models, not only does the solution of the system become difficult but also the model description and the computer representation become complex, which has resulted in an increasing application of *reachability graphs*.

e) Generating Reachability Graphs

The generation of a PN *reachability graph* (an extended and a reduced) is best explained with the aid of an example. Consider a process system based on a queuing client-server paradigm (typically in discrete event, single item and batch processing systems), the PN model being shown in Fig. 4.21. Transitions labelled t_{ek} or s_{tk} are timed transitions that fire after an exponentially distributed firing time EXP (represented by empty rectangles), and transitions labelled t_{ik} are immediate transitions that fire in zero time once they are enabled (represented by thin single-line bars). The system is made up of process units (clients) waiting in a controlled queue, requiring processing (transition t_{e1}) that can be supplied with probability $(1-c)$ (transition t_{i3}) by two servers (processing assemblies) working in parallel, and with probability c (transition t_{i1}) by accessing a resource (place p_{12}) shared by the two servers (in this case, the resource can be envisaged as some or other utility controlling the client queue and the servers, such as a distributed control system DCS). In the case of firing of t_{i3}, a message forwarded by the client is split into two sub-messages each addressed to a different server (places p_5 and p_6). The two servers are characterised by an exponentially distributed service time modelled by transitions s_{t1} and s_{t2} respectively.

It is assumed, in the definition of the process model, that a processing transaction is concluded when all the servers have served the sub-messages they have been assigned. When a server has processed its sub-message, it accesses the shared resource (DCS) to record its processing results (transitions t_{e2} and t_{e3}). After both servers have accessed the shared resource, a *join operation* is performed and the processed result is returned to control the client queue (i.e. transition t_{i6}).

Conversely, with probability c, the message of a client in the queue is already available in the shared resource, so that the service requirement is met by the server accessing the resource, retrieving the message and returning it to control the client queue (transitions t_{i2} and t_{e4}). The reachability graph illustrated in Fig. 4.22 can now be generated from the initial token distribution depicted in the PN model shown in Fig. 4.21 and the markings of Table 4.3.

The *extended reachability graph* of an MSPN comprises both tangible and vanishing states. Elimination of the vanishing states results in a *reduced reachability graph* that is isomorphic to the CTMC. Given a vanishing marking denoted by m_b (which is directly reachable from a tangible marking m_a), and the set of tangible markings S, reached from m_b passing through a sequence of vanishing markings

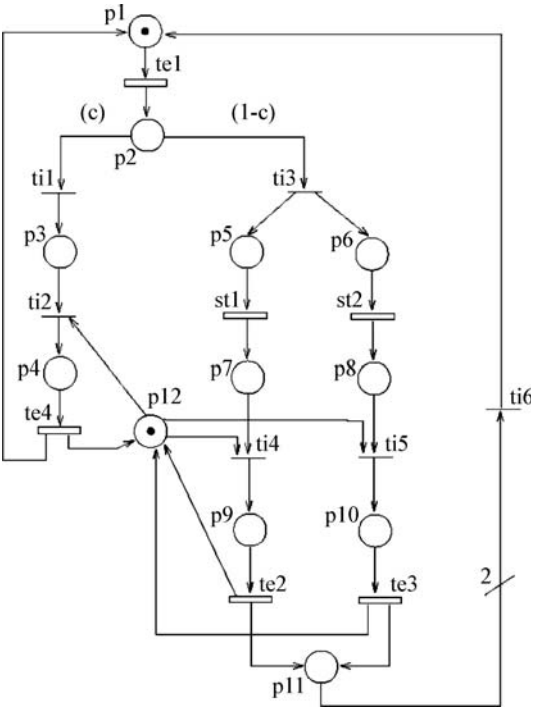


Fig. 4.21 MSPN for a process system based an a queuing client-server paradigm (Ajmone Marson et al. 1995)

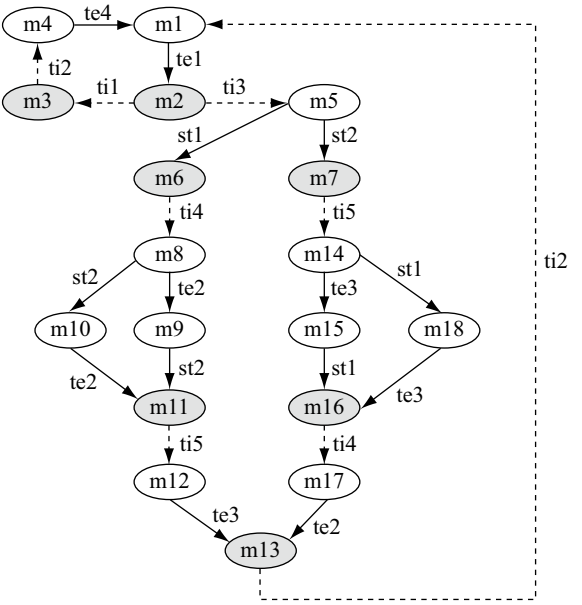


Fig. 4.22 Extended reachability graph generated from the MSPN model (Ajmone Marsan et al. 1995)

only, it is possible to evaluate the probability of the next tangible marking after m_b over S. Furthermore, m_a may belong to S. The vanishing marking m_b and the ones reachable from m_b by the firing of immediate transitions can be eliminated only by introducing arcs directly connecting m_a to $m_c \in S$, $m_c \neq m_a$, and by modifying the firing rate associated with the generic transition t_k enabled in m_a (Ajmone Marsan et al. 1995).

Table 4.3 gives the distribution of the tokens in the reachable markings. It is quite evident that the markings $m_2, m_3, m_6, m_7, m_{11}, m_{13}$ and m_{16} are vanishing (shadowed markings in Fig. 4.22) and can be eliminated. The *reduced reachability graph*, defined over the tangible markings only, can then be generated as illustrated in Fig. 4.23.

Once the reduced reachability graph is obtained, the matrix for the underlying continuous-time Markov chain (CTMC) can be constructed. Let R_0 be the reduced reachability graph of a Markovian SPN, and N its cardinality. The infinitesimal generator of the underlying CTMC is then a $N \times N$ matrix Q, where $Q = [Q_{ij}]$.

Let $\Pi(t)$ be the N-dimensional state probability vector, of which the generic entry $\pi_i(t)$ is the probability of being in state $i (i = 1, 2, \ldots, N)$ at time t in the associated CTMC. Then, $\Pi(t)$ is the solution of the standard linear differential equation:

$$\frac{\mathrm{d}}{\mathrm{d}t}\Pi(t) = \Pi(t) \cdot Q \tag{4.162}$$

with initial condition:

$$\Pi(0) = [1, 0, 0, \ldots, 0] \, .$$

Table 4.3 Distribution of the tokens in the reachable markings

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}
m_1	•											•
m_2		•										•
m_3			•									•
m_4				•								
m_5					•	•						•
m_6						•	•					•
m_7					•			•				•
m_8						•			•			
m_9						•					•	•
m_{10}								•	•			
m_{11}								•			•	•
m_{12}										•	•	
m_{13}											•	•
m_{14}					•					•		
m_{15}					•						•	•
m_{16}							•				•	•
m_{17}									•		•	
m_{18}							•			•		

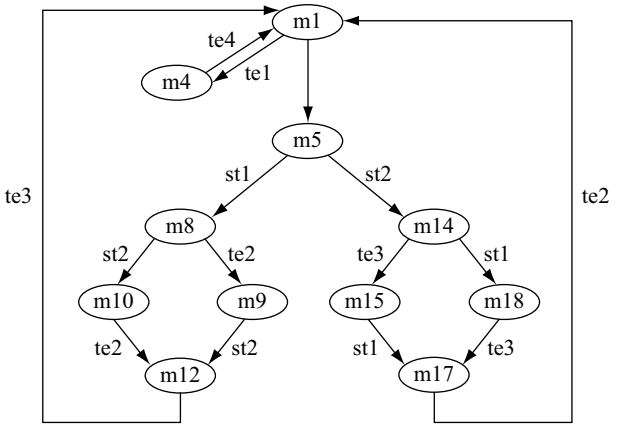


Fig. 4.23 Reduced reachability graph generated from the MSPN model

If the steady-state probability vector $\Pi = \lim_{t\to\infty} \Pi(t)$ of the CTMC exists, it can be calculated that:

$$\Pi Q = 0$$

with:

$$\sum_{i=1}^{N} \pi_i = 1$$

Since some of the output measures depend on the integrals of the probabilities, rather than on the probabilities per se, it is necessary to provide the appropriate computation of the integrals of the state probabilities:

$$L_i(t) = \int_0^t \pi_i(z)\,\mathrm{d}z \tag{4.163}$$

where $L_i(t)$ is the expected time that the CTMC stays in state i during the interval $(0,t)$.

Let $\mathbf{L}(t)$ denote the N-dimensional row vector consisting of the elements $L_i(t)$. Integrating both sides of Eq. (4.162), the following relation is obtained:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{L}(t) = \mathbf{L}(t)\cdot Q + \Pi(0) \tag{4.164}$$

$\mathbf{L}(t)$ = N-dimensional row vector
Q = $N \times N$ matrix of the CTMC
$\Pi(0)$ = initial condition of the N-dimensional state probability vector.

f) Measures of Markovian Stochastic Petri Nets (MSPN)

A fundamental property of the time-dependent representation of system behaviour through SPNs is that they enable the user to define, in a simple and natural way, a large number of different measures related to the performance and reliability of the system.

The stochastic behaviour of a Markovian-SPN is determined by calculating the $\Pi(t)$, $\Pi(0)$ and $\mathbf{L}(t)$ vectors over the reduced reachability set of R_0. However, the final output measures should be defined at the Petri net level as a function of its primitive elements (i.e. places and transitions). The following mathematical models provide a practical outline as how to relate the probabilities at the CTMC level with useful measures at the PN level.

The probability of a given condition on the SPN By means of logical or algebraic functions of the number of tokens in the PN places, a particular condition C (e. g. no tokens in a given place) can be specified, and the subset of states $S \in R_0$ can be identified for which the condition is true. The output measure:

$$C_s(t) = \text{Prob}\,\{\text{condition } C \text{ is true at time } t\}$$

given by:

$$C_s = \sum_{s \in S} \pi_s(t) \tag{4.165}$$

where $\pi_s(t)$ is the probability of being in state s at time t.

Note: if S is the set of operational states, $C_s(t)$ is the usual definition of *system availability*.

A very useful case arises when the measure is the transient probability that the condition is satisfied for the first time. By using such an approach in the analysis of stochastic processes, the states $s \in S$ can be made absorbing (i.e. assimilated), and the quantity evaluated from Eq. (4.165) as the value of the process when entering S. In this way, the above equation can be used to calculate *system reliability*:

$$C_s(t) = \sum_{s \in S} \pi_s(t)$$

System availability:
where $S =$ set of operational states.

System reliability:

$$C_s(t) = \sum_{s \in S} \pi_s(t)$$

where $s \in S$ and process entering S.

The time spent in a marking Let $S \in R_0$ be the subset of markings in which a particular condition is fulfilled. The expected time, $\psi_s(t)$, spent in the markings $s \in S$ during the interval $(0,t)$ is given by:

$$\begin{aligned}\psi_s(t) &= \sum_{s\in S}\int_0^t \pi_s(z)\,\mathrm{d}z \\ &= \sum_{s\in S} L_s(t)\end{aligned} \tag{4.166}$$

Moreover, from the theory of irreducible Markov chains, as t approaches infinity, the proportion of the time spent in states $s \in S$ equals the asymptotic probability (Choi et al. 1994):

$$\begin{aligned}\psi_s(t) &= \sum_{s\in S} \pi_s \\ &= \lim_{t\to\infty} \frac{\psi_s(t)}{t}\end{aligned} \tag{4.167}$$

$\psi_s(t)/t$ represents the *utilisation factor* in the interval $(0,1)$, and ψ_s the expected steady-state utilisation factor. For example, if S is the set of states in which a process is idle, $\psi_s(t)/t$ is the fraction of idle time in $(0,1)$ and ψ_s is the expected idle time.

The mean first passage time Given that $C_s(t)$, as calculated in Eq. (4.165), is the probability of having entered subset S before t for the first time, the mean first passage time μ_s can be calculated as:

$$\mu_s = \int_0^\infty [1 - C_s(z)]\,\mathrm{d}z \tag{4.168}$$

This formula requires the transient analysis to be extended over long intervals. There are other direct techniques for calculating mean first passage times in a CTMC but these are not relevant to this research (Ciardo et al. 1994).

The distribution of tokens in a place The cumulative distribution function (c.d.f.) of the number of tokens in place p_i of the SPN at time t is a step function in which the amplitude of the kth step is obtained by summing up the probabilities of all the states in the set R_0 containing k tokens ($k = 0, l, 2, \ldots, K$) in p_i at time t. The probability function $f_i(k,t)$ is the amplitude of the kth step. The expected value of the number of tokens in place p_i at time t is:

$$ET[m_i(t)] = \sum_{k=0}^{\infty} k f_i(k,t) \tag{4.169}$$

As an example, if place p_i represents identical units in a queue for a common resource, the above quantity gives the expected value of the number of units in the queue at time t. In reliability analysis, the tokens in place p_i represent the number of failed components.

The expected number of firings of a PN transition Given an interval $(0,t)$, the expected number of firings would indicate how many times, on average, an event modelled by a PN transition has occurred in that interval. Let t_k be a generic PN transition, and let S be the subset of R_0 that includes all the markings $s \in S$ enabling t_k. The expected number of firings of t_k in $(0,t)$ is given by:

$$\begin{aligned} \eta_k(t) &= \sum_{s \in S} \lambda_k(s) \int_0^t \pi_s(z)\,\mathrm{d}z \\ &= \sum_{s \in S} \lambda_k(s) \cdot L_s(t) \end{aligned} \tag{4.170}$$

where $\lambda_k(s)$ is the firing rate of t_k in marking s. In steady state, the expected number of firings per unit of time becomes:

$$\eta_k(t) = \sum_{s \in S} \pi_s \lambda_k(s) \tag{4.171}$$

This quantity represents the *throughput* associated with the given transition. If transition t_k represents the completion of a service in a queuing system, $\eta_k(t)$ is the expected number of services completed in time $(0,t)$ and η_k is the expected *steady-state throughput*.

g) Definition of Stochastic Reward Nets

Stochastic reward nets (SRN) introduce a new extension into Markovian-SPNs, allowing for the possibility of associating reward rates to the markings. The reward rates are specified at the PN level as a function of its primitives (i.e. the number of tokens in a place, or the rate of a transition). The underlying CTMC is then transformed into a *Markov reward model*, thus permitting the eva1uation of performance measures. Implementation of this extension allows the reward structure superimposed on the reachability graph to be generated automatically, and easily provides performance measures (Ciardo et al. 1991).

The reward definition is called *rate-based*, to indicate that the system produces reward at rate $r(i)$ for all the time it remains in state $i \in R_0$. Furthermore, *impulse-based* reward models can be implemented where a reward function r_{ij} is associated with each transition from the state $i \in R_0$ to $j \in R_0$. Each time a transition from i to j occurs, the cumulative reward of the system instantaneously increases by r_{ij}. In general, several combinations of the different reward functions can be specified in the same model.

h) Definition of Non-Markovian Stochastic Petri Nets

As indicated previously, in order to define a PN with generally distributed transitions, the following entities must be specified for each transition: $t_g \in T$: the

c.d.f. $G_g(t)$ of the random firing times γ_g, and the execution policy for determining (a_g, r_g).

Several classes of SPN models have been developed that incorporate some non-exponential characteristics in their definition, and that adhere to the individual memory requirements indicated previously.

With the aim of specifying non-Markovian SPN models that are analytically tractable, three approaches can be considered, specifically (Bobbio et al. 1997):

- An approach based on Markovian regenerative theory
- An approach based on the use of supplementary variables
- An approach based on state space expansion.

The first approach originates from a particular definition of a non-Markovian SPN where, in each marking, a single transition is allowed to have associated with it a deterministic firing time with prd execution policy (i.e. a deterministic SPN, or DSPN). The marking process underlying a DSPN is a *Markov regenerative process (MRGP)* in which equations can be derived for the transition probability matrix in transient and in steady-state conditions (Choi et al. 1994).

Generalisation of the previous formulation is proposed by including the possibility of modelling prs transitions and also by including pri transitions. The most general framework under which the Markov regenerative theory has been applied is where any regeneration time period is dominated by a single transition (non-overlapping dominant transitions).

The second approach resorts to the use of *supplementary variables*. This method has been applied to prd execution policies only, and with mutually exclusive general transitions. A steady-state solution has been proposed, while the possibility of applying the methodology to transient analysis has also been explored (German et al. 1994).

The third approach is based on the expansion of the reachability graph of the basic PN. In this approach, the original non-Markovian marking process is approximated by means of a continuous-time Markov chain (CTMC), defined over an augmented state space. According to the definitions given previously, the reachability graph expansion technique can be realised by assigning a continuous distributed random variable to each transition (Neuts 1981).

Basically, the merit of this approach is the flexibility in modelling any combination of prd and prs memory policies, and any number of concurrent or conflicting transitions with generally distributed firing times. Furthermore, the reachability graph expansion technique can be implemented using a computer program. Starting from the basic specification at the PN level, all the solution steps can be hidden from the modeller in an OOP environment. The drawback of this approach is, of course, the explosion of the state space.

4.3.2.2 Designing for Availability Using Petri Net Modelling

Returning to the initial quantitative example of designing for availability with the inclusion of preventive maintenance, Fig. 4.24 illustrates the MRSPN representation of the system (Bobbio et al. 1997).

The working state is modelled by place P_{up}. The generally distributed transition t_f models the failure distribution of which the firing results in the system moving to place P_{down}. Upon system failure, the preventive maintenance activity is suspended; the inhibitor arc from place P_{down} to transition t_{clock} is used to model this fact.

The deterministic transition t_{clock} models the constant inspection interval. It is competitively enabled with t_f, so that the one that fires first disables the other. Once t_{clock} fires, a token moves in place P_{mai}, as well as the activity related with the preventive maintenance (transition t_{mai} starts). During the preventive maintenance phase, the system is down and cannot fail by using the inhibitor arc from place P_{mai} to transition t_f.

The completion of the maintenance (firing of t_{mai}) re-initialises the system in an as-good-as-new condition; hence, t_f is assigned a prd policy. Since upon failure (and repair) a completed interval must elapse before the successive preventive maintenance takes place, t_{clock} also must be assigned a prd policy. As can be observed from Fig. 4.24, t_f and t_{clock} are conflicting prd transitions.

a) Numerical Computations for the Availability Petri Net Model

Since there are no immediate transitions in the PN, all the markings are tangible. Starting from an initial marking m_1, the token distribution of the reachable markings represented in Fig. 4.24 (assuming the following order for the places: P_{up}, P_{clock}, P_{down}, P_{mai}) is given by:

$$m_1 = (1,1,0,0)\,, \quad m_2 = (0,1,1,0)\,, \quad m_3 = (1,0,0,1)\,.$$

From marking m_1, both t_f and t_{clock} may fire, leading to m_2 and m_3 respectively. From m_2, only t_{down} can fire, leading to m_1 and, finally, from m_3 only t_{mai} can fire,

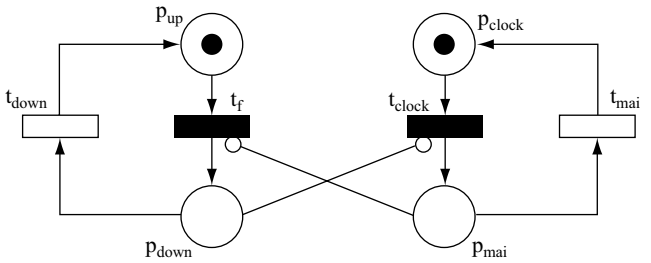


Fig. 4.24 MRSPN model for availability with preventive maintenance (Bobbio et al. 1997)

leading to m_1. As a consequence, the matrices $E(t)$ and $K(t)$ have the following structure:

$$E(t) = \begin{bmatrix} E_{11}(t) & 0 & 0 \\ 0 & E_{22}(t) & 0 \\ 0 & 0 & E_{33}(t) \end{bmatrix} \tag{4.172}$$

$$K(t) = \begin{bmatrix} 0 & K_{12}(t) & K_{13}(t) \\ K_{21}(t) & 0 & 0 \\ K_{31}(t) & 0 & 0 \end{bmatrix} \tag{4.173}$$

Since $E(t)$ is a diagonal matrix, the marking process is an SMP. Let $G_f(t)$ be the c.d.f. of the firing time associated with transition t_f, and d be the deterministic maintenance interval associated with t_{clock}.

Furthermore, let λ_1 and λ_2 be the firing rates associated with the transitions t_{down} and also t_{mai} respectively. The non-zero matrix entries are:

$$K_{12}(t) = \begin{bmatrix} G_f(t) & 0 \le t < d \\ G_f(d) & t \ge d \end{bmatrix} \tag{4.174}$$

$$K_{13}(t) = \begin{bmatrix} 0 & 0 \le t < d \\ 1 - G_f(d) & t \ge d \end{bmatrix} \tag{4.175}$$

$$K_{21}(t) = 1 - \mathrm{e}^{-\lambda_1 t} \tag{4.176}$$

$$K_{31}(t) = 1 - \mathrm{e}^{-\lambda_2 t} \tag{4.177}$$

$$E_{11}(t) = \begin{bmatrix} 1 - G_f(I) & 0 \le t < d \\ 0 & t \ge d \end{bmatrix} \tag{4.178}$$

$$E_{22}(t) = \mathrm{e}^{-\lambda_1 t} \tag{4.179}$$

$$E_{33}(t) = \mathrm{e}^{-\lambda_2 t} \tag{4.180}$$

b) Steady-State Solution to the Availability Petri Net Model

To obtain the steady-state solution, the following procedure is given:
STEP 1:

$$\alpha = \begin{bmatrix} \alpha_{11} = \int [1 - G_f(t)]\,\mathrm{d}t & 0 & 0 \\ 0 & \alpha_{22} = 1/\lambda_1 & 0 \\ 0 & 0 & \alpha_{33} = 1/\lambda_2 \end{bmatrix}$$

$$\varphi = \begin{bmatrix} 0 & G_f(d) & 1 - G_f(d) \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

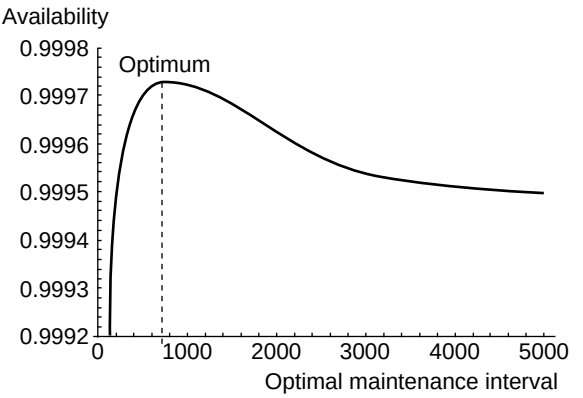


Fig. 4.25 MRSPN model results for availability with preventive maintenance

STEP 2:

$$D = [1/2\,, \quad 1/2[G_f(d)]\,, \quad 1/2[1 - G_f(d)]]$$

STEP 3:

$$v = [1/A_2 \cdot \alpha_{11} \quad 1/A_2 \cdot \alpha_{22} G_f(d)\,, \quad 1/A_2 \cdot \alpha_{33}[1 - G_f(d)]]$$
$$A = 1/2\alpha_{11} + 1/2\alpha_{22} G_f(d) + 1/2\alpha_{33}[1 - G_f(d)]$$

The steady-state availability is given by the probability of being in state m_1 (entry v_1 in Step 3). The effect of the length of the *preventive maintenance* interval d on *system availability* can now be determined.

The numerical computations are made assuming the following values:

i) Transition t_f is distributed according to the Weibull cumulative distribution function $G_f(t) = 1 - \mathrm{e}^{-ct\beta}$, with β the shape parameter and c the scale parameter. Assume $\beta = 2.0$ for an increasing failure rate (i.e. wear-out).
ii) Let $\lambda_1 = 0.1\,\mathrm{h}^{-1}$ and $\lambda_2 = 1.0\,\mathrm{h}^{-1}$ for the firing rates of the transitions t_{down} and t_{mai} respectively.
iii) The preventive maintenance interval d varies from 0 to 5,000 h.

Figure 4.25 is a representative plot of system availability t_1 versus the maintenance interval d. If $d = 0$, the system is always under maintenance and is completely unavailable. As d increases, the steady-state availability increases as well. For large d, however, the effect of the preventive maintenance is nullified by the downtime due to failure and, in the limit $d \to \infty$, the availability approaches the value when there is no preventive maintenance. The optimal maintenance interval d can now be computed at which the availability achieves its maximum value $v_t \to$ maximum.

4.3.3 Analytic Development of Availability and Maintainability Evaluation in Detail Design

Appropriate methods for further development as tools for *availability and maintainability evaluation* in determining the integrity of engineering design during the *detail design* phase are:

i. The application of systems engineering in engineering design, particularly systems engineering analysis (SEA).
ii. The evaluation of complexity in integrated systems through complex systems theory (CST).

4.3.3.1 Systems Engineering and Complex Systems Theory

Systems engineering is a discipline that establishes a structured analysis approach to evaluate complex engineering design problems. Because systems engineering focuses in this case on the methodology of analysis and synthesis for determining the overall integrity of complex integrated systems in engineering design, rather than its execution, describing it precisely is more difficult than for other engineering disciplines. Furthermore, its description varies considerably, particularly between industrial and research applications. Industrial demand for systems engineering is so pervasive that the approach is highly focused on methods for problem-solving in the operation of engineered installations as well as in their engineering design, while systems engineering in research concentrates mainly on mathematical methods and algorithms needed for evaluating the complexity of these designs.

In the development of tools for availability and maintainability evaluation in determining the integrity of engineering design during the detail design phase, key characteristics of systems engineering are considered in industrial applications. Because the initial engineering approach must be *quantitative*, systems engineering relies on mathematics, both for the representation of the real world as models and simulations, and for analysis and synthesis in mathematical methods or algorithms. This focus on mathematical methods and modelling translates the discipline of systems engineering into *systems engineering analysis (SEA)*.

Systems engineering analysis is embodied in computer-based analysis of complexity in engineering design, as well as in software programming. With the growing emphasis on computer aided design (CAD), systems engineering is increasingly providing a central and complementary role as an integrating factor in collaborative engineering design.

Complex systems theory (CST) cuts across the boundaries between conventional scientific disciplines. It makes use of methods and examples from many disparate fields, and its results are widely applicable to a great variety of scientific and engineering problems (Wolfram 1988). Engineering systems, particularly industrial process systems, are often described as being complex (Boullart et al. 1988; Pritsker 1990). The dynamic nature of process systems as well as the complex integration

of these systems make it difficult to predict the effect of design decisions on future system performance. Many integrated systems, which are designed to be flexible, are constrained by their complexity in being inflexible.

An understanding of the effects of systems integration on system complexity is essential for realising the full potential of process systems, their successful deployment in the process industry, and the economic justification of new process technologies. However, literature on system complexity, specifically in the process and manufacturing context, is sparse (Ayres 1988; Deshmukh 1993).

The notion of complex systems has been thoroughly considered in *systems engineering analysis* literature. Extrapolating from various different contexts in which the idea of complexity is used, a complex system may be defined as one having a static structure or dynamic behaviour that is counterintuitive or unpredictable (Casti 1979). A complex system may also be referred to as a system that has patterns of connections among sub-systems, such that any prediction of system behaviour is difficult without substantial analysis or computation; or one in which decision-making of alternative options in engineering design makes the effects of individual choices difficult to evaluate (Simon 1981).

Computational or algorithmic complexity is often used for classifying process control problems (Garey et al. 1979). However, computational complexity does not capture all the aspects of complexity in engineering systems. Also, computational complexity does not always relate to the performance of the system, since computational complexity is an algorithm-related measure.

The complexity of a physical system can be characterised in terms of its static structure or time-dependent behaviour. Static complexity can be viewed as a function of the structure of the system, connective patterns, variety of components, and the strengths of interactions, whereas dynamic complexity is concerned with unpredictability in the behaviour of the system over a time period (Deshmukh 1993).

The process environment consists of physical systems in which concurrent and sequential processes take place in order to produce an output. The nature of these processes is dependent not only upon system capabilities but also on the process characteristics (inputs, throughputs and outputs) being produced in the system. Hence, any measure of system complexity should be dependent on both the system and process characteristics, particularly with integrated systems that result in a multiplicity of characteristic-related event criteria. However, the *complexity principle* states that as the complexity and uncertainty of an engineering system increases, our ability to predict its behaviour diminishes, until a threshold is reached beyond which accuracy and significance become almost mutually exclusive. This is often termed the *threshold of chaos*. Phenomena that are chaotic are unpredictable (non-repeatable) and, hence, cannot be optimised. The main reason for this is an extreme sensitivity to initial conditions. Most complex systems contain some chaos. All that can be done with chaotic phenomena is to increase the analysis of their properties, patterns, structure and occurrence (Zadeh 1979).

The difficulty in making design decisions with complex systems arises from the number of choices available at each decision point relating to each event in the range of the event-criteria possibilities, and the unpredictability of the effects of

these events on system performance. Computational complexity can be considered to be the algorithmic effort required to evaluate these choices. In addition to systems complexity, there is a further aspect that relates to the *control* of process systems: static and dynamic complexities are usually considered assuming constant control schemes. However, different process systems require varying levels of control, further complicating the difficulty of regulating complex processes. The notion of complexity needs to be qualified and quantified in order to compare different system alternatives. The general lack of understanding in this area has hindered designers in deciding to what extent systems integration is beneficial, and beyond which point integration is actually detrimental to system performance, since correct decisions are difficult to make due to high system complexity.

Another important consequence of developing an analytical framework for complexity would be to assist designers in managing desired levels of complexity in the system because, realistically, this cannot be eliminated due to the unavoidability of systems integration in large engineered installations, resulting in unpredictable changes in operating conditions. A fundamental problem of defining the notion of complexity beyond simply a term for a phenomenon that appears to be counter-intuitive or unpredictable, into a more formalised language (i.e. mathematically) whereby the differences between the complex and commonplace can be better understood, is that it involves making that which is fuzzy, precise (Casti 1994).

In an effort to understand what is involved with complex systems, it would be intuitive to first consider some of the properties associated with *simple* systems, before attempting to express complexity in mathematical terms.

- *Predictable behaviour*: simple systems give rise to behaviour that is easy to deduce if the system's process characteristics (i.e. inputs, throughputs and outputs) can be defined. Such predictable behaviour is one of the principal characteristics of simple systems.
- *Interaction and feedback loops*: simple systems generally involve a small number of components with interactions that dominate the linkages among the process characteristic variables. In addition to having only a few variables, simple systems generally consist of a few feedback/feed-forward loops that enable modification or regulation of interactions among the process characteristic variables.
- *Centralised control*: in simple systems, control is centred with very little, if any, independent interactive control between lower-tiered components. Such systems tend to be more robust, as they are better able to absorb process fluctuations.
- *Decomposable and reducible*: a simple system involves relatively weak interactions among its various components that, if disconnected or degraded, would not result in a total loss of process control or unstable behaviour.

By establishing a workable *complex systems theory*, a framework can be structured within which complex systems can be better understood from the perspective of engineering design, process control and operational stability. More importantly, CST can provide a means of determining the limits of reduction of complex systems for systems engineering analysis.

4.3.3.2 Systems Engineering in Engineering Design

Like many other engineering disciplines, systems engineering:

- involves central concepts;
- uses specific methodologies;
- includes both analysis and synthesis for evaluating engineering design;
- relies on mathematics to express knowledge;
- bears an interdependent relationship to other engineering disciplines (since many design problems are cross-disciplinary);
- provides profound benefit to engineering and industry in particular;
- stimulates research for further engineering benefit.

Many of the key thrusts of systems engineering are found within the other engineering disciplines. However, systems engineering is qualitatively different. Systems engineering differs from the basic engineering disciplines in that these disciplines concentrate on using knowledge of the real world for systems construction, e.g. materials, structures, electrical circuits, robotics, whereas systems engineering finds its focus in constructs of analysis and synthesis for problems involving multiple aspects of complex real-world systems. The effectiveness of systems engineering in analysing complex systems is determined by methodologies, algorithms and tools available for advanced systems engineering analysis, such as performance metrics, optimisation methods in the presence of various kinds of constraints, marginal and sensitivity analysis, linear/non-linear programming, dynamic programming, utility theory, decision analysis, mathematical modelling and simulation modelling (INCOSE 2002).

Systems engineering in engineering design involves several distinguishing characteristics, such as:

- Design problems are highly inter-disciplinary: systems engineering in engineering design typically involves a spectrum of conventional engineering and science disciplines.
- Design problems require high-level metrics: systems engineering problems place a high priority on measuring and optimising values at higher levels of systems integration.
- Design problems are hierarchical: as a result of integrating various factors into high-level metrics, systems engineering structures large-scale systems into a vertical hierarchy.
- Multiple metrics and optimisation are crucial: integrating a plurality of system performance metrics leads to difficult challenges in multivariate optimisation of design input variables to achieve reasonable optimisations of the various output metrics.
- There is additional heterogeneity: the behaviour of systems brings additional heterogeneity into systems engineering problems that add more diversity to complexity considerations.

- The problems are dynamic: systems engineering places emphasis on dynamic variations in time, necessitating design-for-integrity through a concurrent engineering approach.
- Methodologies for process life cycle are central: because systems engineering emphasises a structured approach to the analysis of design, analytic methodologies are central.
- Systems definition and development: systems engineering methods such as analysis of systems complexity, hierarchical modelling and concurrent engineering design provide for a more comprehensive approach to process engineering design.
- Integrity of design: uncertainties in the development process underscore the importance of systems engineering approaches to the integrity of process engineering design, together with system performance and life-cycle costs.
- Non-technical components and metrics: while cost and human resource factors are normally considered intrinsic factors in conventional engineering disciplines, systems engineering places an explicit, high priority on these factors.
- Other non-technical disciplines: human factors play a crucial role in systems engineering, such as the disciplinary requirements in computer systems.
- Government regulatory policy and decision-making: systems engineering application in many large-scale process projects involve not only technical components but political, economic and sociological factors as well.

4.3.3.3 Complexity in Engineering Design and Systems Engineering

Systems engineering analysis (SEA) brings out clearly a systematic reasoning process in which all the uncertainties associated with complex integrations of multiple systems that may have impeded decision-making during the early phases of engineering design are properly considered. Systems engineering analysis examines uncertainties and assumptions made in the conceptual and preliminary design phases, to determine the end-result integrity of the engineered installation as a whole. It is a study of *total* systems performance, rather than a study of its elements. It stems from the recognition that, even if each element of a system is optimised from a design point of view, the total systems performance may be *less* than optimal, owing to complex interactions between the elements.

All complex systems have certain characteristics encountered not only in their design but also in their application that cause many of the critical malfunctions in industrial plant and equipment. Among these characteristics are the following:

- *Change*: the present state or condition of a system is the result of past performances or its engineering design. No real-world system remains static over a long period of time. The flow of the process enters and leaves the system either through a birth-and-death occurrence or by passing through system boundaries.
- *Environment*: each system has its own environment and is, in fact, a sub-system of some broader system. The environment of a system is a set of elements and

their relevant properties that, although not a part of the system, if modified can produce a change in the state of the system as a whole.
- *Counteraction results*: examination of some systems might indicate the need for corrective action. This action can often be ineffective or even adverse in its results. Corrective action in complex systems may intensify a problem, rather than solve it.
- *Drift to low performance*: complex systems generally tend towards a condition of reduced performance with time. Components deteriorate and inefficiencies creep in, their counteraction nature causing detrimental design changes.
- *Interdependency*: no activity in a complex system takes place in total isolation. Each event is influenced by its predecessor and affects its successors. In addition, real-world activities are generally parallel and ultimately influence each other.
- *Organisation*: all complex systems consist of highly organised elements or components. These elements are combined into hierarchies of sub-systems, assemblies, components and parts that interact to carry out the function of the system.
- *Variance*: outputs from complex systems tend to have greater variances about a mean result, because of the individual variances in performance of the constituent elements.

Open and closed systems A *closed system* is considered to be one in which only the components within the system are assumed to exist. All other influences or variables from outside the system are considered to be non-existent, or to be insignificant. It is a hypothetical assumptive system, as there probably never is a completely closed system. Components within a system are always subject to outside influences. Closed systems are usually adopted for initial analysis, as they are usually simple and each component in the system is more easily analysed with regard to its effect on the other components in the system.

An *open system* is described by the basic properties of:

- *Inputs*: inputs (exogenous variables) are the independent variables of the system model, and are pre-determined. Input variables can be classified as either controllable or non-controllable. Controllable input variables can be manipulated. Non-controllable input variables are generated by the environment in which the system exists, and not by the system.
- *Throughputs*: throughputs (status variables) are indicative of the capability of the system to achieve the desired output.
- *Outputs*: outputs (endogenous variables) are the dependent or output variables of the system model, and are generated from the interaction of the system's output and status variables, according to the system's operating characteristics.

Added to this are other attributes such as cyclic events, continuity, and differentiation of functions.

An open system recognises and permits all interactions of its components to take place across the boundaries of the system. It is more realistic, though more complex, than closed systems and, therefore, more difficult to analyse or control. Diagrams of a model of a closed system and a model of an open system are given below (Fig. 4.26), together with the typical symbols used in such system models.

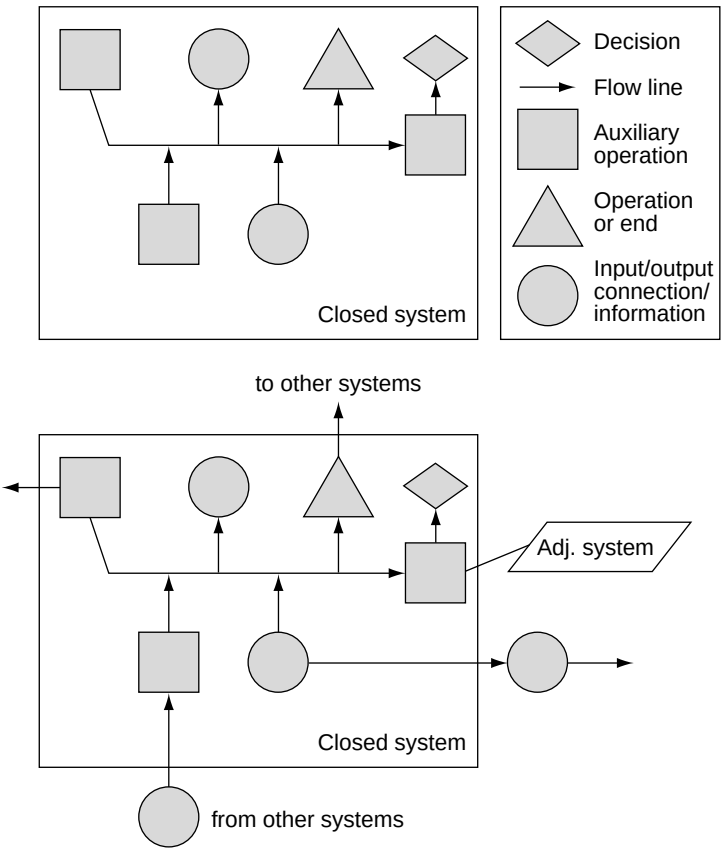


Fig. 4.26 Models of closed and open systems

a) Functions of Systems Engineering Analysis

Systems engineering analysis consists of the following functions:

- Problem definition
- System objectives
- System boundaries
- System components
- Requirements analysis
- Functional analysis
- Effectiveness measures
- Constraints evaluation
- Choosing alternatives
- Evaluating alternatives.

Problem definition The first step in systems engineering analysis is to define the problem. It is extremely important to examine critically whether the statement of the problem expresses the reality of the problem. In most process engineering designs, the design problem considers the criteria of system configuration, process description and problem definition. For example, consider the following process engineering design:

- *Systems configuration*: two coal slurry preparation, gasifier and gas cleaner (scrubber) lines in parallel, each with separate oxygen inputs into the gasifier.
- *Process description*:
  (1) A coal plant feeds coal to two coal slurry preparation mills.
  (2) The slurry mills feed two coal gasifiers, each with separate oxygen inputs from two oxygen compressors.
  (3) The gas from the two gasifiers are fed into two gas cleaners or scrubbers, from which raw fuel gas is obtained.
- *Problem definition*: determine the reduction in plant flow capacity as the number of unavailable sub-systems increases due to system deterioration, and consider the most appropriate alternatives to maintaining optimum availability.

System objectives It is also important to examine statements of objectives carefully for possible inconsistencies. An example of an inconsistent objective is the frequently expressed 'maximising effectiveness at the least cost'. It is, however, highly unlikely that effectiveness can be maximised and costs minimised simultaneously. The objective should be stated as 'maximisation of effectiveness for a given cost' or, alternatively, 'minimisation of cost for a given effectiveness'. For the example coal slurry preparation plant, the system objective may be stated as maximising plant flow capacity by optimising systems availability.

System boundaries A problem always encountered in systems engineering analysis with systems optimisation is the difficulty or impracticality of analysing the entire system or engineered installation (plant). When analysis of the total system is not possible, optimisation of each sub-system may be feasible but the total system may be sub-optimal. If the scope of the total system limits the extent of system optimisation, then definition of the system boundaries within which the analysis will take place must be made. These boundaries are usually identified by the following criteria:

- Material or process flow.
- Mechanical action.
- State changes.
- Changes in process characteristics (inputs, throughputs or outputs).

For the example coal slurry preparation plant, the system boundaries to be taken into consideration will be defined during the functional analysis of the various systems in which, for the sake of simplicity, a *closed system* approach will be taken.

System components This step requires the specification of systems elements within the specified systems boundary. In order to establish uniform terminology for later use, system hierarchy definitions are necessary. These system hierarchy definitions are considered, firstly from the overall plant down to its systems, then to its sub-systems or assemblies, and to its sub-assemblies or components. A *schematic process flow block diagram* of the coal slurry preparation plant is illustrated in Fig. 4.27.

The design objective is concerned with *plant capacity* and the *availabilities* of each of the plant's systems. At this stage, it would suffice to regard a three-level systems hierarchy of a single plant with several *system groups*, and several sub-systems within each group. Finalisation of the hierarchical grouping will coincide with a *requirements analysis* as well as a *functional analysis*. At this stage, the sub-systems are two coal slurry preparation mills, two coal gasifiers, two oxygen compressors, and two gas cleaners or scrubbers, arranged in two parallel coal slurry preparation lines or *system groups*.

Requirements analysis Requirements analysis consists of the identification and evaluation of use. This analysis is possible once a systems hierarchy is identified, and usually takes into consideration the sub-system's assembly level, but can in some instances go down to sub-assembly and/or component level, depending on the

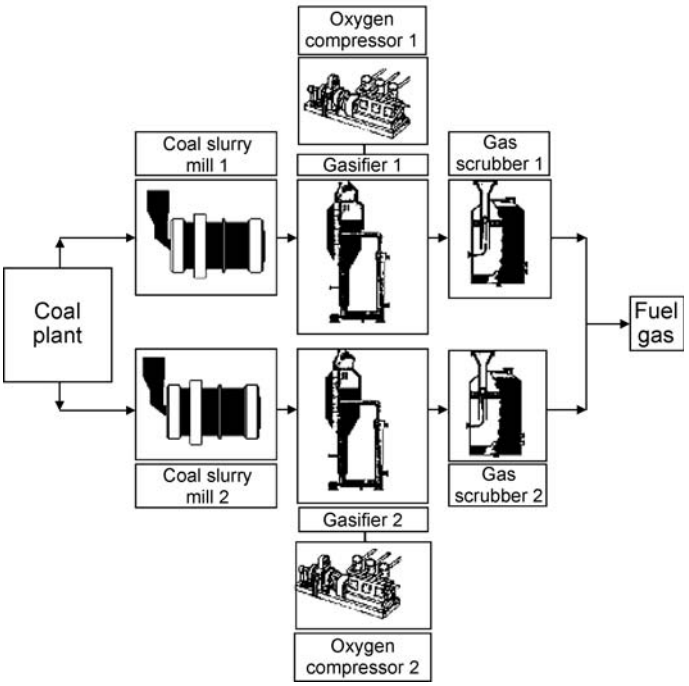


Fig. 4.27 Coal gas production and clarifying plant schematic block diagram

level of detail required for the identification and evaluation of use. Typically, the systems analysis questions are:

WHAT are the sub-system's assemblies (or components)?
FOR WHAT PURPOSE does the assembly (or component) exist?
WHY does the assembly (or component) exist?
WHERE does the assembly (or component) feature?
WHEN does the assembly (or component) feature?

Additional information concerning the requirement for the item would include the following:

- The type of assembly (or component).
- The structure and content of the assembly (or component).
- The relationships of the assembly (or component) to others in the same level of hierarchy.
- The degree to which the assembly (or component) is incompatible with others in the same level of hierarchy.

From the coal slurry preparation plant point of view, the plant can be divided into independent sub-systems to simplify accounting for partial outages. Each of the sub-systems must meet the following requirements:

- It must be binary, i.e. it must be either available or unavailable with no partial outages.
- Its failures and repairs must occur independently of what happens in the rest of the plant.
- It must interconnect with other sub-systems only at its end-points, as represented on an *availability block diagram (ABD)*.

An availability block diagram (ABD) shows how sub-systems or assemblies are grouped schematically into blocks and interconnected from the standpoint of representing a series logic for availability.

The sub-systems or assemblies, depending on the level of detail required of the ABD, are functionally related to or have a functional dependence on one another. It is this functional dependence that is shown in an ABD, and not the physical connections between the sub-systems or assemblies. The blocks within an ABD are basic sub-systems. A basic sub-system is an aggregation of one or more assemblies logically linked together to define how their failures can cause failure of the basic sub-system.

Functional analysis Before quantitative values can be assigned to measure the effectiveness of systems operation, an analysis must be made of the functions that the system performs in the application of the sub-system's assemblies (or components). This analysis starts with a statement of boundary conditions and desired inputs and outputs, then proceeds to a list of functions or operations that must be performed. Each function in a system possesses inputs and outputs. Inputs and outputs of functions are matched to determine the required sequence of operations or flow. The problems that exist at the interface between functions are the most important to

be resolved in systems engineering analysis. The analysis of system function inputs, outputs, and their relationships is essential to be able to resolve any interface boundary problems.

Block diagramming is an important and useful technique in functional analysis. It shows inputs, outputs, relationships, flow, and the functions to be performed at each stage of the system. Block diagrams show specific relationships of one stage of a system to another. Different block diagrams can be developed, such as:

- *Process flow block diagram (PFD)*: these diagrams indicate how inputs are transformed at each stage into outputs that, in turn, become the inputs to the next stage. The major characteristic of a *PFD* is that it depicts flow.
- *Availability block diagram (ABD)*: an availability block diagram is somewhat related to a process flow diagram but is intended to show how systems or subsystems are interconnected in an availability sense. The level of detail of an availability block diagram should be as simple as possible, including the following:

  (1) Availability data can be estimated for systems or sub-systems defined at that level.
  (2) Systems or sub-systems defined at that level can be considered binary, i.e. they are either available or unavailable.

- *Reliability block diagrams (RBD)*: in establishing reliability analysis of a complex systems group, it is almost impossible to analyse the plant or systems group in its entirety. The logical approach in reliability analysis is to apply a systems approach.

A systems approach in block diagramming is where the plant or systems group is broken down into its systems hierarchy to that level where it would be correct to assume that the individual elements of the system's hierarchy are binary—in other words, that they can be regarded as being functionally operational, or having functionally failed. This binary state is usually found at the component level of the system's hierarchy. Subdivision of the two possible states of components, i.e. working or not working, on or off, etc., can be represented in a block diagram.

b) Reliability Block Diagrams

There are two types of reliability block diagrams, depending on the complexity of the interconnectivity of the system's components:

Series configuration reliability block diagram The simplest and perhaps most common systems structure in reliability analysis is the series configuration in which the functional operation of the system depends on the proper operation of all its components. Failure of any component in a series configuration causes the entire system to fail. A series configuration reliability block diagram and its related series reliability graph are illustrated below (Fig. 4.28a,b).

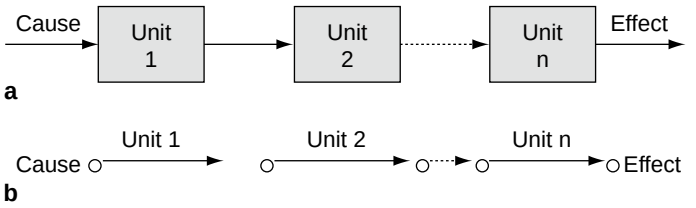


Fig. 4.28 **a** Series reliability block diagram. **b** Series reliability graph

Parallel configuration reliability block diagram In many systems, several functional flow paths perform the same operation. In other words, the system has inherent redundancy or parallel functional paths. If the system's configuration of components is such that failure of one or maybe more components in a specific parallel path still allows the system to function properly, then the system can be represented by a parallel configuration block diagram, indicating the various parallel functional paths. This is sometimes called a redundant configuration.

In a parallel configuration, the system is operational if any one of the parallel functional paths is operational. Failure of any component in a parallel configuration does not cause the entire system to fail but can result in degradation of system performance.

A parallel configuration reliability block diagram, together with its related parallel reliability graph, is illustrated below (Fig. 4.29a,b).

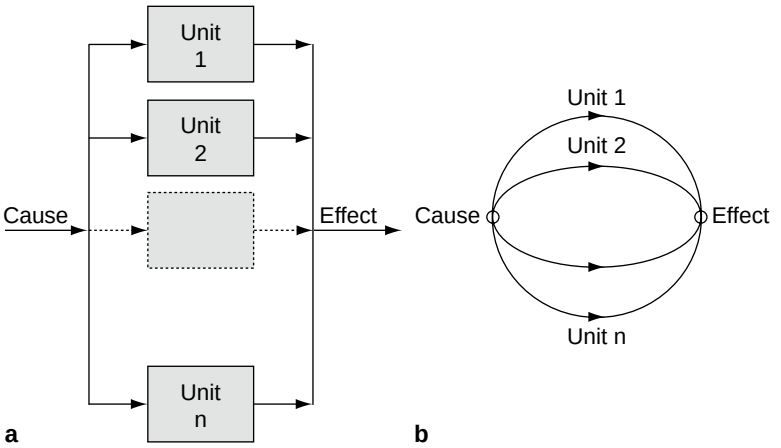


Fig. 4.29 **a** Parallel reliability block diagram. **b** Parallel reliability graph

c) Availability Block Diagrams

On the basis of the definition of a system, and on the basis of the interconnectivity of the various systems, an availability block diagram (ABD) for the example coal slurry preparation plant can now be developed. As indicated previously, an ABD is somewhat related to a process flow diagram but is intended to show how components are interconnected in an availability sense. The coal slurry preparation plant is divided into the smallest possible number of sub-systems, such that each one meets the requirements criteria. Every set of identical sub-systems forms a sub-system group. Figure 4.30 is a block diagram version of the process flow of the coal slurry preparation plant.

The first step in dividing the plant into sub-system groups is to develop an ABD of the plant. Although the oxygen feed is not directly connected to the slurry preparation in the process flow diagram, the two can be connected in the ABD because, if the oxygen feed fails, the corresponding sub-systems will all be inoperable.

Thus, the ABD shows these four sub-systems connected in series (Fig. 4.31).

The level of detail chosen for drawing an ABD should be as simple as possible, subject to the following:

- Data are obtainable or can be estimated for each sub-system defined at that level.
- Each sub-system defined at that level may be considered either available or unavailable.

Each sub-system's process capacity, in terms of the percentage of the plant's process flow that the sub-system should support, is also shown because this information will be used to divide the plant into sub-systems and to define their states. Two further

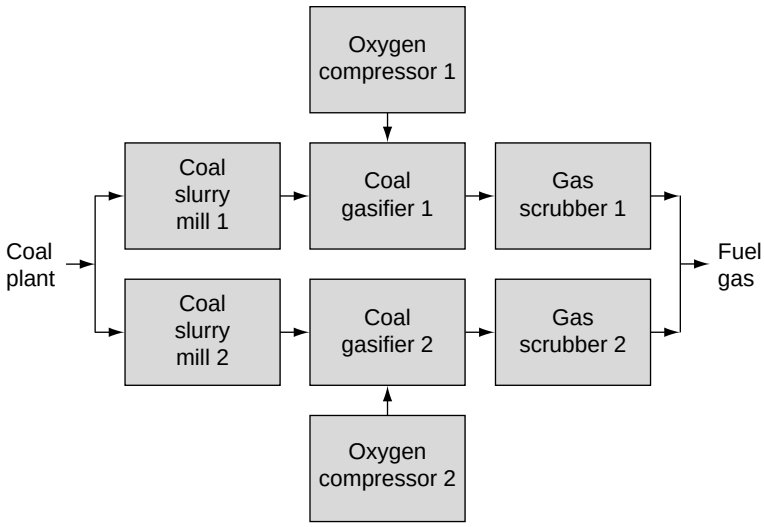


Fig. 4.30 Process flow block diagram

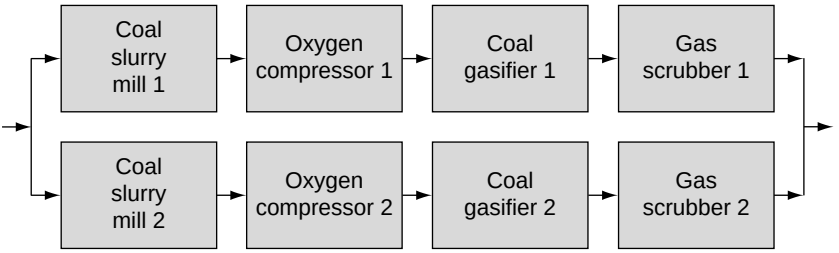


Fig. 4.31 Availability block diagram (ABD)

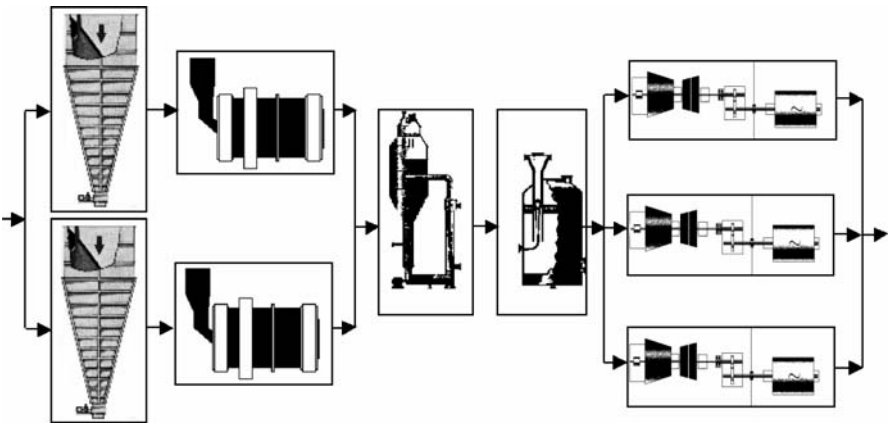


Fig. 4.32 Simple power plant schematic process flow diagram

examples are given for the development of availability block diagrams from process flow diagrams. In the first example, Fig. 4.32 shows a simple process flow block diagram for a simple power plant, and Figs. 4.33 and 4.34 show the development of the ABD.

Example of a simple power plant process flow and availability block diagrams Consider the development of an ABD and further systems engineering analysis for a simple configuration of a power plant consisting of:

Two coal-handling bins.
Two coal grinding mills.
A gasifier and gas scrubbing system.
Three gas turbines.
Three generators.

Figure 4.33 shows that there are cross connections before (X1) and after (X2) the coal-handling bins, after the coal grinding/slurry mills (X3), before the gas tur-

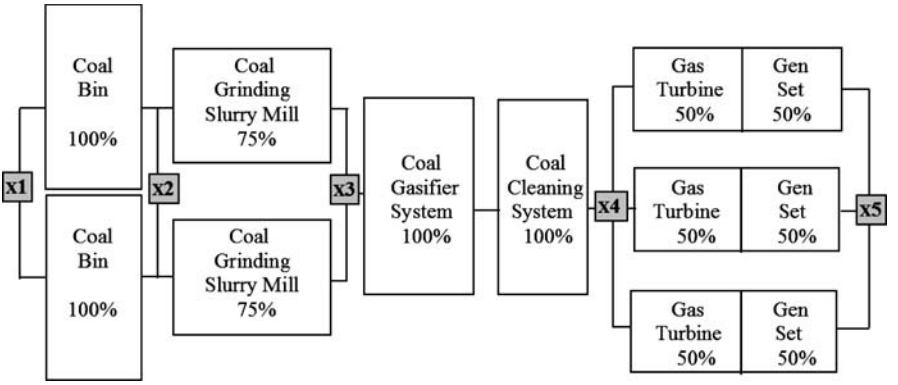


Fig. 4.33 Power plant process flow diagram systems cross connections

bines (X4), and after the generators (X5). Every point on the process flow diagram where all systems or sub-systems are cross connected is marked. Each cross connection in the process flow diagram is numbered and marked with an X.

The significance of a cross connection is that any system on one side of a cross connection can feed, or complement, any equipment on the other side. In the example, either coal-handling bin can feed either coal grinding/slurry mill. Either coal grinding/slurry mill can then ultimately feed, via the gasifier, any of the three gas turbines. All of the systems that have a process flow link along each path between the cross connections are then bound by a hatched boundary line, as indicated in Fig. 4.34. The diagram shows that one coal grinding/slurry mill is a path between cross connections 1 and 2. Similarly, one gas turbine and generator is a path between cross connections 4 and 5. Each set of systems bounded in this way forms a separate group or subgroup of systems. Thus, the two coal-handling bins are grouped with the gasifier and gas cleaning systems to form one system group (A). Each group is then marked with a one-letter designator (A, B or C). Identical groups are given the same designator to form a common system group, such as the three identical 'C' subgroups. The groups thus developed will be binary in operation (i.e. either available or unavailable), and will not contain cross connections to other groups. Furthermore, all 100% capacity systems are grouped together, regardless of their configuration.

In the example, there are three sub-system groups (A, B and C) and six subgroups (one of A, two of B, and three of C) out of a total of 12 systems, as indicated in Fig. 4.34.

The A sub-system group contains one *subgroup* ($1\times$A), which consists of four sub-systems, i.e. the two coal-handling bins, the gasifier and the gas scrubber (Table 4.4). The B sub-system group contains two *subgroups* ($2\times$B), i.e. the two coal grinding and slurry mills. The C sub-system group contains three *subgroups* ($3\times$C), each with two systems, namely a gas turbine and generator.

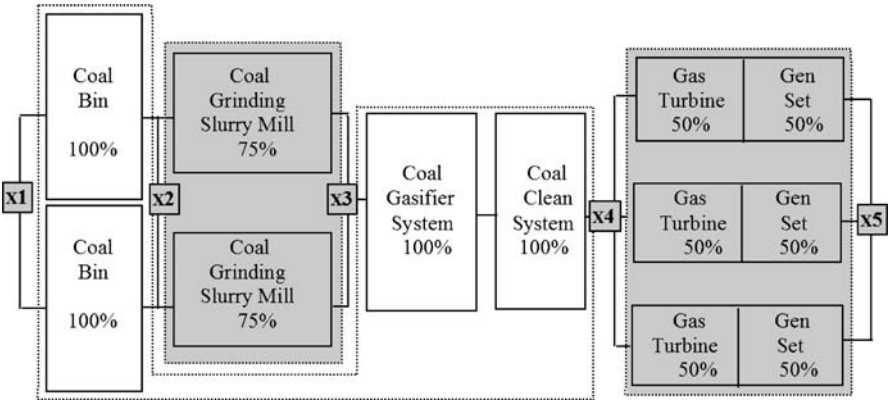


Fig. 4.34 Power plant process flow diagram sub-system grouping

Table 4.4 Power plant partitioning into sub-system grouping

Sub-system group	Number of subgroups	Subgroup contents
A	1	2 × coal bins, 1 × gasifier, 1 × gas scrubber
B	2	2 × coal grinding, slurry mills
C	3	3 × gas turbines, 3 × generators

d) Effectiveness Measures

Before considering any systems constraints for defining the various plant states, it is necessary to establish a set of measures, or criteria, by which the effectiveness of the complex integration of systems can be evaluated. From Eq. (4.27), *process effectiveness* was defined as the design's manufactured and/or installed accomplishment against the design's intended capability.

Effectiveness is a measure of installed output against designed output. Furthermore, from Eq. (4.118) a system's *maximum dependable capacity* was indicated to be equivalent to *process output* at 100% *utilisation*. The following system variables are thus applicable in formulating process (and, therefore, design) effectiveness:

- Utilisation
- Capacities
- Volumes
- Rates.

For the example, capacities are considered as the measure by which a complex integration of systems can be evaluated. All the possible states that the plant can be in are defined in terms of the resultant capacity measures from the plant's systems that are available, and those that are unavailable, in each state. The grouping of sub-systems in the simple power plant example allows for the process of defining the states of plant operation in terms of which *subgroups* are either available or un-

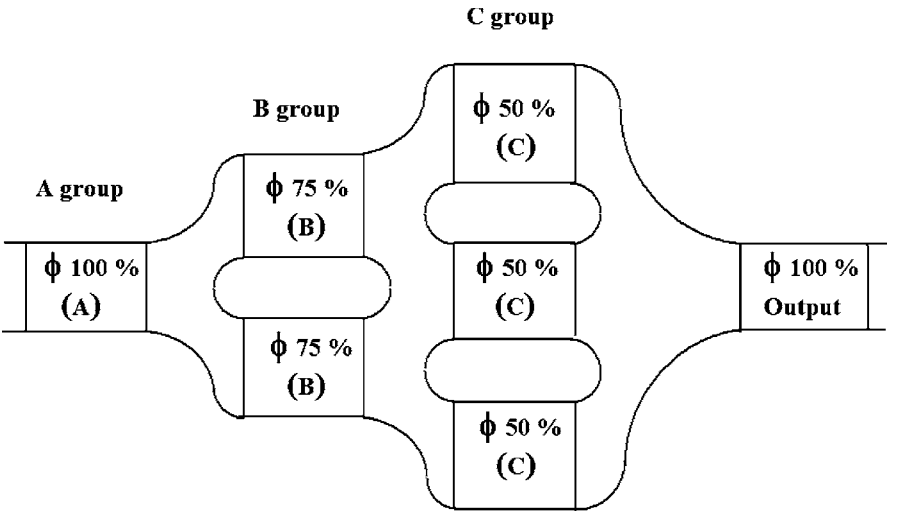


Fig. 4.35 Simple power plant subgroup capacities

available (i.e. binary). One plant state occurs if every subgroup in every sub-system group were available for operation. Another would occur if one of the two B subgroups were unavailable. Also, another plant state would occur if the A subgroup and two C subgroups were unavailable. These are only three of the possible states for the example plant. There are, in total, six possible states that the plant can be in. The system dividing process allows each state to be defined in terms of the number of subgroups in each of the sub-system groups that are unavailable.

A *state* is defined as "*one or more combinations of unavailable and available systems that result in a specific plant effectiveness capability*".

e) Constraints Evaluation

A major part of the systems engineering analysis task is the definition of the boundary between a system and its environment. As indicated previously, this task involves the clarification and establishment of the parameters of the problem, and definition of the specific areas within the general system to be studied. In addition to the boundary conditions, there are some added limits called *constraints*. These include all other aspects that limit or fix many of the external and internal properties of the system. The identification of constraints together with their impact on system effectiveness is an extremely important, yet often overlooked aspect of analysing engineering design problems. Constraints may be classified according to their areas of impact, i.e.:

- Utilisation limitations
- Capacity limitations
- Volume limitations

Table 4.5 Process capacities per subgroup

Sub-system group	No. of subgroups	Capacity per subgroup
A	1	100%
B	2	75%
C	3	50%

- Process limitations
- Quality limitations, etc.

Each state in the simple power plant example has only one subgroup that is the limiting factor, or bottleneck, for the plant's power output capability in that state. This constraint is illustrated in Fig. 4.35 where the example plant is represented as a set of pipes and valves of varying capacities. Each section of pipe and valve corresponds to a subgroup in which the subgroup's unavailability is analogous to a valve being closed:

- The single A subgroup (consisting of two coal bin sub-systems) is wide enough to handle 100% of the flow;
- Each of the two B subgroups (consisting of the two slurry mill sub-systems) is wide enough to handle 75% of the flow;
- Each of the three C subgroups (consisting of three gas turbines and three generators) is wide enough to handle 50% of the flow.

For example, if two C subgroups are unavailable, and one B subgroup is unavailable, the C subgroup is the limiting factor because its remaining capacity is only 50%, whereas the remaining capacity in any one of the B subgroups is 75%. Furthermore, when two C subgroups are unavailable, there could be either *no* unavailable B subgroups or *one* unavailable B subgroup, without further reducing the process flow from the resulting 50% output brought about by the one available C subgroup.

f) Defining Different States

(1) Table 4.5 shows the percentage of the plant's process flow capability that each type of subgroup could support.

(2) Table 4.6 shows the reduction in plant flow capacity as the number of unavailable subgroups in each sub-system group increases, given that all other subgroups are available. Where excess capacity beyond 100% exists in a subgroup, 100% is given as the throughput capacity.

(3) Table 4.7 shows the flow capacities and state definitions. The flow capacities are taken from the previous table. Note that, although the 100% entry appears four times, it is entered only once in the table below. All flow capacities other than 100% are entered as many times as they appear in the previous table. Thus, the 0% flow capacity is entered three times. The capacities should be entered in decreasing order

Table 4.6 Remaining capacity versus unavailable subgroups

Sub-system group	Number of subgroups	Subgroup capacity	Remaining capacity as n subgroups become unavailable			
			$n=0$	$n=1$	$n=2$	$n=3$
A	1	100%	100%	0%		
B	2	75%	100%	75%	0%	
C	3	50%	100%	100%	50%	0%

Table 4.7 Flow capacities and state definitions of unavailable subgroups

State number	Flow capacity	Unavailable subgroups		
		A	B	C
1	100%	0	0	0 or 1
2	75%	–	1	–
3	50%	–	–	2
4	0%	1	–	–
5	0%	–	2	–
6	0%	–	–	3

to simplify the state definition process. The entries under columns A, B and C in Table 4.7 must be the same as the entries under columns $n=0$, $n=1$, $n=2$ and $n=3$ in Table 4.6.

Process of entering the different state definitions

i) Enter for each sub-system group the number of unavailable subgroups that would still allow 100% process flow. In the example, no unavailable subgroups in sub-system group A would allow for 100% process flow. Similarly, no single subgroup in sub-system group B would allow for 100% process flow. In sub-system group C, either zero or one unavailable subgroup allows for 100% process flow.

ii) Enter for each state the number of unavailable subgroups in the appropriate sub-system group that are responsible for that state's capacity. For example, the 75% capacity of state 2 is the result of one of sub-system group B's subgroups being unavailable; the 50% capacity of state 3 is the result of two of sub-system group C's subgroups being unavailable; the 0% capacity of states 4, 5 and 6 each is the respective result that one of A's subgroups, or two of B's subgroups, or three of C's subgroups are unavailable. This is indicated in Table 4.8.

iii) For each state that has a non-zero flow capacity, enter the subgroups in each remaining sub-system group that could be unavailable without further decreasing the flow capacity of that state. For example, state 3 has a 50% flow capacity because of unavailability of two of C's subgroups. Zero subgroups of sub-system group A can be unavailable, and either zero or one of sub-system group B's subgroups can be unavailable without decreasing the flow capacity of 50% for state 3.

Table 4.8 Flow capacities of unavailable sub-systems per sub-system group

State number	Flow capacity	Unavailable subgroups		
		A	B	C
1	100%	0	0	0 or 1
2	75%	0	1	0 or 1
3	50%	0	0 or 1	2
4	0%	1	–	–
5	0%	–	2	–
6	0%	–	–	3

Table 4.9 Unavailable sub-systems and flow capacities per sub-system group

State number	Flow capacity	Unavailable subgroups		
		A	B	C
1	100%	0	0	0 or 1
2	75%	0	1	0 or 1
3	50%	0	0 or 1	2
4	0%	1	0 or 1 or 2	0, 1, 2 or 3
5	0%	0	2	0, 1, 2 or 3
6	0%	0	0 or 1	3

Table 4.10 Unavailable sub-systems and flow capacities per sub-system group: final summary

State number	Flow capacity	Unavailable subgroups		
		A	B	C
1	100%	0	0	< 2
2	75%	0	1	< 2
3	50%	0	< 2	2
4	0%	1	< 3	< 4
5	0%	0	2	< 4
6	0%	0	< 2	3

iv) The remaining entries to be made are in the 0% capacity states. These remaining entries indicate the number of subgroups that can be unavailable in each sub-system group in conjunction with other sub-system groups, where a 0% capacity state can be defined. This is indicated in Table 4.9. The final summary is indicated in Table 4.10.

g) Evaluating Complexity of the Different State Definitions

One of the more significant challenges of engineering design is to provide a rational account of the uncertainty surrounding the state events of unavailable systems that could be responsible for diminishing a design's capacity and/or performance. Classical probability theory offers a feasible approach but it is burdened with well-known

epistemological flaws, considered in Sect. 3.3.2 (Zadeh 1995; Laviolette et al. 1995). Theories of *fuzzy sets* and *possibility* represent attempts to rectify some of the deficiencies in classical probability theory (Dubois et al. 1993). However, all of these theories fundamentally accept the basic fact that *random variables* form a significant part of uncertainty.

Consider the state events of unavailable systems that diminish the overall capacity of the example power plant: Let x_i represent the sub-system states listed in Table 4.10 where $i =$ states $1,2,3,\ldots,6$. Furthermore, let $y_{\theta j}$ represent the *state events* of *unavailable* sub-system groups that could affect the overall capacity of the example power plant, where the subscripts $\theta =$sub-system group A, B or C, and $j =$ subgroup 1, 2, 3. Individual elements of x_i can then be combined into a primary set of state events of unavailable sub-systems, denoted by X, and the elements $y_{\theta j}$ can be combined into a secondary set of state events denoted by Y.

A graphical representation of these elements is called a *complex*, whereby each x_i element is taken as the *vertex* of a surface formed by connected points representing the possible state events of each related subgroup $y_{\theta j}$, the state event elements of Y, which are called *simplices* (Casti 1994).

Thus, the system states represented by x_i are:

$$X = \{x_1, x_2, x_3, x_4, x_5, x_6\} \tag{4.181}$$

and the possible state events represented by $y_{\theta j}$ are:

$$Y = \{y_{A0}, y_{A1}, y_{B0}, y_{B1}, y_{B2}, y_{C0}, y_{C1}, y_{C2}, y_{C3}\} \tag{4.182}$$

The outcomes of the *compound events* resulting from the integration of systems forming each subgroup (as depicted in the availability block diagram of Fig. 4.27) are given by the *values* (expressed as percentages of the overall capacity of the example power plant) of the system states represented by x_i, and are called *random variables*. According to Table 4.10, outcomes of the *compound events* are:

$$\begin{aligned}
x_1 &= (y_{A0} + y_{B0} + y_{C0}, y_{C1}, y_{C2}, y_{C3}) \\
&= 100\% \\
x_2 &= (y_{B1}, y_{B2}, y_{B1}, + y_{C1}, y_{B1} + y_{C2}, y_{B1} + y_{C3}, \\
&\quad y_{B2} + y_{C1}, y_{B2} + y_{C2}, y_{B2} + y_{C3}) \\
&= 75\% \\
x_3 &= (y_{B1} + y_{C1} + y_{C2}, y_{B1} + y_{C1} + y_{C3}, y_{B1} + y_{C2} + y_{C3}, \\
&\quad y_{C1} + y_{C2}, y_{C1} + y_{C3}, y_{C2} + y_{C3}) \\
&= 50\% \\
x_4 &= (y_{A1}) \\
&= 0\%
\end{aligned}$$

Table 4.11 Unavailable subgroups and flow capacities incidence matrix

State number	Flow capacity	Unavailable subgroups
1	100%	$(y_{A0}+y_{B0}+y_{C0}, y_{C1}, y_{C2}, y_{C3})$
2	75%	$(y_{B1}, y_{B2}, y_{B1}, +y_{C1}, y_{B1}+y_{C2}, y_{B1}+y_{C3}, y_{B2}+y_{C1}, y_{B2}+y_{C2}, y_{B2}+y_{C3})$
3	50%	$(y_{B1}+y_{C1}+y_{C2}, y_{B1}+y_{C1}+y_{C3}, y_{B1}+y_{C2}+y_{C3}, y_{C1}+y_{C2}, y_{C1}+y_{C3}, y_{C2}+y_{C3})$
4	0%	(y_{A1})
5	0%	$(y_{B1}+y_{B2})$
6	0%	$(y_{C1}+y_{C2}+y_{C3})$

Table 4.12 Probability of incidence of unavailable systems and flow capacities

State number	Flow capacity	Unavailable subgroups			Probability of incidence
		A	B	C	
1	100%	0	0	0 or 1	0.100
2	75%	0	1	0 or 1	0.200
3	50%	0	0 or 1	2	0.533
4	0%	1	0 or 1 or 2	0, 1, 2 or 3	0.017
5	0%	0	2	0, 1, 2 or 3	0.017
6	0%	0	0 or 1	3	0.133

$$
\begin{aligned}
x_5 &= (y_{B1}+y_{B2}) \\
&= 0\% \\
x_6 &= (y_{C1}+y_{C2}+y_{C3}) \\
&= 0\%
\end{aligned}
$$

Taking the elements of X to be the *vertices* of the unavailability *complex* of the power plant, and denoting the elements of Y to be *simplices* formed from these vertices, the relation R_Y linking the two sets can be established, such that the pairs of elements $(y_{\theta j}, x_i)$ are in the relation R_Y if, and only if, the possible *state events* of *unavailable* subgroups, $y_{\theta j}$, form part of the elementary system states x_i. Thus, (y_{C1}, x_1) is in R_Y; however, (y_{A1}, x_1) and (y_{B1}, x_1) are not.

Computing all the chains of connections in this complex enables the formation of an *incidence matrix* for R_Y. This matrix is the kind of incidence structure for which classical probability theory works well to express the concept of uncertainty in evaluating the integrity of the design.

The complex of which the simplices are the state event elements of Y represents the *sample space* of the various unavailability states, expressed as percentages of the overall capacities, as indicated in Table 4.11. The probability of system unavailability incidence is given in Table 4.12.

h) Evaluation of Alternatives

At this point in systems engineering analysis, alternative design solutions that satisfy system constraints are developed. Effectiveness measures are initially quantified for each solution without serious consideration of cost. Later, both effectiveness and costs are evaluated. After alternative system configurations have been synthesised and the effectiveness requirements have been established for each alternative, they can be compared. A typical trade-off matrix technique is appropriate. In most studies, the analysis is restricted to an evaluation of cost and to some physical attributes of the system such as reliability, availability, maintainability or safety. It is, however, necessary to analyse cost and effectiveness in monetary terms. An adequate analysis cannot be performed unless both parts of the relationship are evaluated in commensurate terms—i.e. when evaluating on the basis of costs, all comparisons must be kept in terms of costs. Prior to such a cost versus effectiveness comparison, however, it is necessary to determine the physical attributes of the system (i.e. *system integrity*).

The following example indicates how overall system integrity can be determined through systems engineering analysis to obtain the system's sub-system and/or assembly attributes of mean times between failures and failure repair times.

Figure 4.36 represents a process block diagram (i.e. a simplified process flow diagram) of a turbine/generator system.

After the development of an availability block diagram (ABD), the overall integrity of the system can be determined based on the ABD configuration and attributes of the system's sub-systems and/or assemblies (Table 4.13).

An ABD of the super-heated steam turbine/generator system illustrated in the process block diagram of Fig. 4.36 is given in Fig. 4.37.

Determining overall mean time to repair (MTTR system) From the integrity values given in Table 4.13:

$$\textbf{MTTR system} = \frac{\Sigma(\lambda R)}{\Sigma(\lambda)} \tag{4.183}$$

where: $\lambda =$ failure rate

$R =$ repair time (h) .

$$\text{MTTR system} = 39{,}227/370.43$$

$$\text{MTTR system} = 105.9$$

Determining overall mean time between failures (MTBF system) From the integrity values given in Table 4.13:

$$\textbf{MTBF system} = \frac{10^6}{\Sigma(\lambda)} \tag{4.184}$$

$$\text{MTBF system} = 2.699$$

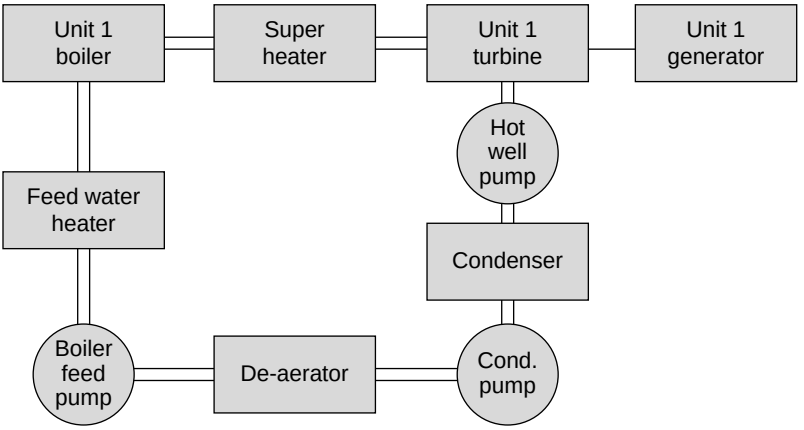


Fig. 4.36 Process block diagram of a turbine/generator system

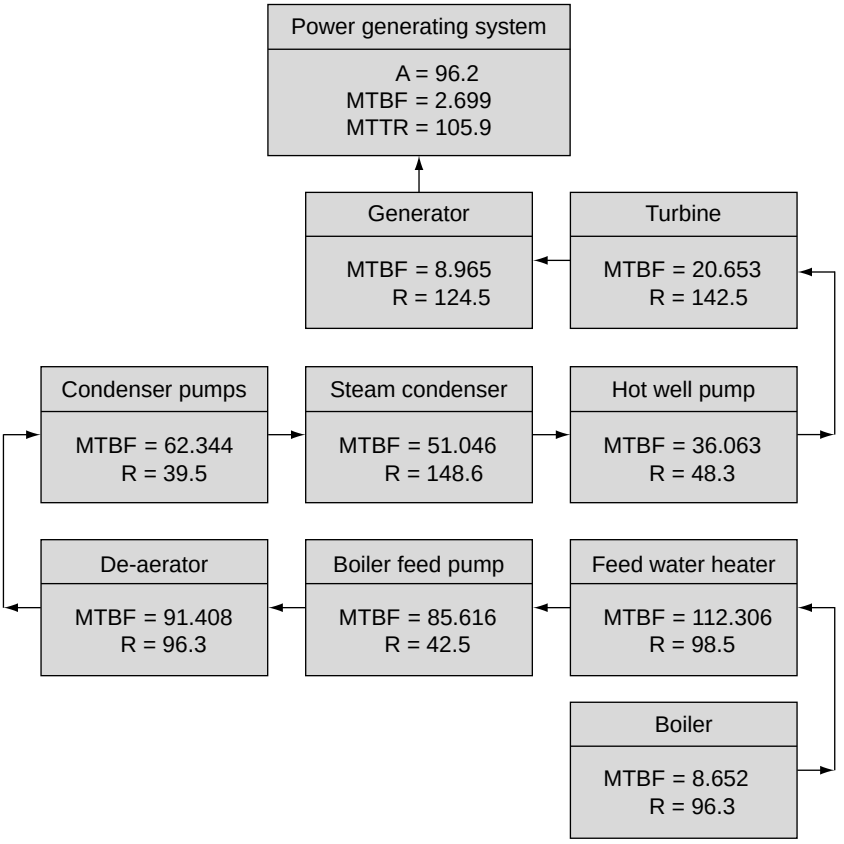


Fig. 4.37 Availability block diagram of a turbine/generator system, where A = availability, MTBF = mean time between failure (h), MTTR = mean time to repair (h)

Table 4.13 Sub-system/assembly integrity values of a turbine/generator system

Power system items	Failure rate (λ fail/10^6 h)	MTBF ($10^6/\lambda$ h)	Repair rate (R, h)	$\lambda \times R$
1. Generator	111.55	8.965	124.5	13.888
2. Turbine	48.42	20.653	142.5	6.900
3. Hot pump	27.73	36.062	48.3	1.339
4. Condenser	19.59	51.046	148.6	2.911
5. Cond. pump	16.04	62.344	39.5	0.634
6. De-aerator	10.94	91.408	96.3	1.053
7. Feed pump	11.68	85.616	42.5	0.496
8. Feed heater	8.90	112.306	98.5	0.876
9. Boiler	115.58	8.652	96.3	11.130
	370.43			*39.227*

Determining overall availability (A system) From the integrity values given in Table 4.13, and from the formula for steady-state availability, we get:

$$\begin{aligned} \mathrm{A} &= \frac{\mathrm{MTBF}}{\mathrm{MTBF} + \mathrm{MTTR}} \\ &= \frac{2.699}{2.699 + 105.9} \\ \mathrm{A} &= 96.2\% \end{aligned} \tag{4.185}$$

where, in Eqs. (4.184) and (4.185):

λ = failure rate
A = availability
MTBF = mean time between failure (h)
MTTR = mean time to repair (h).

4.3.3.4 Evaluating Complexity in Engineering Design

With the phenomenal advancement in process technology, there has been an almost similar increase in the complexity of engineered installations, particularly large integrated systems. Much engineering effort has gone into analysing and understanding *systems complexity* in an attempt to try and manage or reduce it at the design stage. Relatively recent research has shown, however, that the real issue is not so much reducing systems complexity but, rather, reducing *complicatedness*. This is an important distinction because complexity can, in fact, be a desirable property of integrated systems, provided it is specifically *engineered complexity* that reduces *complicatedness* (Tang et al. 2001).

Complexity and complicatedness are not synonymous. Complexity is an inherent property of systems and the integration of systems; complicatedness is a derived

function of complexity, introduced in the notion of *complicatedness of complex systems*. Equations for each can be developed showing that they are separate and distinct properties that not only reflect the fundamental behaviour of complex systems but that also provide a design methodology whereby complicatedness can be evaluated. The implications for systems design engineers are enormous, especially concerning complex systems analysis in engineering design. The difference between complexity and complicatedness can be illustrated by the following example (Tang et al. 2001).

Relative to a manual transmission, a motor vehicle's automatic transmission has more parts and more intricate linkages, making it more *complex*. To the vehicle driver (operator), it is unquestionably less *complicated* but to the mechanic (maintainer), who has to repair it, it is more *complicated*. This illustrates a fundamental fact about systems: *operational control* has an important role on systems to manage their behaviour. Complexity, therefore, is an *inherent property* of systems. Complicatedness is a *derived property* that characterises the ability to *control* a complex system. A system of complexity level C_a may present different degrees of complicatedness K to distinct control units E and F, where:

$$\begin{aligned} K_\mathrm{E} &= K_\mathrm{E}(C_\mathrm{a}) \\ K_\mathrm{F} &= K_\mathrm{F}(C_\mathrm{a}) \end{aligned} \tag{4.186}$$

and:

$$K_\mathrm{E}, K_\mathrm{F} = \text{complicatedness of systems E and F .}$$

a) Complexity in Systems

There is hardly any research on complicatedness and complexity as distinct properties of systems. The focus is on modularisation and integrated interactions with a bias to linear systems and qualitative metrics. Overwhelmingly, the literature considers systems with a large *number of elements* as complex (Suh 1999). Very few studies address *integrated linkages* among the elements (Warfield 2000), and at least one considers their *bandwidth* (Tang et al. 2001). All these factors are *inherent characteristics* of systems; the number of elements, the number of interactions among these, and the *bandwidth* of the interactions determine the complexity of the system. As these increase, system complexity is expected to increase. For example, consider the system $N = \{n_i\}\ i = 1, 2, \ldots, p$ with binary interactions among the elements. Complexity C_N of this system does not exceed p^2, which is denoted by:

$$C_N = O\left(p^2\right)$$

Thus, the system $M = \{m_j\}\ j = 1,2,\ldots,p$ can have complexity:

$$C_M = O(p^k) \quad \text{where} \quad k > 2\,. \tag{4.187}$$

Thus, when M has $\{m_j \times m_r\}_{jr}$ and $\{m_j \times m_r \times m_s\}_{jrs}$ interactions, then $C_M = O(p^3)$.

Furthermore, when M has $\{m_j \times m_r \times m_s \times m_t\}_{jrst}$ interactions, then $C_M = O(p^4)$.

This characterisation of complex systems considers systems with feedback loops of arbitrary *nesting* (i.e. arbitrary loops within loops), and high *bandwidth* (i.e. volume or number) of interactions among system elements. Complexity is a monotonically increasing function, as the size of the system and the number of interactions, as well as the bandwidth of interactions increase (Tang et al. 2001).

In the limit, complexity $\to \infty$. Complexity is thus defined by:

$$C = X^n \sum_b B^b \tag{4.188}$$

where:

X is an integer denoting the number of elements $\{x_e\}\, e = 1,2,\ldots,p$
n is the integer indicated in the relation $O(p^n)$

and:

$$B_1 = \sum_{ij} \lambda_{ij} \beta_{ij} \tag{4.189}$$

$$B_2 = \sum_k \lambda k^{ij} \beta k^{ij} \tag{4.190}$$

where:

λ_{ij} = the *number of linkages* between x_i and x_j
β_{ij} = the *bandwidth of linkages* between x_i and x_j
λk^{ij} = the *number of linkages* between x_k and (x_i, x_j)
βk^{ij} = the *bandwidth of linkages* between x_k and (x_i, x_j).

In general:

$$B_n = \sum_n \lambda\, p^{ijk\ldots n-1} \beta n^{ijk\ldots n-1} \tag{4.191}$$

where:

$\lambda\, p^{ijk\ldots n-1}$ is the number of linkages among x_k and $(x_i, x_j), (x_i, x_j, x_k), \ldots, (x_i, x_j, x_k, \ldots, x_{n-1})$
$\beta n^{ijk\ldots n-1}$ is the *bandwidth of linkages* for x_k and $(x_i, x_j), (x_i, x_j, x_k), \ldots, (x_i, x_j, x_k, \ldots, x_{n-1})$
B_n is a measure of the *capacity* among the n elements of the system.

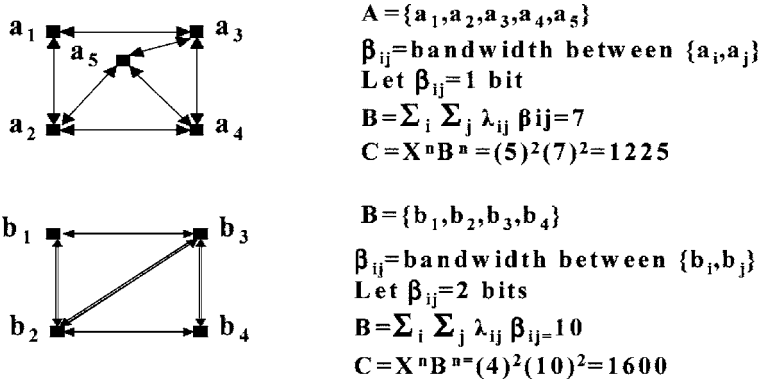


Fig. 4.38 Example of defined computer automated complexity (Tang et al. 2001)

An indicative example of defined computer automated complexity is given in Fig. 4.38 (Tang et al. 2001).

b) Complicatedness as a Function of Complexity

Complicatedness is the degree to which any control over the system is able to manage the level of complexity presented by the system. The means of control can be another system or a person. Complicatedness is a function of complexity, $K = K(C)$. Clearly, at $C = 0$, $K = 0$, the properties of a complicatedness function are essentially the same as those of complexity but they are definitely not identical. For example, consider K when $C \to \infty$. Inevitably, there is a level of complexity at which any means of system control simply cannot cope with the system as a whole. The system then becomes unmanageable through diminished or lack of control.

It is relatively easy to visualise a graph for $g = g(x,y)$ with $C_g = O(p^2)$ (i.e. two-dimensional), and less easy to visualise a graph for $h = h(x,y,z)$ with $C_h = O(p^3)$ (i.e. three-dimensional). However, a surface with four variables is indeed difficult to visualise, although complexity has only reached $O(p^4)$. Consider the incomprehensible systems A and B where $C_a = O(p^{100})$ and $C_b = O(p^{1,000})$. The complicatedness functions are virtually the same in this case, $K_a \approx K_b$, although $O(p^{1,000}) \gg O(p^{100})$. Therefore, when $C = 0, K = 0$ and $C \to \infty$, then $K \to K_{\max}$.

Systems are designed to operate and be controllable at an optimal point of complexity, i.e. C^*. Where $C < C^*$, although complexity increases, it is well within the interval of controllability. Where $C = C^*$, the system complexity is optimal with respect to its ability to be controlled and, where $C > C^*$, complexity is increasing, and the system can be controlled only with decelerating (i.e. exponentially diminishing)

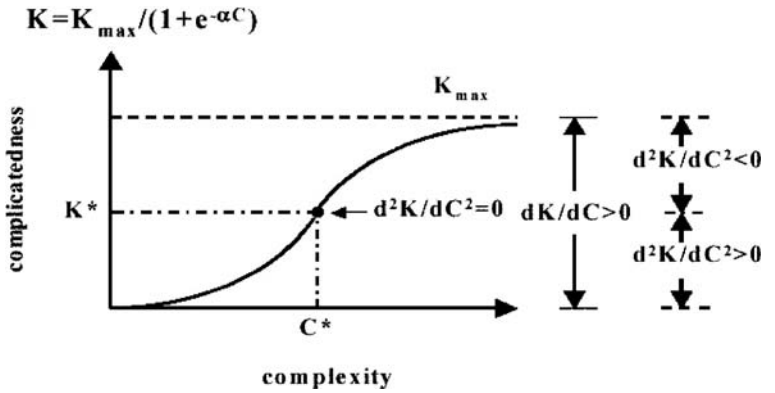


Fig. 4.39 Logistic function of complexity vs. complicatedness (Tang et al. 2001)

effectiveness. This can be expressed mathematically as:

$$\frac{\mathrm{d}K}{\mathrm{d}C} = \{0, \infty\} \quad \text{(i.e. in the open interval between } 0, \infty) \,.$$

$$\frac{\mathrm{d}^2K}{\mathrm{d}^2C} > 0 \quad \text{at} \quad C < C^* \quad \text{where complexity is increasing faster than complicatedness.}$$

$$\frac{\mathrm{d}^2K}{\mathrm{d}^2C} = 0 \quad \text{at} \quad C = C^* \quad \text{where complicatedness has reached an inflection point.}$$

$$\frac{\mathrm{d}^2K}{\mathrm{d}^2C} < 0 \quad \text{at} \quad C > C^* \quad \text{where complicatedness has reached saturation.}$$

For $C < C^*$, $\mathrm{d}^2K/\mathrm{d}^2C > 0$, complexity is increasing faster than complicatedness.

For $C > C^*$, $\mathrm{d}^2K/\mathrm{d}^2C < 0$, the ability to manage complexity has reached diminishing returns.

Because the logistic function is one of the simplest mathematical expressions that has all the properties considered previously, it is adopted to express complicatedness as indicated in the following expression and illustrated in Fig. 4.39 (Tang et al. 2001):

$$K(C) = \frac{K_{\max}}{(1 + \mathrm{e}^{-\alpha C})} \tag{4.192}$$

where:

e	is the transcendental number $\mathrm{e} = 3.27182818284\ldots$
α	is a constant specific to the measure of control
C	is the complexity of the system
$K_{\max} = 1$	indicates absolute complicatedness.

There are other means of expressing complicatedness, such as using the *Weibull distribution*. The major differences, though, are the location of the inflection point, the growth pattern before and after the inflection point, and the symmetry around the inflection point.

c) Designing for Complex but Uncomplicated Systems

The complexity of engineering designs increases relative to the integration of their input vectors. The integration of system elements resulting in new interactions and changes in bandwidth (due to volume or capacity constraints) increases the initial design's complexity. However, *engineered complexity* can reduce intractably complicated input vectors to a minimum number of output vectors that renders the system controllable—and system complexity manageable. The application of *neural networks* is increasingly being considered for process control of complex integrated systems, in situations where there are intractable numbers of data points to analyse. This approach has proven effective for engineering designs in which the process is controlled in real time by adaptive and distributed artificial neural networks (ANN) embedded in distributed control systems. The application of ANN is considered in detail in Sect. 5.3.3.

Earlier, the vehicle transmission was presented as a complex system that is uncomplicated. The automatic transmission presents the *system image* of $A = \{P,R,N,D_1,D_2,D_3\}$, $\lambda_{ij} = 24$, the *number of linkages* between the transmission interactions (four per ratio), and the *bandwidth of linkages* (capacity) between the interactions $\beta_{ij} = 1$; thus, $C_a = (6^2)(24)(1) = 864$ (where $P =$ park, $R =$ reverse, $N =$ neutral and D_1 to $D_3 =$ drive transmission ratios). However, the manual transmission presents the *system image* of $M = \{P,R,N,D_1,D_2,D_3,C\}$ where $C =$ clutch. This needs to be engaged and disengaged, so C's interaction bandwidth is 2. Thus, $\lambda_{ij} = 10$ (two per ratio) with $\beta_{ij} = 1$, and $\lambda_{mn} = 14$ with $\beta_{mn} = 2$. The complexity of the manual transmission is:

$$C_m = \left(7^2\right)\left[10 + (14)\cdot 2\right]^2 = 38{,}416\,.$$

Suppose, for a novice driver, $C^* \approx C_a = 864$ and, at $C \approx 40{,}000$, $K_{\max} = 1$, indicating absolute complicatedness. The analytic form of the complicatedness function for engineering design can now be determined for a system with complexity C and complicatedness K:

- Determine optimal complexity, C^*, which can be optimally controlled.
- At the optimal complexity C^*, set $K^* = 1/2$.
- Solve for α from $K^* = 1/(1 + \mathrm{e}^{-\alpha C^*})$ where $K_{\max} = 1$.
- Determine $K(C) = 1/(1 + \mathrm{e}^{-\alpha C})$.

4.4 Application Modelling of Availability and Maintainability in Engineering Design

In Sect. 1.1, the five main objectives that need to be accomplished in pursuit of the goal of the research in this handbook are:

- the development of appropriate theory on the integrity of engineering design for use in mathematical and computer models;
- determination of the validity of the developed theory by evaluating several case studies of engineering designs that have been recently constructed, that are in the process of being constructed, or that have yet to be constructed;
- application of mathematical and computer modelling in engineering design verification;
- determination of the feasibility of a practical application of intelligent computer automated methodology in engineering design reviews through the development of the appropriate industrial, simulation and mathematical models.

The following models have been developed, each for a specific purpose and with specific expected results, in partly achieving these objectives:

- *RAMS analysis model*, to validate the developed theory on the determination of the integrity of engineering design.
- *Process equipment models (PEMs)*, for application in dynamic systems simulation modelling to initially determine mass-flow balances for preliminary engineering designs of large integrated process systems, and to evaluate and verify process design integrity of complex integrations of systems.
- *Artificial intelligence-based (AIB) model*, in which relatively new *artificial intelligence (AI)* modelling techniques, such as inclusion of *knowledge-based expert systems* within a *blackboard model*, have been applied in the development of intelligent computer automated methodology for determining the integrity of engineering design.

The *process equipment models (PEMs)* for application in *dynamic systems simulation modelling* will now be looked at in detail.

4.4.1 Process Equipment Models (PEMs)

As indicated previously, *process equipment models (PEMs)* have been developed for application in *dynamic systems simulation modelling* to initially determine mass-flow balances for preliminary engineering designs of large integrated process systems. The dynamic systems simulation modelling was developed using the proprietary OOP simulation shell, Extend© (Diamond 1997).

Extend© is a flexible simulation modelling system with a customisable interface where system blocks can be modified or created using a built-in compiled language. It combines the most powerful features of object oriented programming (OOP) for

advanced dynamic simulation with discrete event/continuous system/combined simulation capability, top-down/bottom-up systems hierarchic reachability, animated graphics, advanced statistical and sensitivity analysis, and computer interface with drag-and-drop and point-and-click capabilities.

The PEMs incorporate all the essential preliminaries of process analysis to determine mass-flow balances for preliminary engineering designs of large integrated process systems. The simulation models also incorporate algorithms of process design integrity for assessing reliability, availability, maintainability and safety requirements of process systems. These are incorporated in specific probability distribution modifiers within each PEM. The application of dynamic systems simulation modelling incorporating the PEMs is primarily intended to determine the applicability and capability of simulation modelling during the engineering design stage, in accurately assessing the effect of complex integrations of systems in large engineered installations.

The dynamic systems simulation modelling is based on classic methodology of systems simulation, which is described in detail in the following presentation of the application of computer modelling in engineering design verification. The PEMs have been developed within the Extend© Performance Modelling program (Extend 2001), integrated into a dynamic systems simulation *blackboard model* for application in concurrent engineering design in an integrated collaborative design environment in which automated continual design reviews may be conducted throughout the engineering design process by remotely located design groups communicating via the internet.

Design methodology and dynamic systems simulation The integration of dynamic systems simulation with blackboard design methodology allows for the development and integration of the basic building blocks of systems engineering design that can be represented in a *design knowledge base*. Support systems in the form of general-purpose *design knowledge sources* are similarly developed to support the design knowledge base. The design knowledge base and design knowledge sources form the core of an integrated design support system. The design objects in the design knowledge base can be synthesised to generate conceptual design solutions, as illustrated in Fig. 4.40.

A *dynamic systems simulation blackboard model* (ICS 2002) is developed to control the design knowledge sources and integrate the knowledge-based design applications such as the PEM blocks. The design knowledge base contains design objects, relations, constraints in terms of intended function and interfaces, as well as detailed information in terms of geometry and sizing.

The blackboard model The blackboard model is a paradigm that enables the flexible integration of analytic methodology into a single problem-solving environment. In terms of the type of problems that it can solve, there is only one major assumption—that the problem-solving activity generates a set of intermediate results. This is evident throughout the dynamic systems simulation modelling integrated into the blackboard model, with systems selection in hierarchical structures as illustrated in Fig. 4.41.

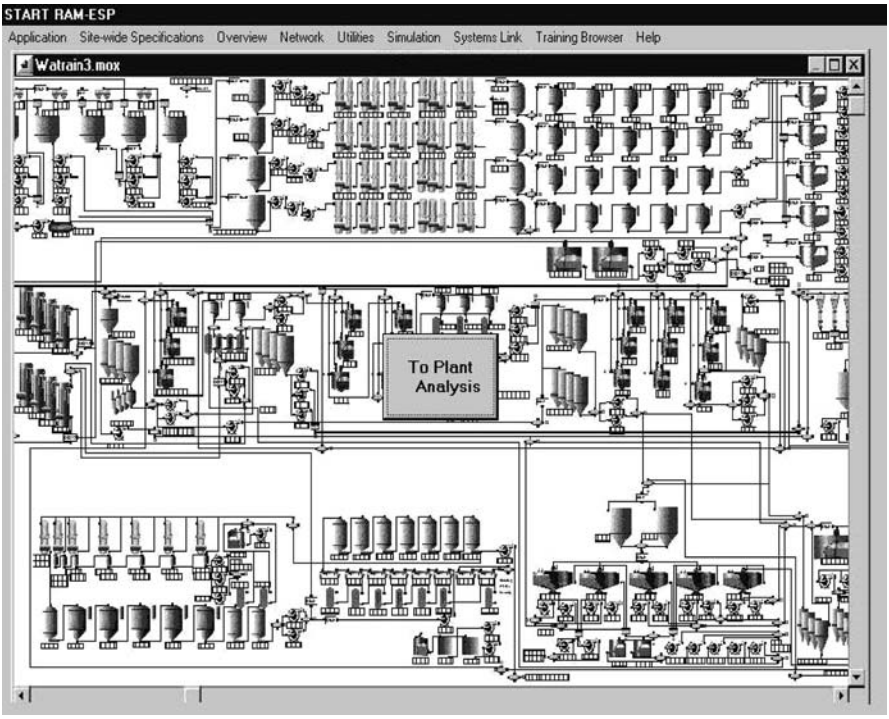


Fig. 4.40 Blackboard model and the process simulation model

The blackboard model consists of a data structure (the blackboard) containing information (the context) that permits a set of modules (knowledge sources) to interact. The blackboard can be seen as a global database or working memory in which distinct representations of knowledge and intermediate results are integrated uniformly. It is also a means of communication among design teams, and can be used as a common display for review and performance evaluation.

Blackboard architecture consists of three major components:

- The *knowledge sources*, which are software specialist modules. Each knowledge source provides specific expertise. The ability to support interaction and cooperation among diverse knowledge sources creates enormous *flexibility* in engineering design.
  Flexibility in this context is the ability to change the blackboard database implementation, the insertion/retrieval strategies, and the representation of blackboard objects without modifying knowledge sources or base data such as design specifications. Flexibility in blackboard architecture for engineering design is important for two reasons. First, understanding of the insertion/retrieval characteristics and the representation of blackboard objects may be uncertain and, therefore, subject to change as the design is developed. Second, even after a schematic model prototype of the design has been completed, the number and placement of black-

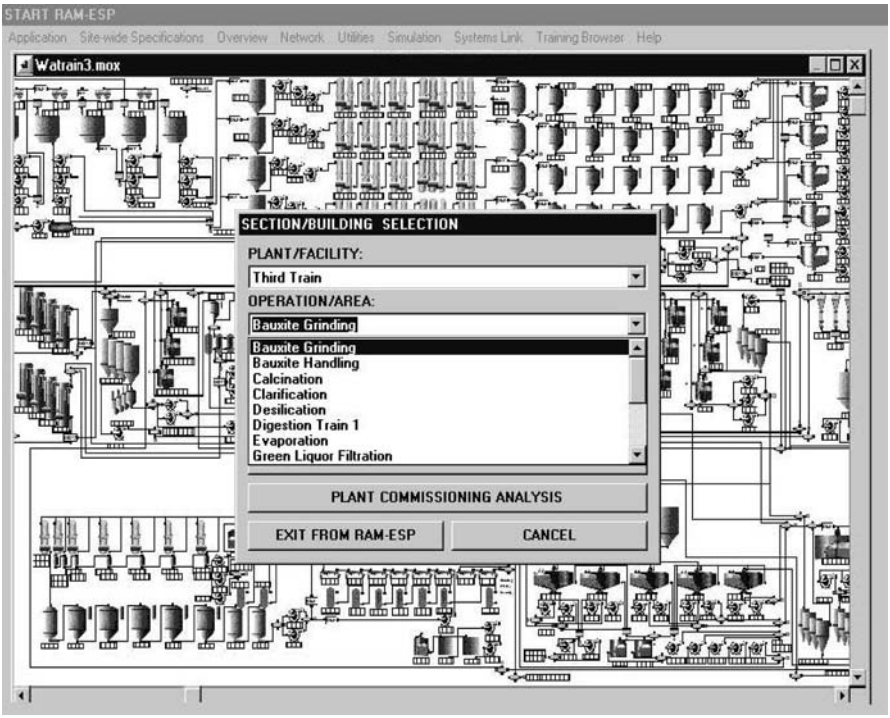
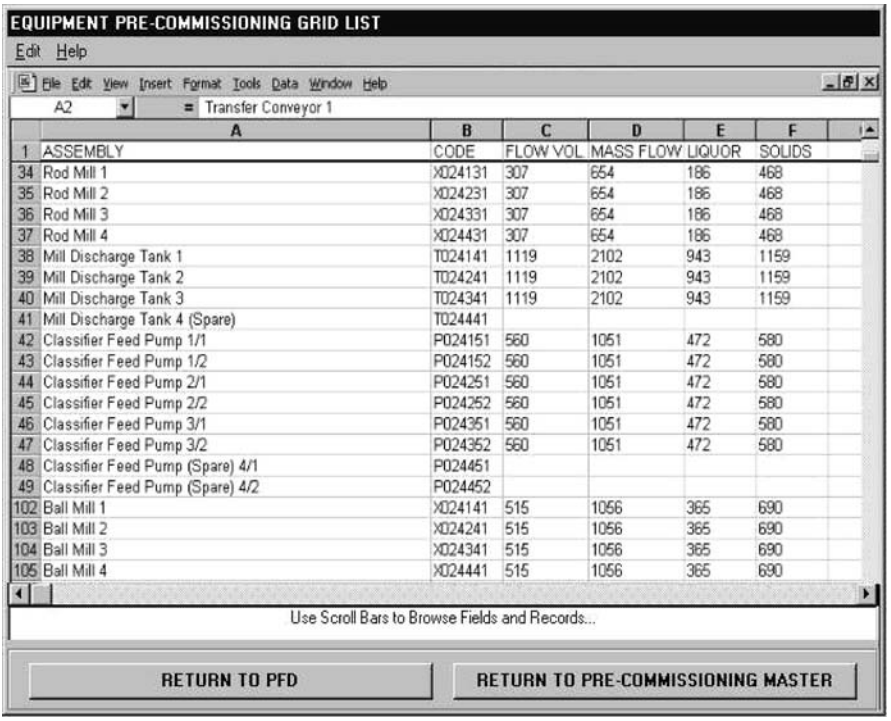


Fig. 4.41 Systems selection in the blackboard model

board objects may differ from those of the prototype. This requires changes to the blackboard representation to achieve the desired level of performance (Corkill et al. 1987).

- The *blackboard*, which is a shared repository of problems, partial solutions, suggestions, and contributed information. The blackboard can be thought of as a dynamic library of solutions to the design problem that have been contributed by other knowledge sources. Thus, a blackboard in engineering design is an approach that allows knowledge sources to cooperate in solving a design problem. This is analogous to a group of designers standing around a blackboard. The blackboard is a database that is used to hold shared information among the participants (or knowledge sources). It may be structured so as to represent different levels of abstraction as well as distinct and possibly overlapping concepts in the design solution. The division of the blackboard into systems hierarchy levels (as with the PEMs) parallels the process of abstraction of the knowledge, allowing elements at each level to be described approximately as abstractions of elements at the next lower level. This partition of the knowledge is useful, in that a partial solution (i. e. group of hypotheses relating to design optimisation) at one level can be used to constrain the design at lower system levels.



EQUIPMENT PRE-COMMISSIONING GRID LIST

Edit Help

File Edit View Insert Format Tools Data Window Help

A2 = Transfer Conveyor 1

	A	B	C	D	E	F
1	ASSEMBLY	CODE	FLOW VOL	MASS FLOW	LIQUOR	SOLIDS
34	Rod Mill 1	X024131	307	654	186	468
35	Rod Mill 2	X024231	307	654	186	468
36	Rod Mill 3	X024331	307	654	186	468
37	Rod Mill 4	X024431	307	654	186	468
38	Mill Discharge Tank 1	T024141	1119	2102	943	1159
39	Mill Discharge Tank 2	T024241	1119	2102	943	1159
40	Mill Discharge Tank 3	T024341	1119	2102	943	1159
41	Mill Discharge Tank 4 (Spare)	T024441				
42	Classifier Feed Pump 1/1	P024151	560	1051	472	580
43	Classifier Feed Pump 1/2	P024152	560	1051	472	580
44	Classifier Feed Pump 2/1	P024251	560	1051	472	580
45	Classifier Feed Pump 2/2	P024252	560	1051	472	580
46	Classifier Feed Pump 3/1	P024351	560	1051	472	580
47	Classifier Feed Pump 3/2	P024352	560	1051	472	580
48	Classifier Feed Pump (Spare) 4/1	P024451				
49	Classifier Feed Pump (Spare) 4/2	P024452				
102	Ball Mill 1	X024141	515	1056	365	690
103	Ball Mill 2	X024241	515	1056	365	690
104	Ball Mill 3	X024341	515	1056	365	690
105	Ball Mill 4	X024441	515	1056	365	690

Use Scroll Bars to Browse Fields and Records...

RETURN TO PFD RETURN TO PRE-COMMISSIONING MASTER

Fig. 4.42 Design equipment list data in the blackboard model

- The *control shell*, which controls the flow of problem-solving activity in the system. Knowledge sources need a mechanism to organise their application in the most effective and coherent fashion. In a blackboard system, this is provided by the control shell.

Knowledge sources Each knowledge source is data-directed, in that the blackboard is monitored for data matching-specific design preconditions. Knowledge sources may be classified in a number of different ways depending on the characteristic that is used to discriminate these. For example, a generic knowledge source may be useful in a whole set of knowledge-based systems (e. g. design equipment list data for application in dynamic systems simulation modelling of a particular design solution, as illustrated in Fig. 4.42), or specific to one application (e. g. specific probability distribution modifiers within each PEM for assessing reliability, availability, maintainability and safety requirements of process systems in a design).

The generic knowledge source in Fig. 4.42 of design equipment list data, for application in dynamic systems simulation models of specific alumina processing stages, gives relevant data of the equipment such as equipment code, flow volumes, mass-flow volumes, liquid volumes and solids volumes.

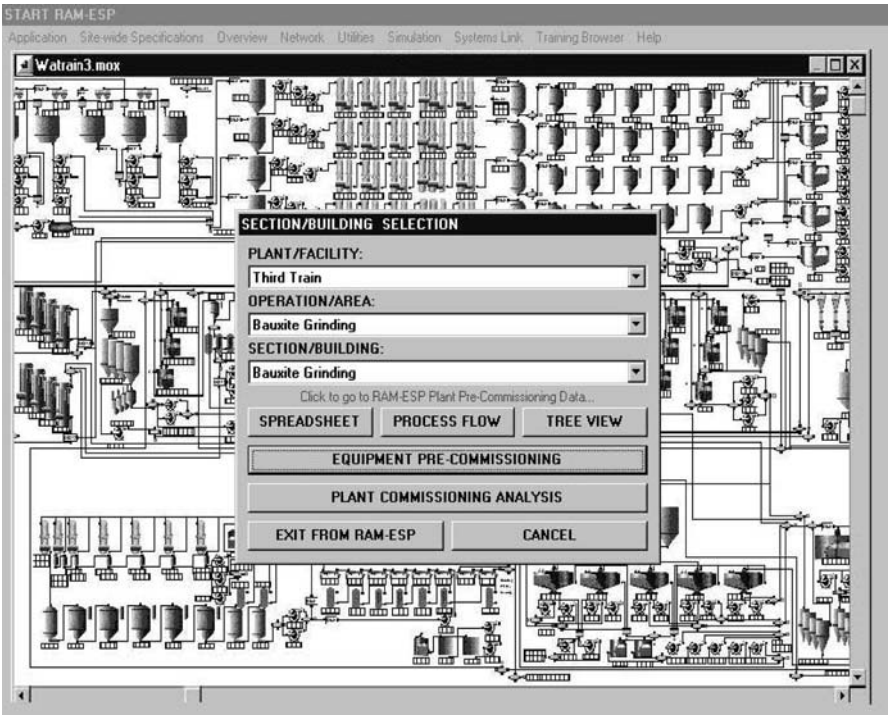


Fig. 4.43 Systems hierarchy in the blackboard model context

The context The context is a set of entries or context elements in the blackboard that contain the information representing the state of the solution process. For example, in the dynamic systems simulation blackboard model, PEMs are selected according to a systems hierarchy, as illustrated in Fig. 4.43. Those entries may include perceptions, observations, hypotheses, decisions, goals, interpretations, judgements or expectations. Also, they may have relationships to one another. In particular, one such organisation may combine a set of entries as the representation of a single object viewed from different levels of abstraction. There can be objects that represent goals, questions and information, knowledge sources, and other general concepts in the blackboard, as well as domain-specific objects.

Figure 4.43 illustrates the selection of information representing the state of the alumina process by plant/facility (third train), operation/area (bauxite grinding) and section/building (also bauxite grinding).

The user interface The user interface permits the interaction of the user (designer) with events inside the blackboard and indirectly with the rest of the knowledge sources comprising the system. This interaction may occur in both directions—by the users modifying the flow of control of the system by means of commands and answers to questions, or by the system informing the user of important events, prompting for answers, or explaining decisions. The user interface manages the question

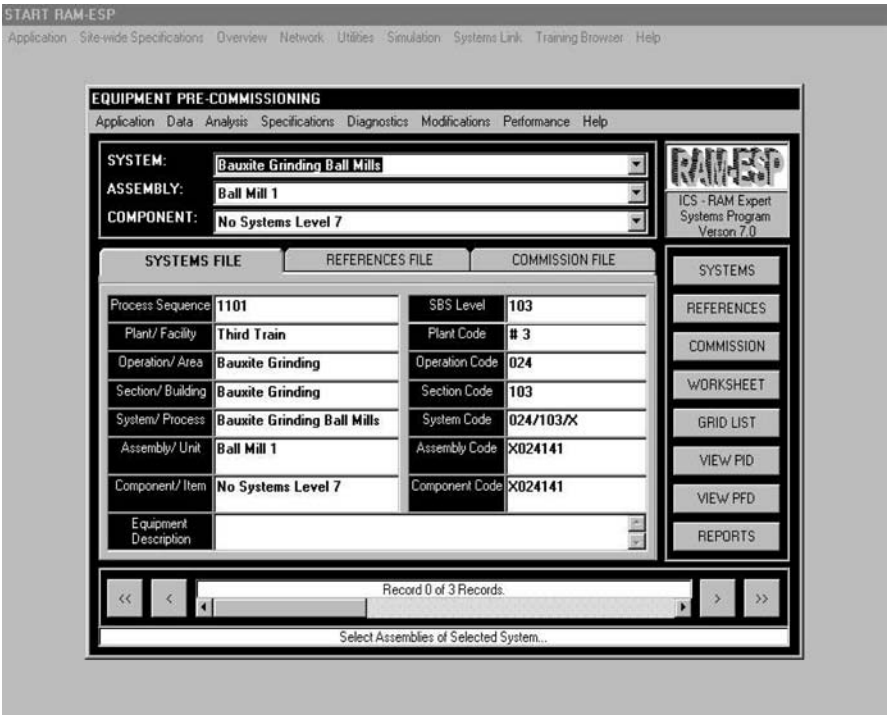


Fig. 4.44 User interface in the blackboard model

and answer protocols, and informs the user of important events during the program's execution. Among its most important capabilities are the following: it checks if an answer is valid, (based on pre-specified or dynamic menus or constraints), advises the user on valid or desirable answers, manages default values, and automatically completes queried answers.

Figure 4.44 illustrates a process pre-commissioning user interface in the blackboard model for information relating to a specific alumina process equipment: the bauxite grinding system, and ball mill assembly.

Dynamic system simulation in engineering design Dynamic system simulation in engineering design provides for typical *virtual prototyping* of engineering processes, rather than experiments on the physical prototype. Not only does virtual prototyping make design verification faster and less expensive but it also provides various design teams in a collaborative design environment with immediate feedback on design decisions. This, in turn, promises a more comprehensive exploration of design alternatives and a better performing final design. To fully exploit the advantages of virtual prototyping, dynamic system simulation is the most efficient and effective. However, these simulation models have to be easy to create. Creating dynamic simulation models is a complex activity that can be quite time-consuming.

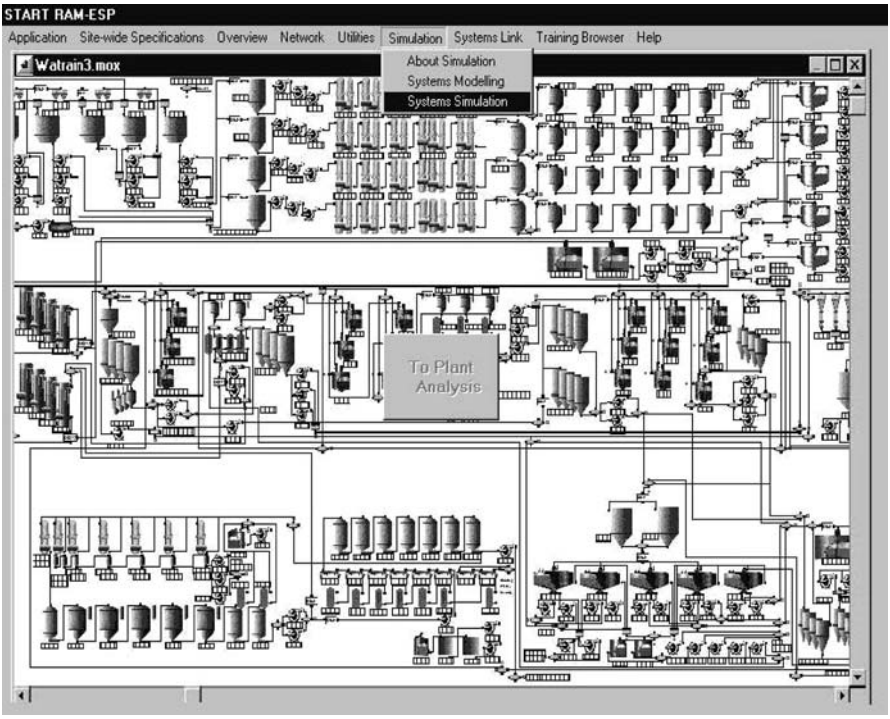


Fig. 4.45 Dynamic systems simulation in the blackboard model

To take full advantage of virtual prototyping, it is necessary to develop a modelling paradigm that supports model reuse, that is integrated with the design environment, and that provides a simple and intuitive user interface that requires a minimum of analysis expertise, as shown in Fig. 4.45.

General configuration of process simulation model Many engineered installations have a *modular architecture* that is based on the optimum selection and composition of systems, assemblies and components from older designs. When the new design is created, these system compositions are selected and then connected together in a *configuration*. In addition to the dynamics of the systems per se, physical phenomena occur at the interfaces between the system's components. These interactions must also be modelled in an integrated framework that supports the following aspects of interaction modelling:

- *Model organisation*: several models can represent a particular physical phenomenon. These models are classified and organised so that the designer is not inundated with choices. An interaction model taxonomy is developed with each (PEM), based on a theoretical formalism to represent and organise the interactions.

- *Model reuse* through standardised representation: all interactions between the system composition and its environment occur through the component's interface. Therefore, a library of interaction models can be indexed by their interfaces. Candidate interaction models can be selected by searching the library for models with interfaces compatible with the interfaces of the connected components.
- *Capturing component interaction dynamics*: when two components are connected via their interfaces, the connection implies that there is an intended physical interaction between the two components. This interaction is captured in the behavioural model and the results represented by graphic display.

Modular architecture is the configuration of a composition of systems showing what types of modules are components of the system, how many components of each type there are, and how components interact.

In *object connection modular architecture*, the component only has interfaces and does not specify what dependencies it has. Thus, dependencies are created implicitly by invoking information from other generic knowledge sources embedded in the blackboard. This has a disadvantage, in that changes made to the initial configuration of the composition of systems modify the modular architecture (e. g. replacing a component causes different connections).

In *interface connection modular architecture*, all dependencies are explicit. Interfaces define what is required in order to function correctly.

Model configuration In many design processes, the target systems are designed using predefined *model components*. In such processes, these model components are selected, configured and assembled in such a way that the design specifications are met. A model component is a modular design entity with a complete specification describing how it may be connected to other model components in a *configuration*. For example, a modelled pump assembly has intake and outlet ports to connect it to other model components on each side. The pump's intake and outlet collectively form the *ports* or *interface* to this component.

A model component is instantiated in the design by specifying *instantiation parameters* that describe its specification. Once instantiated, the model component is connected to other instantiated components via its *ports* or *interface*. Figure 4.46 illustrates several series of model components (in this case, complete PEMs) connected together in a general configuration of the simulated process.

Composition of systems A *configuration* is created when two or more model components are connected to each other via their *interfaces*. A model component can itself encapsulate a configuration of numerous model components, thus allowing for a hierarchical structure of systems in a composition of systems. Multiple configurations can represent a particular system composition, and are bound to the system's configuration interface. For example, a process system can be represented as a single component or as a configuration of several model components. The candidate configurations are all equivalent specifications of the same model components, and the choice of configuration is independent of model behaviour.

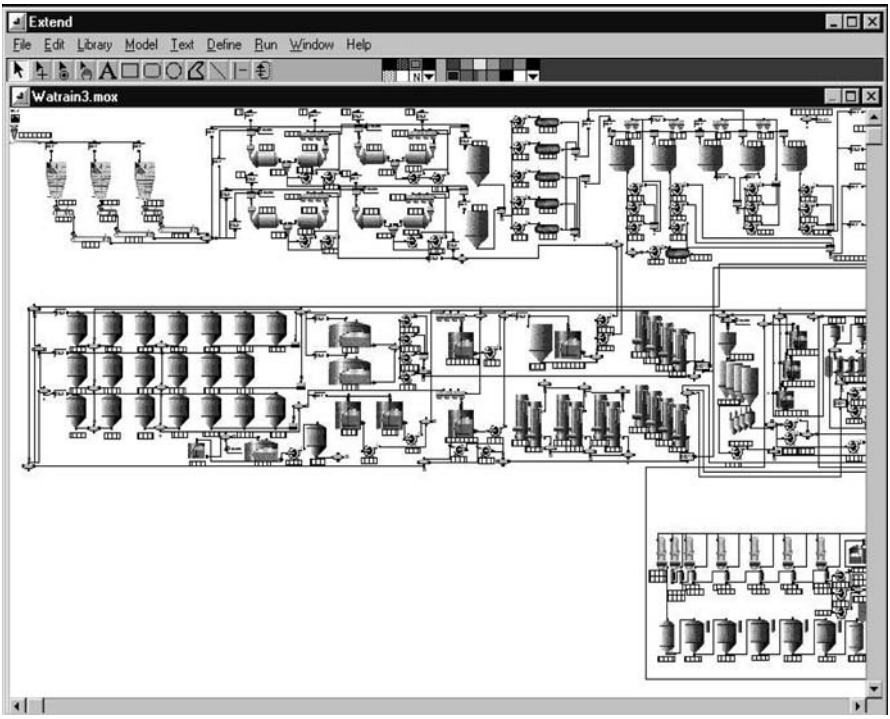


Fig. 4.46 General configuration of process simulation model

Figure 4.47 illustrates the very many compositions of systems of the process simulation model, where each system (PEM), consisting of model components, is connected in a series-parallel configuration with object connection modular architecture at the composition of systems level, and interface connection modular architecture at the model components level.

Logical flows in process equipment models (PEMs) The overall configuration of a comprehensive process simulation model in a composition of systems can include a large amount of PEM blocks, each connected to another in many complex flow configurations. There are two types of *logical flows* between the PEM blocks. The first type of flow represents the *items* that move through the system. Items can be associated with attributes and priorities. The second type of logical flow changes over time during the simulation run and is represented by values. Examples of values include the number of items in a queue, the result of a random sample, or the level of fluid in a tank. In Extend© Performance Modelling, each block (model component) has connectors that are the interface points of the block. Connections are lines used to specify the logical flow from one block to another. Double lines represent item connections and single lines represent value connections. The concept of value

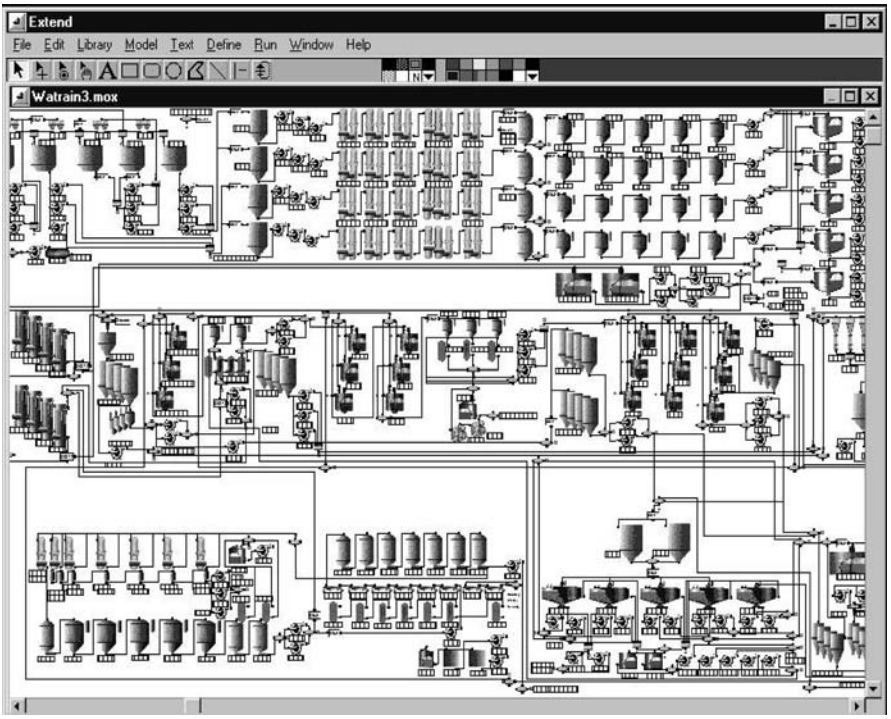


Fig. 4.47 Composition of systems of process simulation model

connections in addition to item connections is unique to Extend©. Other contemporary simulation applications require that a function be written whenever a simulation input is based on a value from another point in the model. In the Extend© Performance Modelling program, this type of logic is performed without programming of any type. More importantly, the logic of the model is visible to anyone examining the model structure. To simplify the appearance of the model, the connections can be hidden.

Optimisation using evolutionary algorithms The focus is on characterising, modelling and organising the interactions between model components. The configuration also contains analysis models, with rules imposed by a set theoretic formalism. The model configuration is based on the context of design specification. This framework also incorporates *optimisation capabilities*. The Extend© Performance Modelling program includes an 'evolutionary optimiser' that employs powerful enhanced *evolutionary algorithms (EA)* to determine the best possible model configuration. Using a drag-and-drop interface, performance metrics and parameters that can be varied are entered into the 'optimiser' block. These parameters are used in an equation that defines the objective function. When the model is run, the 'optimiser'

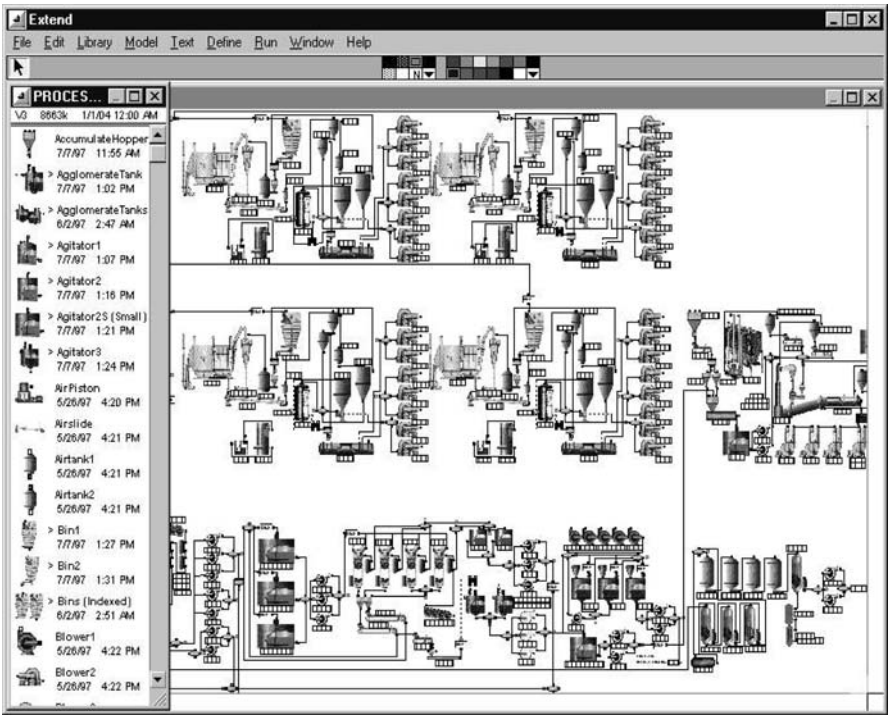


Fig. 4.48 PEM library and selection for simulation modelling

block generates alternatives and locates the statistically best configuration. Unlike external optimisers, the Extend© simulation model's optimisation is well integrated into the program. For example, when the optimisation process is complete, model parameters are automatically set to the optimal configuration. In addition, because the optimiser has been implemented in a block, the source code is available for examination and modification.

Model component library The PEM models have been constructed with library-based iconic blocks. These iconic blocks have been developed using a powerful programming language, namely the Extend© ModL language. Block dialogs are the mechanism for entering model data and reporting block results. Blocks reside in libraries, and each library represents a grouping of blocks with similar characteristics. Blocks are placed on the model worksheet by dragging these from the library window, as illustrated in Fig. 4.48.

Logical flow is established between the blocks. Interfaces, components and graphics are created that tailor the model to a specific application. By modifying an existing interface or creating a new one, the simulation modeller develops a model that can be used by someone more familiar with the system than with the simulation

tool, where each block then describes a calculation or a step in the process. Models can thus be aptly built for the conceptual framework of a collaborative design environment.

Model development programming There are several advantages in a dynamic systems simulation development environment applied concurrently in a collaborative design environment. Design model builders are able to easily and reliably create new or modified modelling constructs for demanding design modelling situations or new applications. The significance of a powerful programming language such as ModL further enhances the simulation modelling capability. Traditional simulation languages or scripting environments typically lack full sets of language features such as flexible condition statements (many are limited to a single condition at a time), user-defined data structures, and user interface development tools. These features are especially suited to the inclusion of dynamic systems simulation as an imbedded knowledge source in a blackboard system. With ModL, only one language and interface needs to be learned and, since ModL is based on the C language, its learning curve is typically short. With less time learning and switching between languages, design model developers are able to develop more sophisticated models in less time. This level of extensibility further prompts designers to develop libraries of custom blocks for specific engineered installations.

The ability of the dynamic systems simulation blackboard model to communicate with other knowledge source applications in the blackboard allows the model to exchange information with the knowledge base and with the expert systems within the blackboard, using inter-process communication (IPC) and dynamic data exchange (DDE) capability. Through IPC, the systems simulation model can act as a server application, allowing the blackboard and other knowledge source programs, such as expert system shells, or an artificial neural network (ANN) application, to request that a specific simulation model perform any task that the ModL language allows.

The dynamic systems simulation blackboard model can also act as a client application, requesting data and services from other programs. For example, an expert system application can start and run a specific simulation model, or the simulation model can instruct a spreadsheet to execute a macro and report the results back through the blackboard. Several features of the Extend© Performance Modelling program provide the means for exchange information communication of the dynamic systems simulation blackboard model, specifically *scripting* and use of the Extend© *notebook* and *cloning* features.

Scripting Scripting is a feature that allows models to be created and/or modified through a suite of ModL functions. With this functionality, designers can create objects that automatically build and modify models. With scripting, designers can develop their own model-building wizards or self-modifying models. This is especially attractive in a blackboard application. Coupled with the ability to communicate with other knowledge source applications in the blackboard, and with the expert systems within the blackboard using inter-process communication (IPC), scripting provides

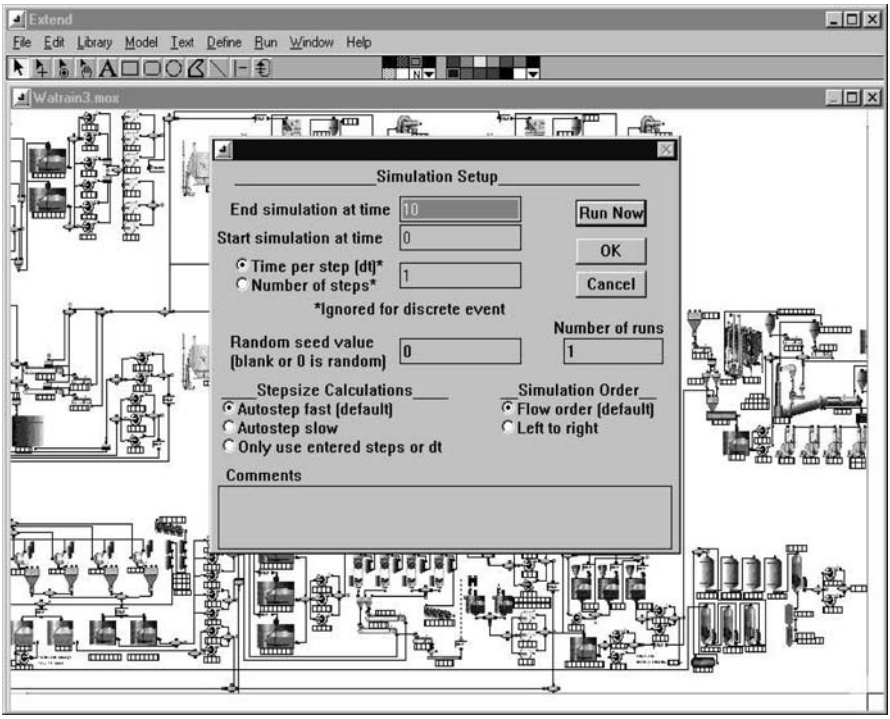


Fig. 4.49 Running the simulation model

an easy method of allowing applications such as the multi-disciplinary collaborative expert systems to control every aspect of the dynamic systems simulation model, including building the model, importing or exporting data, and running the simulation with a specific simulation set-up.

Running the simulation is initiated by accessing the simulation set-up dialog, which then provides the facility of setting various simulation time properties, as illustrated in Fig. 4.49.

Simulation model output Input and output parameters associated with a model can be found in the dialogs of the appropriate blocks or in the output document. While this provides an intuitive association between system metrics and the constructs used to model these, it can make searching for specific data cumbersome. This is especially true when working with large models containing many layers of hierarchy, typical of engineering designs. An effective way of dealing with this is to use the *notebook* and *cloning* feature. With the notebook, a single custom interface is created that consolidates critical parameters, results, and model control to a central location. The notebook is a separate window associated with each simulation model. Initially, the notebook is a blank worksheet to which text, pictures and clones can be added. Clones are direct links to dialog parameters. Once a clone is

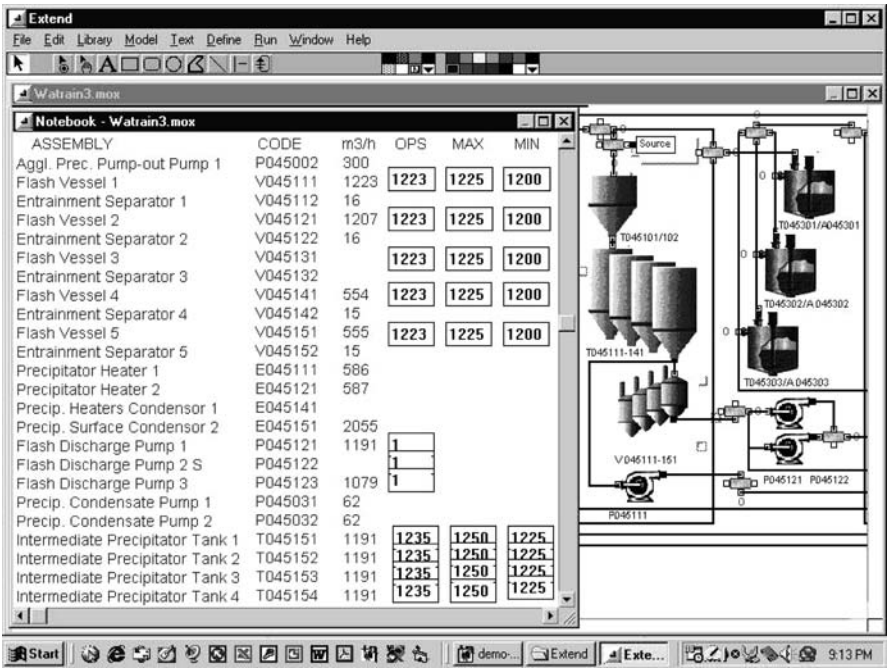


Fig. 4.50 Simulation model output results

created, any changes to the clone are immediately reflected in the block (or PEM) and vice versa.

Creative use of the *notebook* can result in an effective modelling interface for a large, complex engineering design, as illustrated in Fig. 4.50.

4.4.2 Evaluation of Modelling Results

The process of dynamic systems simulation evaluation can be divided into three categories:

- *Model verification*, to ensure that the model's functional behaviour is similar to the real system being modelled;
- *Model validation*, to test the agreement between the *results* of the behaviour of the model and that of the real system, i. e. determining a correlation between the model's results and the real system's output variables;
- *Problem analysis*, which deals with the interpretation of the data generated by the model. In other words, evaluation of dynamic systems simulation is concerned with determining the consistency of *functional behaviour* of the model, its

resulting correspondence with the real system, and the correct *analytic interpretation* of the model's resulting data.

Model verification Model verification implies proving that the model is functionally true according to a set of functional criteria that is applicable for comparison between the model and the real system (i. e. similar types of exogenous, status and endogenous variables). The problem of model verification can thus be reduced to the problem of searching for a set of basic assumptions underlying the functional behaviour of the real system. Such a procedure requires the formulation of a set of postulates or hypotheses describing the behaviour of the real system. This includes the specification of model components and the selection of variables, as well as the formulation of functional relationships, all of which are inherent in the dynamic systems simulation blackboard model that is used to control the design knowledge sources and integrate the knowledge-based design applications such as the process equipment model (PEM) blocks.

The design knowledge base and design knowledge sources form the core of an integrated design support system that enables model verification. The following figures illustrate the design details of these PEM blocks for the various simulation model sectors. In contrast to model verification, the *validity* of a model depends *not* on the formulation of a set of postulates or hypotheses describing the behaviour of the real system but, rather, on the ability of the model to predict the *results* of model behaviour.

Model validation Model validation is the process of developing an acceptable level of confidence that an inference about the results of a simulated process is a valid inference for the outputs of the real system. The problem of model validation can thus be reduced to two characteristic problems: to validate the *results* of a specific model's function, and not the mechanism that generated the results, and to compare the input-output transformations generated by the model to those generated by, or specified for, the real system. In the use of dynamic simulation models to represent real systems, different types or classes of error can result, any one of which can lead to erroneous conclusions, such as errors in model design through the exclusion of significant variables, errors in the modelling approach whereby relevant variables may be represented incorrectly, and errors in programming, input data, or interpretation of results. The validity of the model is made probable, *though not certain*, by analysis of the assumptions underlying the model, whereby the inductive inferences are analysed through statistical methods.

Problem analysis The data generated by computer simulation models represent, in effect, the inductive reasoning of the modelling process as the conclusion of a set of inductive inferences, i. e. assumptions of behavioural results or the outcome of operating characteristics about the behaviour of the real system. The rules for analysing the data generated by computer simulation models are predominantly statistical sampling rules based on the theory of probability. Statistical tests used for analysing these assumptions, and also conclusions of the inductive inferences drawn from simulation runs of the model are, in general, *hypothesis testing* and *estimation*.

Hypothesis testing normally includes the following:

- tests on estimates of parameters assuming an underlying probability distribution (i. e. parametric tests);
- tests on estimates of parameters that are *not* dependent on assuming an underlying probability distribution (non-parametric tests);
- tests to establish the probability distribution from which sample data are generated (goodness of fit tests such as Kolmogorov–Smirnov and Chi-square tests);
- tests on the relationship among variables (correlation analysis).

Estimation includes the calculation of point and interval estimations of parameters, as well as a determination of quantitative equations relating two or more variables (i. e. regression analysis). Statistical methods used for hypothesis testing and estimating are, therefore, mainly tests of means, analysis of variance and covariance, goodness of fit tests, regression and correlation analysis. The results of dynamic simulation models are often used to determine estimates of the parameters of the response variable or, in this case, the flow volume and/or mass flow consisting of liquid and solids. Because these values are estimates, it is essential to assess their accuracy, which is usually done by placing confidence bands or intervals about the estimates. For example, if the simulation model estimate of the mean flow volume of a particular PEM is a value designated by $\bar{E}$, and the design flow volume is μ, an upper limit UL and lower limit LL could be established such that the probability of the design flow volume being the mean of these two limits is equal to a specified exact probability (using the t-distribution as inference).

Dynamic systems simulation case study The case study selected to determine the applicability of dynamic systems simulation modelling in evaluating and verifying process design integrity of complex integrations of systems is a typical alumina refinery process. The data given in Tables 4.14, 4.17 and 4.20 are extracts from a dynamic systems simulation case study (simulation model sectors 1, 2 and 3), and represent preliminary design data of the real system process parameters of flow volume, mass flow, solids content and liquid content used in alumina production. The estimates of the maximum, minimum and mean flow volumes of each PEM in the specific sectors are given in the simulation model's output notebook. Validation of the dynamic systems simulation would include a comparative analysis of the preliminary design data of the real system process parameters, as listed in the tables, with the model parameter estimates of each PEM listed in the model's output notebook. Analysis of the flow volume data generated by the computer simulation model runs would constitute a determination of the *confidence intervals* about the estimates, such that the probability of correspondence with the design flow volume is equal to a specified exact probability. The case study dynamic systems simulation model evaluation for simulation model sector 1 is given in Figs. 4.51 through to 4.55. Case study model evaluation for simulation model sector 2 is given in Figs. 4.56 through to 4.59, and case study model evaluation for simulation model sector 3 is given in Figs. 4.60 through to 4.63.

Table 4.14 Preliminary design data for simulation model sector 1

Assembly	Code	Flow vol.	Mass flow	Liq.	Solids
Transfer conveyor 1	C015141	575	1,508	106	1,403
Transfer conveyor 2	C015241	575	1,508	106	1,403
Rev.shuttle conveyor 1	C024111M	575	1,508	106	1,403
Rev.shuttle conveyor 2	C024211M	575	1,508	106	1,403
Storage bin 1	U024111	192	503	35	468
Storage bin 2	U024211	192	503	35	468
Storage bin 3	U024311	192	503	35	468
Belt feeder 1	Y024121	192	503	35	468
Belt feeder 2	Y024221	192	503	35	468
Belt feeder 3	Y024321	192	503	35	468
Mill feed conveyor 1	C024121	192	503	35	468
Mill feed conveyor 2	C024221	192	503	35	468
Mill feed conveyor 3	C024321	192	503	35	468

a) Evaluation of Simulation Model Sector 1

A major characteristic of the *process flow diagram (PFD) of sector 1* is that it depicts material flow and indicates how inputs are generated and then transformed by each system (or assembly) into outputs that, in turn, become the inputs to the next system (or assembly), as depicted in the preliminary design data given in Table 4.14. These are specifically mass-flow volumes of solids. The PFD is systematically examined to analyse deviations in process flow and system performance and, in this case, to determine mass-flow balance through the integrated assemblies. Each assembly is graphically represented in the simulation model by a virtual prototype *process equipment model (PEM)*.

Each of the assemblies of the PFD depicted in Fig. 4.51 (i. e. the feeder, three storage bins, three chute conveyors, and three transfer conveyors) is a process equipment model.

Each PEM contains selected *model components* that are linked together with logical flows. A model component is a modular design entity with a complete specification describing how it is connected to other model components in a *model configuration*. Model configurations are created when two or more model components are connected to each other via their *interfaces*. Each model component has connectors that are the interface points of the component (or block). Connections are lines used to specify the logical flow between the connectors from one block's output to another's input. Thus, a process system or assembly can be represented either as a single model component or as a configuration of several components.

Logical flow initiation—the random number generator The model components are selected, configured and assembled in such a way that the design specifications of each system are met through the component's attributes, and the linked *logical flows*. Thus, the feeder assembly PEM, for example, has its own specific model configuration, in contrast to that of the storage bins, as depicted in the design details of

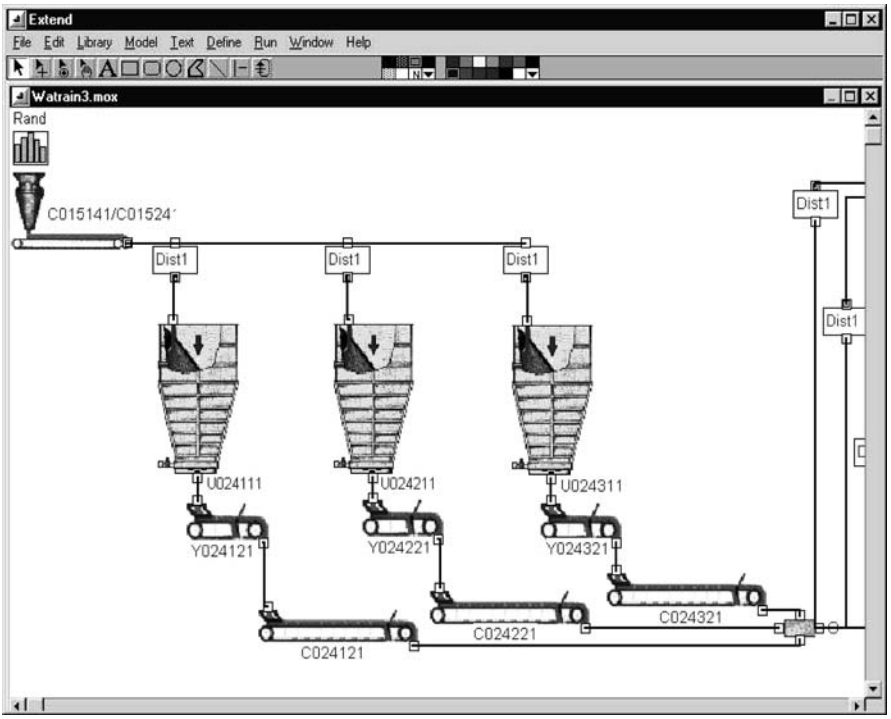


Fig. 4.51 Process flow diagram for simulation model sector 1

Figs. 4.52 and 4.53. However, all simulation models, especially Monte Carlo simulation, have random number generators for 'seed' values initiation of the simulation model's input flow variable(s) that constitute the initial flow of the linked logical flows thereafter.

The model component's attributes depicted in Fig. 4.52 generate random numbers according to a statistical probability distribution, convert the outputs of a conversion function by modifying the component's inputs through a selection of statistical functions, as well as calculating the mean, variance and standard deviation of the component's inputs.

Logical flow storage—the process equipment models (PEMs) Logical flow in the context of process systems simulation modelling represents *upstream material feed* that, in effect, causes the initiation of the process equipment model (PEM). Logical flow storage PEMs are process simulation models in which the model configuration incorporates a model component attribute of an *output conversion function* that modifies the component's inputs through a selection of statistical functions, and statistical probability distributions. As previously indicated, the PEMs incorporate all the essential process analysis preliminaries for preliminary engineering designs of large integrated process systems.

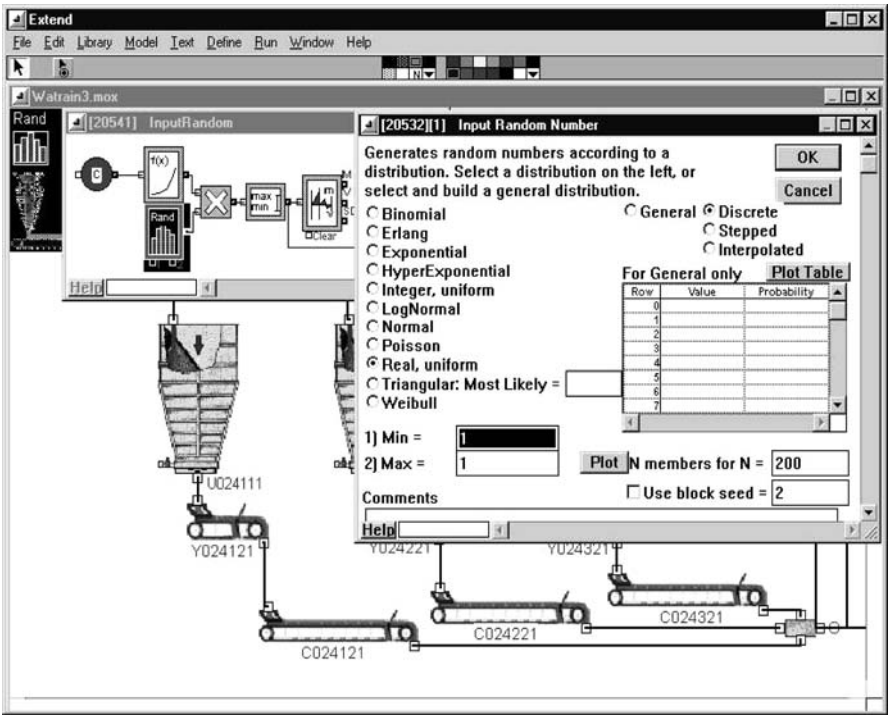


Fig. 4.52 Design details for simulation model sector 1: logical flow initiation

The application of dynamic systems simulation modelling incorporating the PEMs is primarily to determine the effect of logic flow in complex integrations of systems in large engineered installations. The model component's attributes depicted in Fig. 4.53 incorporate probability distribution modifiers of the logic flow within each PEM.

Output performance results The Extend© Performance Modelling program provides a powerfully flexible graphical output presentation through dynamic plotters. These plotters can be placed anywhere in the modelled system configuration, and connected between any of the PEM input/output interface connectors, or within each PEM between model component connectors.

Figure 4.54 illustrates a typical output document showing performance results of the storage bin assembly. These performance variables relate to system or assembly contents, input and output flow quantities, as well as flow surges. The flow surge gives an indication of material flow balancing in the process, subject to upstream material feed. The storage bin PEM illustrated in Fig. 4.54 has a plotter connected to the output model components of the PEM. The plotted graph in the figure shows the trend of material flow through the storage bin from start-up to steady state.

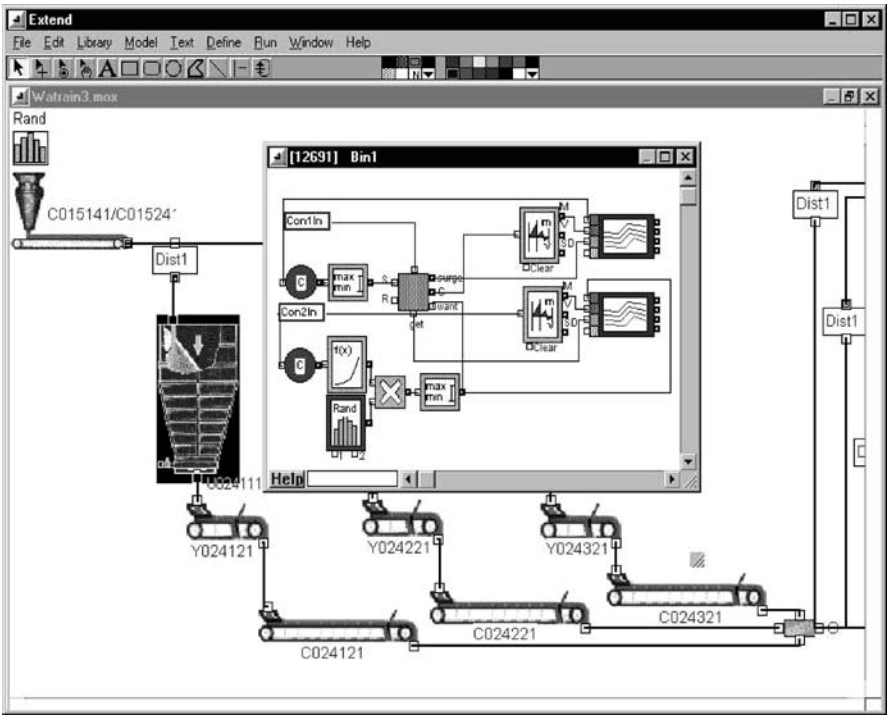


Fig. 4.53 Design details for simulation model sector 1: logical flow storage PEMs

b) Conclusion of Simulation Model Sector 1 Evaluation

Table 4.15 below indicates the values of a comparative analysis of preliminary design data and simulation output data for simulation model sector 1. Column 2 of the table gives the specified preliminary design flow volumes, and column 3 gives the mean of the simulation model's output data. On first scrutiny, these two values are identical with an expectation of a 100% correlation, resulting in the conclusion that the model's output is a perfect match to the specified preliminary design flow volumes of the listed assemblies in simulation model sector 1.

The evaluation of simulation model output data is, however, not that simple, as other factors must be included such as requirements for meeting the full design specification *inclusive of allowable tolerances*, and determining whether the minimum and maximum values, i. e. the range of output variances for each simulation run of the model's output data, fall within the expected confidence intervals of the design specification. The test of whether the simulation model's output variances fall within the allowable design tolerances is set at a 99% level of confidence. The allowable design tolerance for throughput flow volumes is set at $\pm 2.5\%$ of the mean.

Figure 4.55 indicates the simulation model's output for simulation model sector 1, including operational flow throughput (OPS), maximum and minimum flow (MAX) and (MIN), and mean flow output (MEAN). However, an acceptable lower

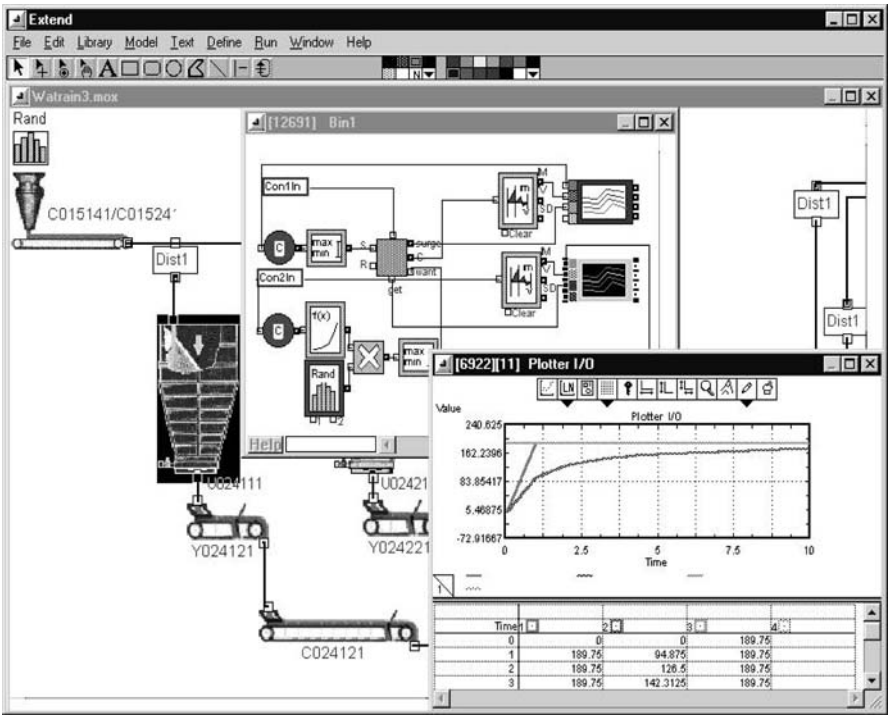
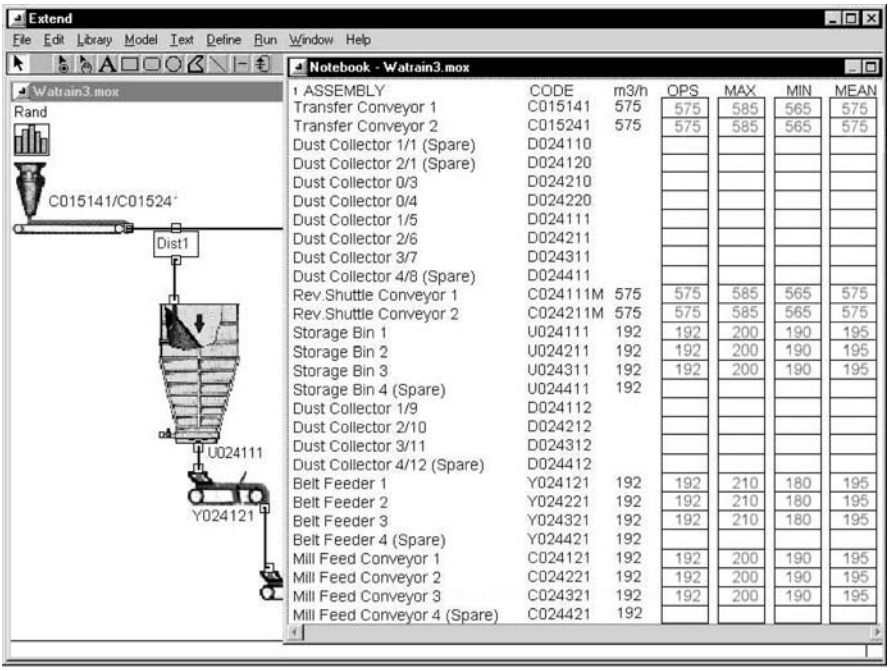


Fig. 4.54 Design details for simulation model sector 1: output performance results

Table 4.15 Comparative analysis of preliminary design data and simulation output data for simulation model sector 1

Assembly	Design flow vol.	Model flow vol.	Model min. flow vol.	Model max. flow vol.
Transfer conveyor 1	575	575	565	585
Transfer conveyor 2	575	575	565	585
Rev.shuttle conveyor 1	575	575	565	585
Rev.shuttle conveyor 2	575	575	565	585
Storage bin 1	192	195	180	210
Storage bin 2	192	195	180	210
Storage bin 3	192	195	180	210
Belt feeder 1	192	195	180	210
Belt feeder 2	192	195	180	210
Belt feeder 3	192	195	180	210
Mill feed conveyor 1	192	195	180	210
Mill feed conveyor 2	192	195	180	210
Mill feed conveyor 3	192	195	180	210

tolerance limit (LL) and an upper tolerance limit (UL), against which the minimum and maximum values of the simulation model's output data can be compared, need to be established to determine whether the range of variances of the model's output data falls within these tolerance limits.



ASSEMBLY	CODE	m3/h	OPS	MAX	MIN	MEAN
Transfer Conveyor 1	C015141	575	575	585	565	575
Transfer Conveyor 2	C015241	575	575	585	565	575
Dust Collector 1/1 (Spare)	D024110					
Dust Collector 2/1 (Spare)	D024120					
Dust Collector 0/3	D024210					
Dust Collector 0/4	D024220					
Dust Collector 1/5	D024111					
Dust Collector 2/6	D024211					
Dust Collector 3/7	D024311					
Dust Collector 4/8 (Spare)	D024411					
Rev.Shuttle Conveyor 1	C024111M	575	575	585	565	575
Rev.Shuttle Conveyor 2	C024211M	575	575	585	565	575
Storage Bin 1	U024111	192	192	200	190	195
Storage Bin 2	U024211	192	192	200	190	195
Storage Bin 3	U024311	192	192	200	190	195
Storage Bin 4 (Spare)	U024411	192				
Dust Collector 1/9	D024112					
Dust Collector 2/10	D024212					
Dust Collector 3/11	D024312					
Dust Collector 4/12 (Spare)	D024412					
Belt Feeder 1	Y024121	192	192	210	180	195
Belt Feeder 2	Y024221	192	192	210	180	195
Belt Feeder 3	Y024321	192	192	210	180	195
Belt Feeder 4 (Spare)	Y024421	192				
Mill Feed Conveyor 1	C024121	192	192	200	190	195
Mill Feed Conveyor 2	C024221	192	192	200	190	195
Mill Feed Conveyor 3	C024321	192	192	200	190	195
Mill Feed Conveyor 4 (Spare)	C024421	192				

Fig. 4.55 Simulation output for simulation model sector 1

Table 4.16 Acceptance criteria of simulation output data, with preliminary design data for simulation model sector 1

Assembly	Design min. vol. 2.5% tol.	Design max. vol. 2.5% tol.	Model min. vol.	Model max. vol.	Yes/no at 99%
Transfer conveyor 1	565	585	565	585	Yes
Transfer conveyor 2	565	585	565	585	Yes
Rev.shuttle conveyor 1	565	585	565	585	Yes
Rev.shuttle conveyor 2	565	585	565	585	Yes
Storage bin 1	187	197	180	210	No
Storage bin 2	187	197	180	210	No
Storage bin 3	187	197	180	210	No
Belt feeder 1	187	197	180	210	No
Belt feeder 2	187	197	180	210	No
Belt feeder 3	187	197	180	210	No
Mill feed conveyor 1	187	197	180	210	No
Mill feed conveyor 2	187	197	180	210	No
Mill feed conveyor 3	187	197	180	210	No

Validation of the simulation model's output data is thus not confined to a mere correlation of the mean values, whereby problems of autocorrelation can be significant, and the simulation model runs are not large enough to justify statistical spectral analysis of the output data (especially with very large, complex dynamic

simulation models), but the range or variance of the model's output data is compared to acceptable lower and upper confidence limits within a specified exact probability. The design specification is thus used as the mean, and the allowable design tolerance of $\pm 2.5\%$ of the mean is used as the square root of the variance, or standard deviation in the statistical t-distribution, to determine a confidence range or interval with lower tolerance limit (LL) and an upper tolerance limit (UL) at a 99% level of confidence for ten simulation runs. The minimum and maximum values of the simulation model's output data are then compared against this confidence range or interval. The last column of Table 4.16 indicates whether the model's output is acceptable in meeting the design criteria within a 99% level of confidence.

c) Evaluation of Simulation Model Sector 2

A major characteristic of the *process flow diagram (PFD) of sector 2* is that it depicts the conversion of solids to a solids and liquid slurry flow (through the action of the mills), and indicates how inputs are transformed into logical flow outputs that become modified inputs to the following assemblies (through the action of the

Table 4.17 Preliminary design data for simulation model sector 2

Assembly	Code	Flow vol.	Mass flow	Liq.	Solids
Rod mill 1	X024131	307	654	186	468
Rod mill 2	X024231	307	654	186	468
Rod mill 3	X024331	307	654	186	468
Rod mill 4	X024431	307	654	186	468
Mill discharge tank 1	T024141	1,119	2,102	943	1,159
Mill discharge tank 2	T024241	1,119	2,102	943	1,159
Mill discharge tank 3	T024341	1,119	2,102	943	1,159
Mill discharge tank 4	T024441				
Classifier feed pump 1/1	P024151	560	1,051	472	580
Classifier feed pump 1/2	P024152	560	1,051	472	580
Classifier feed pump 2/1	P024251	560	1,051	472	580
Classifier feed pump 2/2	P024252	560	1,051	472	580
Classifier feed pump 3/1	P024351	560	1,051	472	580
Classifier feed pump 3/2	P024352	560	1,051	472	580
Classifier feed pump (S)	P024451				
Classifier feed pump (S)	P024452				
Screen feed pot 1	V024161	1,152	2,142	982	1,159
Screen feed pot 2	V024261	1,152	2,142	982	1,159
Screen feed pot 3	V024361	1,152	2,142	982	1,159
Screen feed pot 4	V024461				
Ball mill 1	X024141	515	1,056	365	690
Ball mill 2	X024241	515	1,056	365	690
Ball mill 3	X024341	515	1,056	365	690
Ball mill 4	X024441	515	1,056	365	690

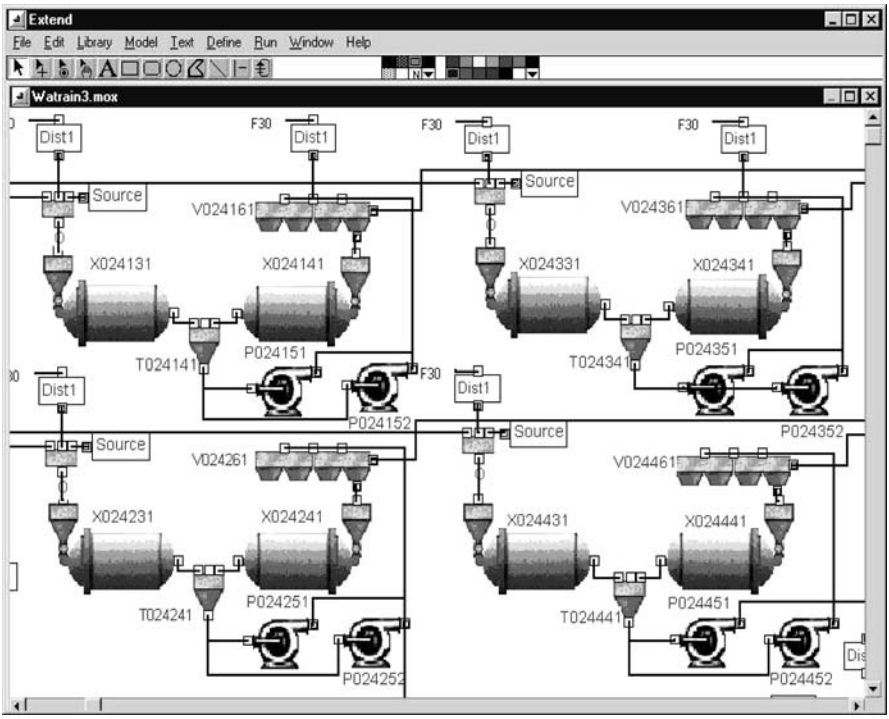


Fig. 4.56 Process flow diagram for simulation model sector 2

pumps), as depicted in the preliminary design data given in Table 4.17. The PFD is systematically examined to analyse deviations in process flow and system performance and, in this case, to determine solids to fluids mass-flow balance through the integrated assemblies.

Each assembly is graphically represented in the simulation model by a virtual prototype process equipment model (PEM). Each of the assemblies of the PFD depicted in Fig. 4.56 (i. e. the four mill feeder chutes, eight mills, eight pumps, four mixer chutes, four multi-bin feeders) is a process equipment model.

Process design specifications Each PEM contains model components that are configured in such a way that the *design specifications* of each system or assembly are met through the component's attributes. The model component's attributes for the mill input feeder chute and the mill output mixer chute connect three input values to a single output, and two input values to a single output respectively. The attributes for the multi-bin feeder convert the output by modifying the component's multiple inputs through a selection of statistical functions. The attributes for the mill pump also convert the pump's output by modifying the component's inputs through a selection of statistical functions representing typical pump delivery characteristics.

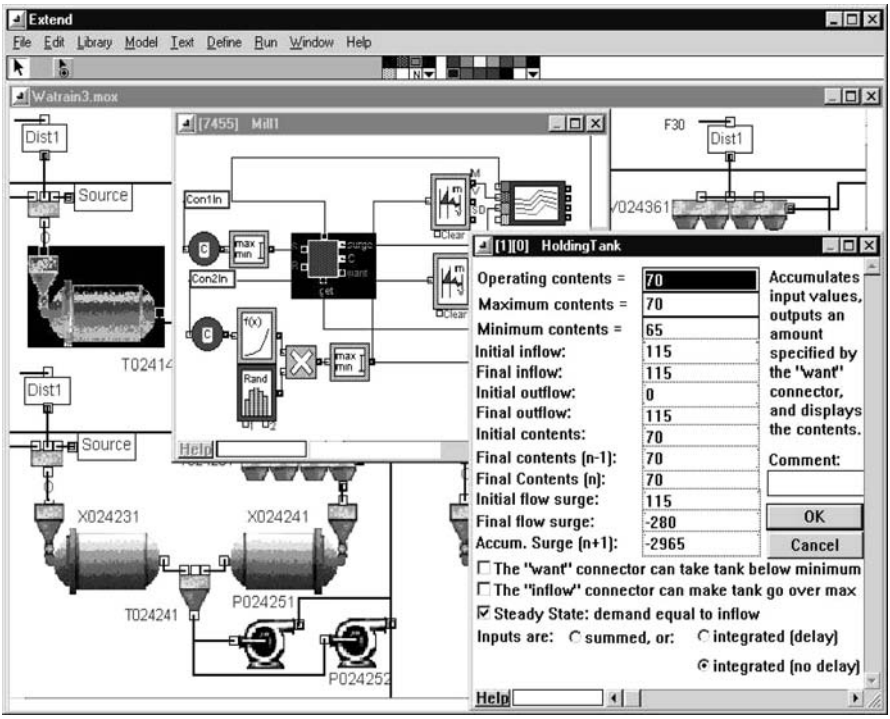


Fig. 4.57 Design details for simulation model sector 2: holding tank process design specifications

Figure 4.57 illustrates the model component's attributes of the rod mill, specifically the holding tank process characteristics such as operating contents, maximum and minimum contents, initial flow, final flow, initial contents, final contents, as well as initial and final flow surge.

Output performance results Performance variables relate to system or assembly contents, input and output flow quantities, as well as flow surges. The flow surge gives an indication of mass-flow balancing in the process. The output document is particular to each PEM and can be opened at any time, anywhere, in the dynamic systems simulation to determine the status of the process flow. The Extend© Performance Modelling program plotters can be placed anywhere in the modelled system configuration, and connected between any of the PEM input/output interface connectors, or within each PEM between model component connectors. The different process equipment models illustrated in Fig. 4.58 have plotters connected to the model components of each PEM's model configuration.

Figure 4.58 illustrates a typical output document showing performance results of the second-stage mill assembly. The plotted graph shows the trend of flow through the mill from start-up to steady state.

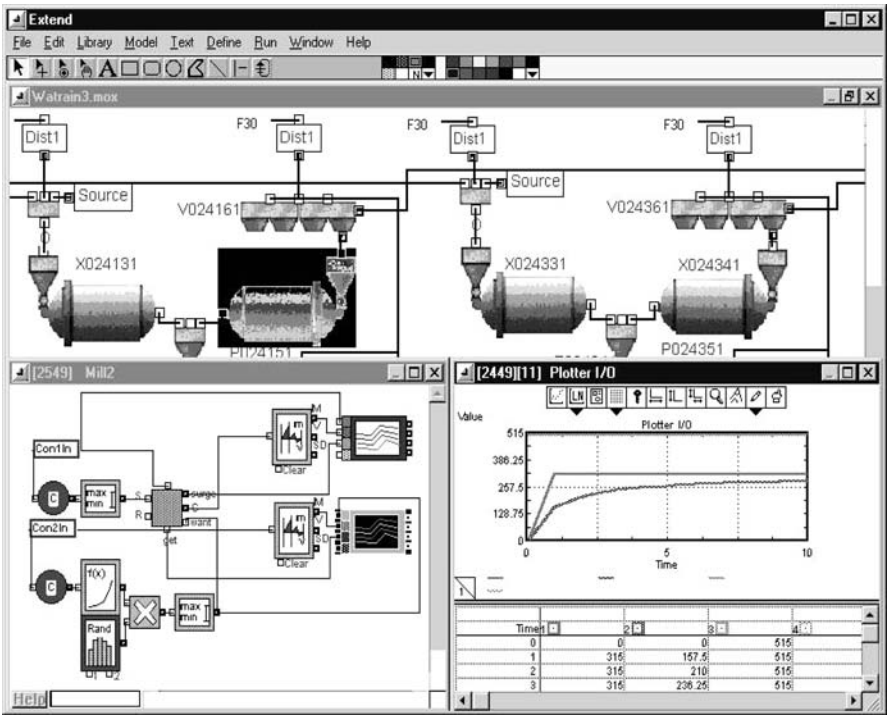


Fig. 4.58 Design details for simulation model sector 2: output performance results

d) Conclusion of Simulation Model Sector 2 Evaluation

Table 4.18 gives the values of a comparative analysis of preliminary design data and simulation output data for simulation model sector 2.

Figure 4.59 shows the simulation model's output for simulation model sector 2. As with simulation model sector 1, the range or variance of the model's output data is compared to acceptable lower and upper confidence limits within a specified exact probability. The design specification is again used as the mean, and the allowable design tolerance of $\pm 2.5\%$ of the mean is used as the standard deviation in the t-distribution, to determine a confidence range or interval with lower tolerance limit (LL) and an upper tolerance limit (UL) at a 99% level of confidence for ten simulation runs. The minimum and maximum values of the simulation model's output data are similarly compared against this confidence range or interval. The last column of Table 4.19 indicates whether the model's output is acceptable in meeting the design criteria within a 99% level of confidence. As can be seen, the mills and classifier feed pumps have a flow volume variance that is not acceptable within the 99% confidence interval as set by the design criteria, whereas the ball mills partially comply with the design criteria in that the simulated minimum flow is within the acceptable lower limit (LL).

Table 4.18 Comparative analysis of preliminary design data and simulation output data for simulation model sector 2

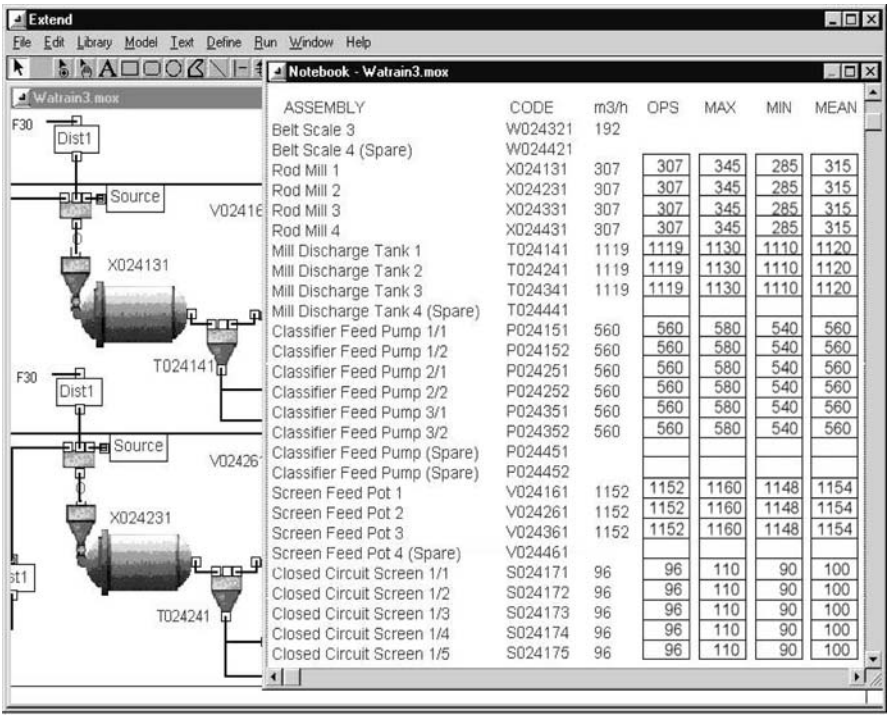
Assembly	Design flow vol.	Model flow vol.	Model min. flow vol.	Model max. flow vol.
Rod mill 1	307	315	285	345
Rod mill 2	307	315	285	345
Rod mill 3	307	315	285	345
Rod mill 4	307	315	285	345
Mill discharge tank 1	1,119	1,120	1,110	1,130
Mill discharge tank 2	1,119	1,120	1,110	1,130
Mill discharge tank 3	1,119	1,120	1,110	1,130
Mill discharge tank 4				
Classifier feed pump 1/1	560	560	540	580
Classifier feed pump 1/2	560	560	540	580
Classifier feed pump 2/1	560	560	540	580
Classifier feed pump 2/2	560	560	540	580
Classifier feed pump 3/1	560	560	540	580
Classifier feed pump 3/2	560	560	540	580
Classifier feed pump (S)				
Screen feed pot 1	1,152	1,154	1,148	1,160
Screen feed pot 2	1,152	1,154	1,148	1,160
Screen feed pot 3	1,152	1,154	1,148	1,160
Screen feed pot 4				
Ball mill 1	515	520	580	540
Ball mill 2	515	520	580	540
Ball mill 3	515	520	580	540
Ball mill 4	515	520	580	540

e) Evaluation of Simulation Model Sector 3

A major characteristic of the *process flow diagram (PFD) of sector 3* is that it depicts continuous fluid flow and indicates how inputs are transformed by each assembly into outputs that, in turn, become modified logical flow inputs to the next assembly, as depicted in the preliminary design data given in Table 4.20. The PFD is systematically examined to analyse deviations in process flow and system performance and, in this case, to determine mass fluid flow balance through integrated assemblies. Each assembly is graphically represented in the simulation model by a virtual prototype *process equipment model (PEM)*.

Each of the assemblies of the PFD depicted in Fig. 4.60, consisting of four processing tank systems containing 23 assemblies (four double tank feeder chutes, four processing tanks plus one standby, four sets of three-up parallel pumps, and one pump/condensate assembly), is a process equipment model.

A fluid mass-flow balance is the application of conservation of mass to the analysis of physical systems. By accounting for materials (solids or fluids) entering and leaving a system, mass flows can be identified from one system, or assembly, to the next. The exact mass-balance theory used in the analysis of the system depends on



ASSEMBLY	CODE	m3/h	OPS	MAX	MIN	MEAN
Belt Scale 3	W024321	192				
Belt Scale 4 (Spare)	W024421					
Rod Mill 1	X024131	307	307	345	285	315
Rod Mill 2	X024231	307	307	345	285	315
Rod Mill 3	X024331	307	307	345	285	315
Rod Mill 4	X024431	307	307	345	285	315
Mill Discharge Tank 1	T024141	1119	1119	1130	1110	1120
Mill Discharge Tank 2	T024241	1119	1119	1130	1110	1120
Mill Discharge Tank 3	T024341	1119	1119	1130	1110	1120
Mill Discharge Tank 4 (Spare)	T024441					
Classifier Feed Pump 1/1	P024151	560	560	580	540	560
Classifier Feed Pump 1/2	P024152	560	560	580	540	560
Classifier Feed Pump 2/1	P024251	560	560	580	540	560
Classifier Feed Pump 2/2	P024252	560	560	580	540	560
Classifier Feed Pump 3/1	P024351	560	560	580	540	560
Classifier Feed Pump 3/2	P024352	560	560	580	540	560
Classifier Feed Pump (Spare)	P024451					
Classifier Feed Pump (Spare)	P024452					
Screen Feed Pot 1	V024161	1152	1152	1160	1148	1154
Screen Feed Pot 2	V024261	1152	1152	1160	1148	1154
Screen Feed Pot 3	V024361	1152	1152	1160	1148	1154
Screen Feed Pot 4 (Spare)	V024461					
Closed Circuit Screen 1/1	S024171	96	96	110	90	100
Closed Circuit Screen 1/2	S024172	96	96	110	90	100
Closed Circuit Screen 1/3	S024173	96	96	110	90	100
Closed Circuit Screen 1/4	S024174	96	96	110	90	100
Closed Circuit Screen 1/5	S024175	96	96	110	90	100

Fig. 4.59 Simulation output for simulation model sector 2

the context of the design problem, specifically where the theory is used to analyse alternative processes.

Process design specifications Each PEM contains selected *model components* that are configured in such a way that the design specifications of each assembly are met through the component's attributes. The model component's attributes for the four double tank feeder chutes convert the chutes' output by modifying the component's inputs through a selection of statistical functions based on feed specifications. The model component's attributes for each of the processing tanks' pumps convert a pump's output by modifying the inputs through a selection of statistical functions representing the appropriate pump delivery characteristics.

Figure 4.61 illustrates the application of *Petri net (PN)*-based optimisation algorithms in dynamic systems simulation. The optimisation algorithm is a model component inherent to the processing tank PEM and determines process flow pressure surge through the tank.

Output performance results The fluid mass that enters a system must, by conservation of mass, either leave the system or accumulate within the system. Basically, the fluid mass-flow balance equation for a system without internal chemical reactions is: input $=$ output $+$ accumulation. In the absence of a chemical reaction,

Table 4.19 Acceptance criteria of simulation output data, with preliminary design data for simulation model sector 2

Assembly	Design min. vol. 2.5% tol.	Design max. vol. 2.5% tol.	Model min. vol.	Model max. vol.	Yes/no at 99%
Rod mill 1	300	315	285	345	No
Rod mill 2	300	315	285	345	No
Rod mill 3	300	315	285	345	No
Rod mill 4	300	315	285	345	No
Mill discharge tank 1	1,080	1,148	1,110	1,130	Yes
Mill discharge tank 2	1,080	1,148	1,110	1,130	Yes
Mill discharge tank 3	1,080	1,148	1,110	1,130	Yes
Mill discharge tank 4					
Classifier feed pump 1/1	545	575	540	580	No
Classifier feed pump 1/2	545	575	540	580	No
Classifier feed pump 2/1	545	575	540	580	No
Classifier feed pump 2/2	545	575	540	580	No
Classifier feed pump 3/1	545	575	540	580	No
Classifier feed pump 3/2	545	575	540	580	No
Classifier feed pump (S)					
Screen feed pot 1	1,122	1,182	1,148	1,160	Yes
Screen feed pot 2	1,122	1,182	1,148	1,160	Yes
Screen feed pot 3	1,122	1,182	1,148	1,160	Yes
Screen feed pot 4					
Ball mill 1	502	528	510	530	Part
Ball mill 2	502	528	510	530	Part
Ball mill 3	502	528	510	530	Part
Ball mill 4	502	528	510	530	Part

the logical fluid flow in and out of a system or assembly will be the same. To perform a balance, the boundaries of the system must be well defined. Fluid mass-flow balances can be taken over physical systems at multiple scales, taking into consideration flow surges, and can be simplified with the assumption of steady state, where the accumulation term is zero.

Figure 4.62 illustrates a typical output document showing performance results of the processing tank PEM. These performance variables relate to assembly contents, input and output flow quantities, as well as flow surges. The flow surge gives an indication of deviations from steady-state flow. The plotted graph shows the trend of flow from start-up to steady state.

f) Conclusion of Simulation Model Sector 3 Evaluation

Table 4.21 gives the values of a comparative analysis of preliminary design data and simulation output data for simulation model sector 3.

Figure 4.63 shows the simulation model's output for simulation model sector 3. As with simulation model sectors 1 and 2, the range or variance of the model's

Table 4.20 Preliminary design data for simulation model sector 3

Assembly	Code	Flow vol.	Mass flow	Liq.	Solids
Desilicator 1	T026021	1,250	2,136	1,213	937
Desilicator 2	T026031	968	1,642	928	721
Desilicator 3	T026041	968	1,642	928	721
Desilicator 4	T026051	1,250	2,136	1,213	937
Desilicator 5	T026061	968	1,642	928	721
Slurry splitter box 1	L026031	1,250	2,136	1,197	938
Slurry splitter box 2	L026041	968	1,642	928	721
Slurry splitter box 3	L026051	968	1,642	928	721
Slurry splitter box 4	L026061	1,250	2,136	1,197	938
Slurry forwarding pump 1	P026011	1,250	2,136	1,197	938
Slurry forwarding pump 2	P026021	968	1,642	928	721
Slurry forwarding pump 3	P026031	968	1,642	928	721
Slurry forwarding pump 4	P026041	968	1,642	928	721
Slurry forwarding pump 5	P026051	968	1,642	928	721
Slurry forwarding pump 6	P026061	1,250	2,136	1,197	938
Discharge pump 1	P026071	323	547	312	235
Discharge pump 2	P026171	323	547	312	235
Discharge pump 3	P026271	323	547	312	235
Discharge pump 4	P026301	323	547	312	235
Discharge pump 5	P026302	323	547	312	235
Discharge pump 6	P026303	323	547	312	235

Table 4.21 Comparative analysis of preliminary design data and simulation output data for simulation model sector 3

Assembly	Design flow vol.	Model flow vol.	Model min. flow vol.	Model max. flow vol.
Desilicator 1	1,250	1,255	1,245	1,265
Desilicator 2	968	967.5	960	975
Desilicator 3	968	967.5	960	975
Desilicator 4	1,250	1,255	1,245	1,265
Desilicator 5	968	967.5	960	975
Slurry splitter box 1	1,250	1,250	1,245	1,255
Slurry splitter box 2	968	967.5	960	975
Slurry splitter box 3	968	967.5	960	975
Slurry splitter box 4	1,250	1,250	1,245	1,255
Slurry forwarding pump 1	1,250	1,250	1,240	1,260
Slurry forwarding pump 2	968	967.5	955	980
Slurry forwarding pump 3	968	967.5	955	980
Slurry forwarding pump 4	968	967.5	955	980
Slurry forwarding pump 5	968	967.5	955	980
Slurry forwarding pump 6	1,250	1,250	1,240	1,260
Discharge pump 1	323	325	320	330
Discharge pump 2	323	325	320	330
Discharge pump 3	323	325	320	330
Discharge pump 4	323	325	320	330
Discharge pump 5	323	325	320	330
Discharge pump 6	323	325	320	330

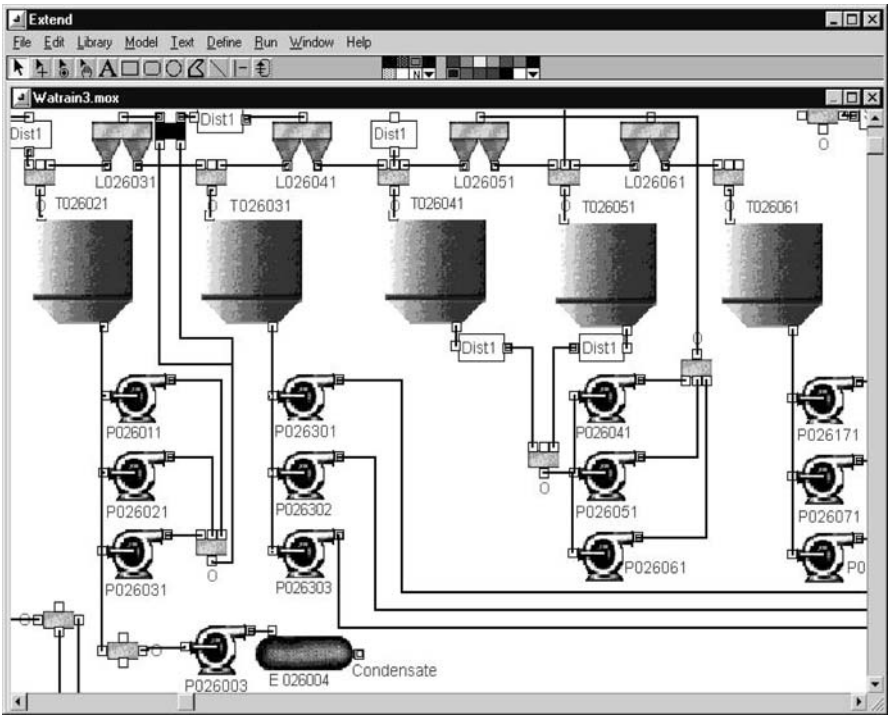


Fig. 4.60 Process flow diagram for simulation model sector 3

output data is compared to acceptable lower and upper confidence limits within a specified exact probability. The design specification is again used as the mean, and the allowable design tolerance of $\pm 2.5\%$ of the mean is used as the square root of the variance, namely the standard deviation, in the t-distribution, to determine a confidence range or interval with lower tolerance limit (LL) and an upper tolerance limit (UL) at a 99% level of confidence for ten simulation runs. The minimum and maximum values of the simulation model's output data are similarly compared against this confidence range or interval. The last column of Table 4.22 indicates whether the model's output is acceptable in meeting the design criteria within a 99% level of confidence. As can be seen, all the assemblies have a flow volume variance that is acceptable within the 99% confidence interval as set by the design criteria.

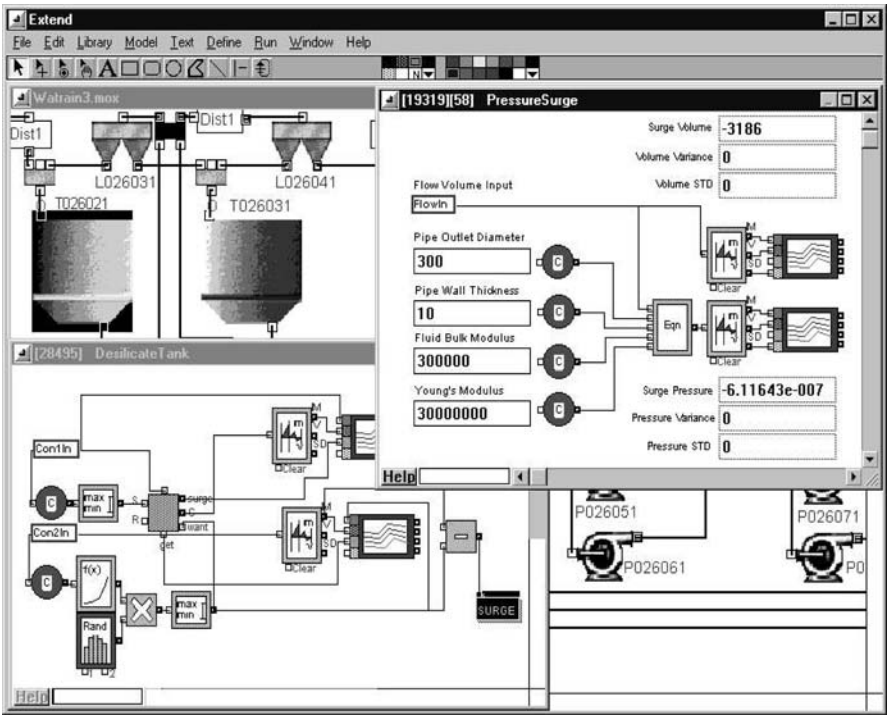


Fig. 4.61 Design details for simulation model sector 3: process design specifications

4.4.3 Application Modelling Outcome

Verification of the process simulation model with the PEM blocks included the *specification of model components* as well as the *formulation of functional relationships*, all of which are inherent in the *dynamic systems simulation blackboard model* that is used to control the design knowledge sources and integrate the knowledge-based design applications. In contrast to model verification, the *validity* of the simulation model depended on the ability of the model to predict the *results* of the model's behaviour. However, validation of the simulation model was *not* based on a correlation of the mean values of the model's output data and the specified design flow volumes for each PEM, due to possible problems of autocorrelation and the limited number of simulation model runs not being large enough to justify statistical spectral analysis of the output data. Rather, statistical inference was applied to determine whether the range of the model's output data fell between acceptable lower and upper confidence limits within a specified exact probability.

In order to determine a confidence range or interval with a lower tolerance limit (LL) and an upper tolerance limit (UL), the specified design flow volume was used as the mean, and the allowable design tolerance of $\pm 2.5\%$ of the mean was used as

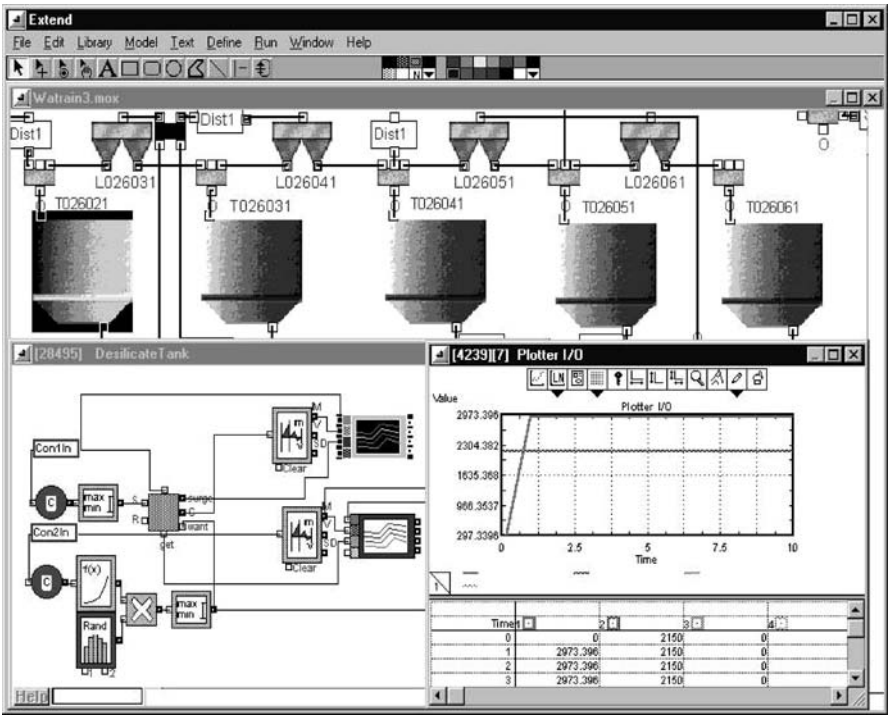
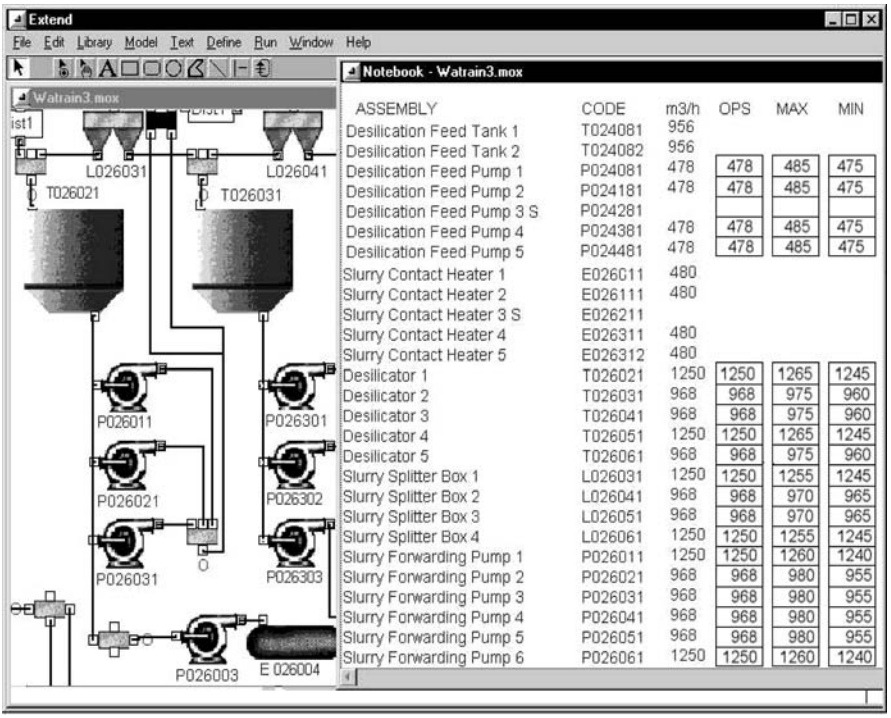


Fig. 4.62 Design details for simulation model sector 3: output performance results

standard deviation in the statistical t-distribution at a 99% level of confidence for ten simulation runs. The minimum and maximum values of the simulation model's output data were then compared against this confidence range or interval to determine whether the model's output was acceptable in meeting the design criteria.

As indicated in Tables 4.16, 4.19 and 4.22, not all of the assemblies listed met the required design criteria, indicating that the simulation model failed at a 99% level of confidence specifically for those assemblies. However, the statistical approach of determining confidence intervals with the t-distribution was repeated for 95% and 90% levels of confidence. Close on 85% of the simulation model's output data was found to meet the required design criteria at a 95% level of confidence, and *all* of the simulation model's output data met the required design criteria at a 95% level of confidence. This implies that the process simulation model with the PEM blocks is capable of predicting process output within a 10% margin of error for each PEM. Due to the fact that the model simulates a complex integrated continuous process flow, a 90% level of confidence is acceptable for the preliminary design phase of the engineered installation.



ASSEMBLY	CODE	m3/h	OPS	MAX	MIN
Desilication Feed Tank 1	T024081	956			
Desilication Feed Tank 2	T024082	956			
Desilication Feed Pump 1	P024081	478	478	485	475
Desilication Feed Pump 2	P024181	478	478	485	475
Desilication Feed Pump 3 S	P024281				
Desilication Feed Pump 4	P024381	478	478	485	475
Desilication Feed Pump 5	P024481	478	478	485	475
Slurry Contact Heater 1	E026011	480			
Slurry Contact Heater 2	E026111	480			
Slurry Contact Heater 3 S	E026211				
Slurry Contact Heater 4	E026311	480			
Slurry Contact Heater 5	E026312	480			
Desilicator 1	T026021	1250	1250	1265	1245
Desilicator 2	T026031	968	968	975	960
Desilicator 3	T026041	968	968	975	960
Desilicator 4	T026051	1250	1250	1265	1245
Desilicator 5	T026061	968	968	975	960
Slurry Splitter Box 1	L026031	1250	1250	1255	1245
Slurry Splitter Box 2	L026041	968	968	970	965
Slurry Splitter Box 3	L026051	968	968	970	965
Slurry Splitter Box 4	L026061	1250	1250	1255	1245
Slurry Forwarding Pump 1	P026011	1250	1250	1260	1240
Slurry Forwarding Pump 2	P026021	968	968	980	955
Slurry Forwarding Pump 3	P026031	968	968	980	955
Slurry Forwarding Pump 4	P026041	968	968	980	955
Slurry Forwarding Pump 5	P026051	968	968	980	955
Slurry Forwarding Pump 6	P026061	1250	1250	1260	1240

Fig. 4.63 Simulation output for simulation model sector 3

4.5 Review Exercises and References

Review Exercises

1. Discuss cost modelling for design availability and maintainability.
2. Explain economic loss and the cost of dependency.
3. Give a brief account of life-cycle analysis and life-cycle costs.
4. Consider life-cycle cost elements in engineering design.
5. Describe present value calculations for life-cycle costs.
6. Discuss trade-off measurement for life-cycle costs.
7. Give a brief account of availability modelling based on system performance, considering process capability, process characteristics and functional effectiveness.
8. Explain the concept of sizing maximum or design capacity.
9. Define inherent availability (A_i)
10. Discuss inherent availability modelling with uncertainty.
11. Discuss the significance of the application of the exponential function for determining inherent availability.
12. Describe confidence determination of inherent availability predictions.

Table 4.22 Acceptance criteria of simulation output data, with preliminary design data for simulation model sector 3

Assembly	Design min. vol. 2.5% tol.	Design max. vol. 2.5% tol.	Model min. vol.	Model max. vol.	Yes/no at 99%
Desilicator 1	1,220	1,280	1,245	1,265	Yes
Desilicator 2	943	993	960	975	Yes
Desilicator 3	943	993	960	975	Yes
Desilicator 4	1,220	1,280	1,245	1,265	Yes
Desilicator 5	943	993	960	975	Yes
Slurry splitter box 1	1,220	1,280	1,245	1,255	Yes
Slurry splitter box 2	943	993	960	975	Yes
Slurry splitter box 3	943	993	960	975	Yes
Slurry splitter box 4	1,220	1,280	1,245	1,255	Yes
Slurry forward pump 1	1,220	1,280	1,240	1,260	Yes
Slurry forward pump 2	943	993	955	980	Yes
Slurry forward pump 3	943	993	955	980	Yes
Slurry forward pump 4	943	993	955	980	Yes
Slurry forward pump 5	943	993	955	980	Yes
Slurry forward pump 6	1,220	1,280	1,240	1,260	Yes
Discharge pump 1	315	330	320	330	Yes
Discharge pump 2	315	330	320	330	Yes
Discharge pump 3	315	330	320	330	Yes
Discharge pump 4	315	330	320	330	Yes
Discharge pump 5	315	330	320	330	Yes
Discharge pump 6	315	330	320	330	Yes

13. Discuss preliminary maintainability modelling.
14. Give a brief account of Markov modelling for design availability and maintainability with regard to the two-state Markov model, and the multi-state Markov model.
15. Define Markov model supplementary variables.
16. Define achieved availability.
17. Discuss achieved availability modelling subject to maintenance.
18. Consider maintainability assessment with maintenance modelling.
19. Discuss the impact of maintenance assessment on systems design.
20. Describe maintainability measures and maintenance assessment.
21. Discuss maintenance strategies and cost optimisation modelling.
22. Give a brief account of the basic principles of maintenance.
23. Describe a model of preventive maintenance physical checks.
24. Describe a model of preventive maintenance replacement shuts.
25. Define maintenance strategy.
26. Explain the concepts of reliability, availability and maintainability in maintenance strategy and discuss their differences.
27. Give a brief account of the three principles of a maintenance strategy.
28. Discuss establishing maintenance strategies for engineering design.
29. Describe maintenance cost optimisation modelling.
30. Define dependability modelling.

31. Discuss the significance of dependability modelling for design availability and maintainability.
32. Define operational availability (A_o).
33. Discuss operational availability modelling with logistic support.
34. Consider a general approach for evaluating operational availability.
35. Give a brief account of system availability evaluation considerations.
36. Discuss maintainability evaluation and built-in or non-destructive testing (BIT).
37. Describe maintainability evaluation indices.
38. Give a brief account of diagnostic systems and built-in testing.
39. Explain basic system and BIT concurrent design and evaluation.
40. Discuss the evaluation of BIT systems.
41. Consider application modelling of availability and maintainability in engineering design.
42. Define equivalent availability (EA).
43. Discuss and compare the equivalent maintainability measures of downtime and outage.
44. Describe outage measurement with the ratio of ER over EM.
45. Discuss system performance measures and limits of capability.
46. Describe performance parameters for system integrity and their significance in engineering design.
47. Discuss analysis of the parameter profile matrix.
48. Discuss the significance of the design checklist.
49. Explain integrity prediction of common items of equipment.
50. Give a brief account of a design review of performance parameters for system integrity.
51. Discuss the significance of reliability and maintainability checklists.
52. Describe system performance analysis and simulation modelling in engineering design.
53. Consider different types of system performance models.
54. Briefly describe the significance and contribution of system simulation modelling in engineering design.
55. Discuss uncertainty in system performance simulation modelling.
56. Explain propagation of the effect of uncertainties.
57. Describe the extreme condition approach for uncertainty analysis.
58. Describe the statistical approach for uncertainty analysis.
59. Give an explanation for mitigating the effect of uncertainty.
60. Describe maximising design availability using Petri net models.
61. Discuss Petri net theory and its application in engineering design.
62. Define the basic Petri net model and compare it to the definitions of stochastic Petri nets as well as Markovian stochastic Petri nets.
63. Briefly explain the process of generating reachability graphs.
64. Discuss the measures of Markovian stochastic Petri nets.
65. Define stochastic reward nets and non-Markovian stochastic Petri nets.
66. Consider designing for availability using Petri net modelling.
67. Describe numerical computations for the availability Petri net model.

68. Consider a steady-state solution to the availability Petri net model.
69. Explain complex systems theory.
70. Discuss systems engineering and complex systems theory.
71. Consider the application and significance of systems engineering in engineering design.
72. Briefly discuss complexity in engineering design and its significance in systems engineering.
73. Give a brief account of the functions of systems engineering analysis.
74. Describe reliability block diagrams (RBDs) and availability block diagrams (ABDs), and indicate their fundamental differences.
75. Consider effectiveness measures in systems engineering and their significance in engineering design.
76. Give a brief account of evaluating complexity in engineering design.
77. Define complexity in systems design.
78. Describe various system state definitions and evaluating complexity of the different state definitions.
79. Define complicatedness in systems design.
80. Describe complexity in systems and complicatedness as a function of complexity in designing for complex but uncomplicated systems.

References

Ajmone Marsan M, Balbo G, Conte G, Donatelli S, Franceschinis G (1995) Modelling with generalised stochastic Petri nets. Wiley, New York

Alfredsson P, Wååk O (1999) Constant vs. non-constant failure rates: some misconceptions with respect to practical applications. Systecon AB, Stockholm

Ayres RU (1988) Complexity, reliability, and design: manufacturing implications. Manufacturing Rev 1(1):26–35

Barringer PH (1998) Life cycle cost and good practices. In: NPRA Maintenance Conf, May, San Antonio, TX

Barringer PH, Weber DP (1996) Life cycle cost tutorial. Fifth Int Conf Process Plant Reliability, Gulf, Houston, TX

Batill SM, Renaud JE, Xiaoyu Gu, (2000) Modeling and simulation uncertainty in multidisciplinary design optimization. In: AIAA-2000-4803, 8th AIAA/NASA/USAF/ISSMO Symp Multidisciplinary Analysis and Optimization, American Institute of Aeronautics and Astronautics, California, September, pp 5–8

Bing G (1996) Due diligence techniques and analysis: critical questions for business decisions. Quorum Books, Westport, CT

Blanchard BS, Verma D, Peterson EL (1995) Maintainability: a key to effective serviceability and maintenance management. Prentice Hall, Englewood Cliffs, NJ

Bobbio A, Telek M (1997) Non-exponential stochastic Petri nets: an overview of methods and techniques. Computer Systems Sci Eng

Booker JM, Bement TR, Meyer MA, Kerscher WJ (2000) PREDICT: a new approach to product development and lifetime assessment using information integration technology. Los Alamos National Laboratory Rep LA-UR-00-4737

Boullart L (1988) Artificial intelligence and expert systems: next generation tools. In: Boullart L, Van Ravenzwaaij E, Jansen JP (eds) Industrial process control systems: reliability availability and maintainability. Proc IFAC Worksh, Bruges, Belgium, pp 45–52

Box GEP, Hunter WG, Hunter JS (1978) Statistics for experiments. Wiley, New York
Bulgren WG (1982) Discrete system simulation. Prentice Hall, Englewood Cliffs, NJ
Bussey LE (1978) The economic analysis of industrial projects. International Series in Industrial and Systems Engineering, Prentice Hall, Englewood Cliffs, NJ
Carter CL (1978) The control and assurance of quality, reliability and safety. C.L. Carter, Richardson, TX
Casti J (1979) Connectivity, complexity, and catastrophe in large-scale systems. International Series on Applied Systems Analysis, Wiley, New York
Casti J (1994) Complexification. Harper Collins, New York
Chen R, Ward AC (1995) The RANGE family of propagation operations for intervals on simultaneous linear equations. Artificial Intelligence Eng Design Anal Manufacturing 9(3):183–196
Cheremisinoff NP (1984) Fluid flow. Gulf, Houston, TX
Choi H, Kulkarni VG, Trivedi K (1994) Markov regenerative stochastic Petri nets. Performance Evaluation 20:337–357
Ciardo G, Muppala J, Trivedi KS (1991) On the solution of GSPN reward models. Performance Evaluation 12:237–253
Ciardo G, German R, Lindemann C (1994) A characterization of the stochastic process underlying a stochastic Petri Net. IEEE Trans Software Eng 20:506–515
Conlon JC, Lilius WA (1982) Test and evaluation of system reliability, availability and maintainability. Office of the Under Secretary of Defense for Research and Engineering, USA Department of Defense, DoD 3235.1-H
Corkill DD, Gallagher KQ, Johnson PM (1987) Achieving flexibility, efficiency, and generality in blackboard architectures. Department of Computer and Information Science, University of Massachusetts, Amherst, MA
Deshmukh AV (1993) Complexity and chaos in manufacturing systems. PhD Thesis, School of Industrial Engineering, Purdue University, West Lafayette, IN
Dhillon BS (1983) Reliability engineering in systems design and operation. Van Nostrand Reinhold, Berkshire
Dhillon BS (1999b) Engineering maintainability. Gulf, Houston, TX
Diamond B (1995) Performance modelling for decision support. Imagine That, San Jose, CA
Diamond R (1997) Extend: performance modelling for decision support. Imagine That, San Jose, CA
DoD 3235.1-H. (1982) Test and evaluation of system reliability, availability and maintainability. Office of the Under Secretary of Defense for Research and Engineering, USA, DoD 3235.1-H
DoD 5000.2-R. (1997) Reliability, availability and maintainability (RAM). USA Department of Defense, Office of the Under secretary of Defense for Research and Engineering, Rep DoD 5000.2-R, March
Drenick RF (1960) The failure law of complex equipment. J Soc Industrial Appl Math 8:680–690
Du X, Chen W (1999a) Towards a better understanding of modeling feasibility robustness in engineering design. ASME Design Technical Conf, Pap no DAC-8565, Las Vegas, NV
Du X, Chen W (1999b) A methodology for managing the effect of uncertainty in simulation-based design. Sem Pap, 1999, University of Illinois at Chicago, Chicago, IL
Du X, Chen W, Garimella R (1999c) Propagation and management of uncertainties in simulation-based collaborative systems design. University of Illinois at Chicago, Chicago, IL
Elsayed EA (1996) 'Reliability engineering'. Addison-Wesley Longman, Reading, MA
Emshoff JR, Sisson RL (1970) Design and use of computer simulation models. Macmillan, New York
Extend (2001) Extend performance modelling for decision support. Imagine That, San Jose, CA
Fabrycky WJ, Blanchard BS (1991) Life-cycle cost and economic analysis. Prentice Hall, Englewood Cliffs, NJ
Garey M, Johnson D (1979) Computers and intractability: a guide to the theory of NP-completeness. W.H. Freeman, New York
German R, Lindemann C (1994) Analysis of stochastic Petri nets by the method of supplementary variables. Performance Evaluation J 20:317–335

Goldratt EM (1990) What is this thing called the theory of constraints? North River Press, Croton-on-Hudson, NY
Gunter BH (1989a) The use and abuse of C pk. Quality Progress, January, pp 72–73
Gunter BH (1989b) The use and abuse of C pk, part 2. Quality Progress, March, pp 108–109
Gunter BH (1989c) The use and abuse of C pk, part 3. Quality Progress, May, pp 79–80
Gunter BH (1989d) The use and abuse of C pk, part 4. Quality Progress, July, pp 86–87
Hicks CR (1993) Fundamental concepts in the design of experiments. Oxford University Press, Oxford
Hill PH (1970) The science of engineering design. Holt, Rinehart and Winston, New York
Hillestad RJ (1982) Multi-echelon technique for recoverable item control. Rand Corporation Project Air Force Rep R-2785-AF, Santa Monica, CA
Hoover SV, Perry RF (1989) Simulation: a problem-solving approach. Addison-Wesley, Reading, MA
Huggett PJ, Edmundson JB (1986) Machinery damage control. Edmundson Huggett, New Doornfontein, Johannesburg
Huzdovich JM (1981) Power plant availability engineering—methods of analysis, program planning, and applications. Electricity Power Research Institute Final Rep EPRI NP-2168 Nuclear Power Division
ICS (2002) The dynamic systems simulation blackboard model. ICS Industrial Consulting Services, Miami, Gold Coast City, Queensland
INCOSE (2002) Systems engineering. International Council on Systems Engineering, Seattle, WA, Wiley, New York
Jardine AKS (1973) Maintenance, replacement and reliability. Wiley, New York
Kececioglu D (1995) Maintainability, availability, and operational readiness engineering. Prentice Hall, Englewood Cliffs, NJ
Lam C, Yeh R (1994) Optimal maintenance policies for deteriorating systems under various maintenance strategies. IEEE Trans Reliability 43
Laviolette M, Seaman J Jr, Barrett J, Woodall W (1995) A probabilistic and statistical view of fuzzy methods. Technometrics J 37:249–281
Law AM, Kelton WD (1991) Simulation modelling and analysis, 2nd edn. McGraw-Hill, New York
Lee DE, Melkanoff ME (1993) Issues in product life cycle analysis. In: ASME Design Automation Conf, Advances in Design Automation, Albuquerque, NM, ASME Press, New York, pp 75–86
Lindemann C, Thummler A (1999) Transient analysis of deterministic and stochastic Petri nets with concurrent deterministic transitions. Elsevier, Amsterdam, Performance Evaluation 36/37:35–54
Little JDC (1961) A proof for the queuing formula: L=lW. Operations Res 9:383–387
McGuire JG, Kuokka DR, Weber JC, Tenenbaum JM, Gruber TR, Olsen GR (1993) SHADE: technology for knowledge-based collaborative engineering. Concurrent Eng Res Appl 1(3)
McKinney M, Thompson G (1989) A survey of process plant maintainability problems. Proc Inst Mech Engrs Part F J Process Mech Eng 203(El):29–35
Mead C (1994) Preface to Workshop report on New Paradigms for Manufacturing. In: Mukherjee A, Hilibrand J (eds) National Science Foundation Rep NSF 94-123, Arlington, VA, pp 1–2
MIL-HDBK-470A (1997) Designing and developing maintainable products and systems. Department of Defense, Washington, DC
MIL-HDBK-471A (1996) Maintainability demonstration. Department of Defense, Washington, DC
MIL-HDBK-472 (1996) Maintainability prediction. Department of Defense, Washington, DC
MIL-STD-470 (1996) Maintainability Improvement Program. DoD, Washington, DC
MIL-STD-470A (1996) Maintainability Program for Systems and Equipment. DoD, Washington, DC
MIL-STD-471A (1996) Maintainability verification/demonstration/evaluation. Department of Defense, Washington, DC

MIL-STD-1472D (1996) Human engineering design criteria for military systems, equipment and facilities. DoD, Washington, DC
MIL-STD-46855B (1996) Human engineering requirements for military systems, equipment and facilities. DoD, Washington, DC
Molloy MK (1982) Performance analysis using stochastic Petri nets. IEEE Trans Computers C31:913–917
Montgomery DC (1991) Introduction to statistical quality control, 2nd edn. Wiley, New York
Murata T (1989) Petri nets: properties, analysis and applications. Proc IEEE 77:541–580
Naylor TH, Balintfy JL, Burdick DS, Chu K (1966) Computer simulation techniques. Wiley, New York
Nelson ME (1981) Handbook of availability improvement methodology. Trident Engineering Associates, Annapolis, MD, US Department of Energy, Economic Regulatory Administration, Division of Power Supply and Reliability
Neuts MF (1981) Matrix geometric solutions in stochastic models. Johns Hopkins University Press, Baltimore, MD
Olsen GR, Cutkosky MR, Tenenbaum JM, Gruber TR (1995) Collaborative engineering based on knowledge sharing agreements. Concurrent Eng Res Appl 3(2):145–159
Orlicky JA, Plossi GW, Wight OW (1970) Material requirements planning systems. 13th Int APICS Conf, Cincinnati, OH
Pancerella C, Hazelton A and Frost HR (1995) 'An autonomous agent for on-machine acceptance of machined components', Proceedings of Modeling, Simulation, and Control Technologies for Manufacturing, SPIE's International Symposium on Intelligent Systems and Advanced Manufacturing.
Parkinson A, Sorensen C and Pourhassan N (1993) 'A General Approach for Robust Optimal Design', Trans. of the ASME, Vol. 115, pp. 74–80
Patton JD (1980) Maintainability and maintenance management. Instrument Society of America, Research Triangle Park, NC
Pecht M (1995) Product reliability, maintainability, and supportability handbook. CRC Press, New York
Peterson JL (1981) Petri net theory and the modeling of systems. Prentice Hall, Englewood Cliffs, NJ
Phadke MS (1989) Quality engineering using robust design. Prentice Hall, Englewood Cliffs, NJ
Pritsker AB (1990) Papers, experiences, perspectives. Systems Publishing, New York
Shannon RE (1975) Systems simulation: the art and science. Prentice Hall, Englewood Cliffs, NJ
Simon HA (1981) The sciences of the artificial. MIT Press, Cambridge, MA
Smith DJ (1981) Reliability and maintainability in perspective. Macmillan Press, London
Steiner S, Bovas A, MacKay J (1995) Understanding process capability indices. Institute for Improvement of Quality and Productivity, Department of Statistics and Actuarial Science, University of Waterloo, Waterloo, Ontario
Suh NP (1999) A theory of complexity, periodicity, and the design axioms. Res Eng Design 11:116–131
Suri R, Otto K (1999) System-level robustness through integrated modeling. ASME Design Technical Conf, Pap no DETC99/DFM-8966, Las Vegas, NV
Taguchi G (1993) Taguchi on robust technology development: bringing quality engineering upstream. ASME Press, New York
Taguchi G, Elsayed E, Hsiang T (1989) Quality engineering in production systems. McGraw-Hill, New York
Tang V, Salminen V (2001) Towards a theory of complicatedness: framework for complex systems analysis and design. 13 Int Conf Engineering Design, Glasgow, Scotland, UK
Thompson G, Geominne J, Williams JR (1998) A method of plant design evaluation featuring maintainability and reliability. Proc Inst Mech Engrs vol 212 Part E
Vajda S (1974) Maintenance replacement and reliability. Topics in Operational Research, University of Birmingham

Virtanen I (1975) Application of supplementary variables and Laplace transforms to operational behaviour and reliability of a complex system. Proc Turku School of Economics and Business Administration, Series A II(1):385–399

Virtanen I (1977) On the concepts and derivation of reliability in stochastic systems with states of reduced efficiency. Doctoral Thesis Publ no 10, Institute for Applied Mathematics, University of Turku, Turku

Warfield JN (2000) A structure-based science of complexity: transforming complexity into understanding. Kluwer, Dordrecht

Wolfram S (1988) Emerging syntheses in science. In: Proc Founding Workshops of Santa Fe Institute, Addison-Wesley, Reading, MA, pp 183–189

Zadeh LA (1995) Probability theory and fuzzy logic are complementary rather than competitive. Technometrics, August, vol 37, no 3, pp 271–276

Zakarian A, Kusiak A (1997) Modeling manufacturing dependability. IEEE Trans Robotics Automation 13(2)

Chapter 5
Safety and Risk in Engineering Design

Abstract In this chapter, the introduction of new or modified systems into an engineering process is considered, whereby safety with respect to risk and loss through accidents or incidents resulting from the complex integration of systems is predicted, assessed and evaluated, to ensure that the design will have as minimum a risk as is reasonably practicable. Risk relates to a combination of the likelihood of occurring hazards, and to the severity of their outcome or consequence. Safety in engineering design begins with identifying possible hazards that could occur, as well as the corresponding system states that could lead to an accident or incident in the designed system. This is determined through *hazards analysis*. The initial hazards analysis should begin at the earliest concept formation stages of systems design, and the information should be used to guide the emerging design with respect to safety requirements throughout the engineering design process. Safety in engineering design normally includes a *causal analysis*, which involves identifying various cause-effect sequences of hazardous events that may combine to cause the identified hazards. Thereafter, a *consequence analysis* identifies the sequences of events that could lead from a hazard to an accident or incident. Working through these phases of hazards and safety analysis, and iterating where appropriate, a *safety case* is prepared that relates to the assurance that the system is relatively safe. Hazards and safety analyses provide a comprehensive methodology for designing for safety. Designing for safety includes risk reduction measures and involves conducting risk mitigation strategies to, first, reduce the likelihood that a hazard could result in an accident or incident and, second, to aim at reducing the severity of the likely event. Because designing for safety strives for a significant level of confidence in the results of these strategies, and the need for an objective systems scrutiny from a safety viewpoint, it typically involves systematic safety analysis with independent safety prediction, safety assessment, and safety evaluation during the schematic, preliminary and detail design phases respectively of the overall engineering design process.

R.F. Stapelberg, *Handbook of Reliability, Availability, Maintainability and Safety in Engineering Design*, © Springer 2009

5.1 Introduction

The previous two chapters dealt with an analysis of engineering design that considered prediction, assessment and evaluation of systems reliability and functional performance, and of systems availability and maintainability during engineering process operations. In this chapter, the introduction of new or altered systems into a complex engineering process environment is considered, whereby *safety* with respect to *risk* and *loss* through accidents or incidents resulting from the complex integration of systems is predicted, assessed and evaluated, to ensure that the design will have as minimum a risk as is reasonably practicable. Risk relates to the combination of the likelihood of occurring *hazards*, and to the severity of their outcome or consequence. An accident or incident may be viewed as an unintended event that results in either a critical or non-critical loss, and may include events such as death or personal injury, and environmental or financial losses, according to a relative scale of *safety criticality*.

Safety in engineering design starts by identifying the possible hazards of the new system, which are system states that can lead to an accident or incident. This is typically conducted through a series of collaborative *hazards analysis* sessions, during which keyword prompts and checklists are used to aid identification of hazardous system states. Suitably qualified experts representing all the areas that are relevant to the system being designed must participate in these sessions. Normally, a *causal analysis* is then conducted, which involves identifying various cause-effect sequences of hazardous events that may combine to cause the hazards already identified. Thereafter, a *consequence analysis* is conducted, which identifies the next sequences of events that could lead from a hazard to an accident or incident. Working through these phases of analysis, and iterating where appropriate, a *safety case* is prepared, which relates to an assurance that the system is relatively safe. This assurance is not a statement that the system is risk free—almost no system of any complexity can demonstrate this property. Instead, risks are typically divided into three categories, and each category is treated slightly differently.

The three categories of risks are the following:

- *Intolerable risks*:
  These are risks that are not acceptable under any circumstances—for example, the hazardous exposure to process products of a system that have a high likelihood of affecting workers occupational safety and health. The engineering design will need to include ways of removing such risks, or of drastically reducing their severity. The safety case must show that no such risks remain in the system.
- *Tolerable risks*:
  These are risks that are considered acceptable provided they confer some benefit, and the risk has been reduced as much as was reasonably practicable. The 'benefit' may be hard to measure objectively, especially in placing a cost value on accidents such as personal injury or death with respect to the cost of preventive measures. A typical example is the consideration of tolerable risks in the case of large construction projects of engineered installations during which accidents

and incidents are inevitable. The safety case would argue that there is a trade-off benefit of allowing certain risks at a given criticality level.

- *Negligible risks*:
  These are risks that are so small as to be insignificant, and no further precautions are considered necessary. The safety case would only include negligible risks that merit attention, such as those previously considered to be relatively significant risks.

Designing for safety entails definitive risk reduction measures and involves conducting or specifying mitigation strategies to, first, reduce the likelihood that a hazard will result in an accident or incident and, second, to aim at reducing the severity of the likely event. Because designing for safety strives for confidence in the results of these strategies, and the need for an objective systems scrutiny from a safety viewpoint, it typically involves systematic safety analysis, with independent safety prediction, safety assessment, and safety evaluation audits dovetailing with the respective schematic, preliminary and detail design phases of the overall engineering design process. Designing for safety tends to be both costly and time consuming because of the number of domain and other experts needed to determine those areas of high safety risk in the total integrated engineering design, the wide range of factors that need to be considered, and the implementation of additional safety control systems.

Techniques that are to be added into this work must therefore be cost and time effective, whilst fitting within existing as well as new methodologies in determining the integrity of engineering design.

Hazards and safety analyses provide a comprehensive methodology for *designing for safety*. The initial hazards analysis should begin at the earliest concept formation stages of systems design, and the information should be used to guide the emerging design with respect to safety requirements throughout the engineering design process. Later equipment hazards analysis information is used to evaluate the integrity of the design and to make trade-off decisions. The development of a *safety intent specification* supports both the evolution of systems design as well as system safety analysis. The design rationale for safety issues that are normally lost during the design's development stages is preserved in a single, logically structured document (or electronic database) that is based upon fundamental principles of human problem solving. Safety-related requirements and design constraints are traced from the highest systems levels, down through system design to component design and into hardware schematics and detail design specifications. An important feature of the safety intent specification is that it integrates formal and informal design specifications.

It is thus during the design stage of an engineering project when major improvements in safety and occupational health relating to construction, ramp-up and operation of an engineered installation can be achieved. However, there are real challenges involved in designing for safety in order to achieve the required step change in a safe and healthy environment in the construction and operation of industrial process plant and facilities. To date, there have been many factors that have limited improvements in this area, such as a lack of time and funding—besides the lack

of communication, understanding and commitment. The culture of a segmented engineering construction industry with its fragmented processes, along with the fact that many project clients are reticent in fully appreciating the significant added costs of designing for safety, must be critically addressed in order to break through into a new arena of safe working practices and performance. In appreciation of the challenges involved in designing for safety with the construction and operation of engineered installations, an agenda for change was developed at a major international conference on Designing for Safe and Healthy Construction, organised by the European Construction Institute (ECI) and the Conseil International du Bâtiment (CIB) in London in June 2000.

These changes—in particular with respect to changes required of process engineering designs—included the following (ECI 2001):

- Recognising the fact that engineering designs will dictate, to a considerable degree, the nature and extent of hazards that will pose a threat to worker safety and health, not only during construction but throughout the life cycle of the project.
- Concentrating on significant complex risks that competent contractors would not be expected to be aware of, rather than on easily identified residual risks.
- Achieving better risk identification methods.
- Utilising different levels of risk assessment at different stages in the project.
- Concentrating on interfaces between systems where high risks occur.
- Developing a better awareness of safe working practices and ergonomics.
- Making occupational safety and health (OSH) a top priority in the design process.
- Considering OSH implications in the earlier part of the engineering design process, such as safety predictions during the conceptual design phase.
- Recognising duty of care in considering OSH requirements in engineering designs, and its impact on construction activities.
- Maximising the use of innovative techniques and methodology that reduces OSH risk, such as pre-assembly and/or off-site manufacturing, and standardisation of equipment.
- Using the appropriate CAD systems to schematically examine the project during the preliminary design phase, to determine engineering design integrity.
- Using intelligent computer automated methodology for determining the integrity of engineering design through the application of automated continual design reviews throughout the engineering design process.
- Applying safety constructability reviews that contribute towards addressing construction worker safety in the design.
- Maintaining communication feedback and risk data to reduce unplanned construction work greater than required in the design.
- Designing for safe access for maintenance personnel to restricted areas, including access for routine and preventive maintenance and for installation of replacement equipment.
- Including risk analysis not only for construction, commissioning, ramp-up and operation but also for decommissioning or deconstructing of plant and facilities.

Safety engineering has also received much attention from the defence industry for several decades, particularly the US Department of Defence. The first military safety

document titled "System Safety Engineering for the Development of United States Air Force (USAF) Ballistic Missiles" was published in 1962. In 1963, the USAF published a document titled "Safety Engineering of Systems and Associated Sub-Systems and Equipment" (MIL-STD-38130 1963). This document was superseded in 1969 by a document titled "Requirements for Safety Engineering of Systems and Associated Sub-Systems and Equipment" (MIL-STD-882), which has subsequently been updated in 1977 (MIL-STD-882A), in 1984 (MIL-STD-882B), in 1993 (MIL-STD-882C) and in 2000 (MIL-STD-882D).

Additional military safety documentation covering *system safety* includes the following handbooks:

- the US Army handbook 'System safety design guide for army materiel' (MIL-HDBK-764 1994),
- the US Air Force Systems Command handbook 'System safety design handbook' (AFSC DH 1-6 1967),
- the US National Aeronautics and Space Administration (NASA) handbook 'System safety handbook' (NASA DHB-S-00 1999).

In any engineered installation, *human factors* are an important part of process control. Therefore, an effective safety program cannot consider only the automated systems hierarchy but must also consider the impact of human error on the system, and the effect of systems design on errors in human judgement and control.

Increased automation in complex systems has led to changes in the human controller's role, and to new types of technology-induced human error. Such errors abound in records of major process engineering catastrophes. In a detailed survey of safety incidents in the US nuclear power industry (INPO 84-027. 1984, 1985), it was revealed that of the roughly 1,000 identified root causes of incidents that were investigated, 51% were classified as "human performance problems", and 74% of these (i.e. 38% of all root causes) were "maintenance related", this being broadly defined to include preventive and corrective maintenance, surveillance testing and modification work.

The Three Mile Island nuclear power generator accident in 1979 demonstrated the significance of human error. The accident was attributed to mechanical failure *and* operator error. Despite the fact that about half of the reactor core melted, the containment building that housed the reactor prevented any release of radioactivity, and the reactor's other protection systems also functioned as designed. The emergency core cooling system would have prevented the accident but for the intervention of the operators. Investigations following the accident led to a new focus on the human factors in nuclear safety. No major design changes were called for in nuclear reactors but controls and instrumentation were improved and operator training was overhauled.

By way of contrast, the Chernobyl reactor in the Ukraine did *not* have a containment structure like those used in the West or in post-1980 Soviet designs. The April 1986 disaster at the Chernobyl nuclear power plant was the result of major design deficiencies in the type of reactor, the violation of operating procedures and the absence of a safety culture. The accident destroyed the reactor, killed 31 people, 28 of

whom died within weeks from radiation exposure. It also caused radiation sickness in a further 200–300 staff and fire fighters, and contaminated large areas of Belarus, Ukraine, Russia and beyond. It is estimated that at least 5% of the total radioactive material in the Chernobyl-4 reactor core was released from the plant, due to the lack of any containment structure. Most of this was deposited as dust close by. Some was carried by wind over a wide area. About 130,000 people received significant radiation doses (i.e. above internationally accepted ICRP limits) and have been closely monitored. About 800 cases of thyroid cancer in children have been linked to the accident. Most of these were curable, though about ten have been fatal. No increase in leukaemia or other cancers has been observed but some ongoing occurrences are expected.

The World Health Organisation is closely monitoring most of those affected. An OECD expert report concluded that "the Chernobyl accident has not brought to light any new, previously unknown phenomena or safety issues that are not resolved or otherwise covered by current reactor safety programs for commercial power reactors in OECD member countries" (OECD NEA 1995).

The IAEA has given a high priority to addressing the safety of nuclear power plants in Eastern Europe, where deficiencies remain. However, energy demand in these countries is such that there is little flexibility for closing even those plants that are of most concern, though the European Union is bringing pressure to bear, particularly in countries that aspire to EU membership. A major international program of assistance has been carried out by the OECD, IAEA and Commission of the European Communities to bring early Soviet-designed reactors up to near-Western safety standards, or at least to effect significant improvements to the plants and their operation. Modifications have been made to overcome deficiencies in the 13 reactors still operating in Russia and Lithuania. Automated inspection equipment has also been installed in these reactors as added safety precaution. Another class of reactors that has been the focus of international attention for safety upgrades is the first-generation of pressurised water reactors. These were designed before formal safety standards were issued in the Soviet Union, and they lack many basic safety features. Eleven are operating in Bulgaria, Russia, Slovakia and Armenia (ANSTO 1994). From 1996 on, the Nuclear Safety Convention (NSC) came into force as the first international legal instrument on the safety of nuclear power plants worldwide. It commits participating countries to maintain a high level of safety by setting international benchmarks to which they subscribe and against which they report. The NSC has 65 signatories and has been ratified by 41 states.

For the past two decades, the University of Washington, Seattle, WA, has been developing a theoretical foundation and methodology for analysing safety in complex systems, under grants from the US National Aeronautics and Space Administration (NASA Langley, NASA Ames). The methodology includes safety analysis, system hazard analysis, control software design, and special techniques for the design of human–machine interaction (Leveson 1995). What is especially appealing in this methodology is that it not only formulates system safety using control software in system automation for enhanced control of complex integrations of systems but also considers *human error analysis*.

The problem of technology-induced human error on process systems control has been approached in two ways in *designing for safety*.

The first approach is the detection of *error-prone automation features* early in the conceptual design phase of the engineering design process while significant changes can still be made. The information produced from this approach can be used to redesign process automation to eliminate any error-inducing features, or to design better human-machine interfaces, process operator procedures, and training programs.

The second approach to safety analysis of human error is the more traditional form of *human factors analysis*. This method looks at the types of human errors that could arise in the system, and then performs a *comparative analysis* of the controller's job before and after system automation control. Potential safety issues are identified that involve decreased awareness, increased vigilance requirements, and skills degradation. Identification, classification and evaluation of potential hazards are done through modelling and analysis in which the hardware, software, as well as human components in the system are considered.

Risk in engineering design may simply be described as the process of risk analysis of hazardous systems at the conceptual, preliminary and detail design phases, with respect to risk prediction, risk assessment and risk evaluation respectively. The risk analysis process in engineering design is both iterative and progressive, in that it is composed of five basic steps that are repeated for each progressive design phase as the design becomes increasingly complex and detailed. These five steps include the following:

- Design definition
- Hazards definition
- Risk estimation
- Risk verification
- Results application.

Design definition entails defining the system under consideration according to the level of detail achieved at each particular design phase. Thus, at the *conceptual design phase*, the process and major systems are defined together with environmental conditions and general system physical and functional boundaries.

At the *schematic* or *preliminary design phase*, the systems are reviewed inclusive of their major items of equipment (predominantly sub-systems and assembly sets), together with integrated systems conditions and specific equipment physical and functional boundaries.

At the *detail design phase*, the systems are reviewed in greater depth to include all items of equipment (e.g. assemblies and components) as well as major parts of components, together with intrinsic system conditions and component physical and functional boundaries.

Hazards definition is concerned with the identification of hazards that are evident at each progressive level of design detail in the systems hierarchical structure. This step includes estimates of the *significance* of the identified hazards, whereby each phase of the risk analysis process results not only in an accumulation of potential

hazards but also in the elimination of hazards that are found to be non-significant through progressive clarity of the level of detail achieved at each design phase.

Analysis of hazards is done either through *causal analysis* or through *consequence analysis*, or both, depending on the need to identify causes or consequences of the hazards respectively. Identifying the causes of hazards usually makes use of techniques such as root cause analysis, whereas consequence analysis makes use of systems engineering analysis.

Risk estimation may be perceived as the application of a variety of methods and techniques for risk prediction, risk assessment and risk evaluation. The prediction of risk is usually at a higher process and systems level with minimal clarity on detail, and is fundamentally useful in determining the configuration (inclusion of parallel redundancy) and initial sizing (maximum strength-stress safety margins) of the engineering design.

Risk assessment is usually conducted at equipment level, and includes investigation of potential sources of hazards to determine the probability/likelihood of occurrence of the originating hazard and its associated consequences for the system's operation as a whole. Risk assessment may also be targeted at the component level whereby functional failures are identified based on the severity of their intrinsic effects and the likelihood of their occurrence.

Risk verification is basically concerned with verifying the suitability of the risk estimation techniques and their end results. It is fundamentally a design review process used to determine the integrity of engineering design through verification of the risk estimates. This is accomplished by considering the relevance and suitability of the various risk estimation and analysis methods with respect to their appropriateness in analysing the type of system and hazard being studied, as well as the format of the results with respect to a correct understanding of the priority, occurrence and severity of the risk.

Results application effectively incorporates the contents and results of the previous four risk analysis steps with the application of automated continual design reviews in concurrent engineering design throughout the engineering design process. In this research, these design reviews are modelled in an *artificial intelligence-based (AIB) blackboard system* that is targeted for use by multi-disciplinary groups of design engineers, whereby each designed system is evaluated for integrity by locally or remotely located design groups communicating via an intranet or via the internet, within an integrated collaborative engineering design environment. The reviews should contain information such as systems definition, analysis methodology and associated assumptions and limitations, modelling descriptions, quantitative data and methods of accumulation, the specific techniques of risk estimation and the results obtained, together with a discussion of the results, associated assumptions and sensitivity analysis—all within intelligent computer automated methodology for determining the integrity of engineering design.

5.2 Theoretical Overview of Safety and Risk in Engineering Design

Safety, in contrast to risk, is a system property, not a component property. Therefore, safety analysis must consider the entire system, and not its component parts. However, there exists no single safety analysis technique that can cope with all aspects of complex systems or complex integrations of systems. Safety analysis of complex systems is an inter-disciplinary effort, and must include systems design engineers, software engineers, and human factors and cognitive experts. Safety analysis in engineering design is, in effect, a program consisting of systems design activities and special safety tasks and techniques that significantly interact with one another at each progressive phase of the engineering design process. Such a program is highly iterative and includes continual updating of what has been done previously in the earlier phases, as new information and clarity of the design are gained. A safe system is one that is free from accidents or incidents resulting in unacceptable losses. Accidents or incidents result from hazards, where a hazard is defined as a system state or condition that can lead to an accident or incident, given certain uncontrollable or unpredictable environmental conditions.

Thus, safety in engineering design starts with a *hazards analysis* that identifies and analyses the system for hazards. Once these hazards have been identified, steps can be taken to eliminate these, reduce their likelihood, or mitigate their effect. In addition, some hazard causes can be identified and eliminated or controlled. Although it is usually impossible to anticipate all potential causes of hazards, obtaining more information about these usually allows greater protection to be provided with fewer trade-offs, especially if the hazards are identified in the early design phases. Identifying hazards, and hazard causes, enables safety requirements to be established during the engineering design process.

A *hazard* may be defined as "*a source of potential harm or a situation with a potential for harm*", where *harm* is defined as "*a physical injury or damage to health, property or the environment*". Furthermore, an *accidental event* is defined as "*an event which can cause harm*" (IEC60300-3-9 1995). A hazard may thus lead to an accidental event. To create a sound basis for further analysis, all the hazards have to be identified in a systematic way. A commonly used technique for such a survey is *hazard identification (HAZID)*.

Hazard identification (HAZID) analysis is usually carried out in the early design phases of a system. The objective of the analysis is to reveal potential hazards at an early stage, such that the hazards may be eliminated, minimised or controlled as early as possible in the development process. For each hazard that is identified, all possible causes, effects and severity of potential accidents are described. Possible improvements and precautions are also described. It is important that this analysis is based on previous experience with similar equipment. Checklists of various types are useful during the analysis. The analysis should be conducted with one or two experienced engineers in attendance, with a background in safety engineering. Since the HAZID analysis is carried out in the early phases of the engineering design

process, a limited amount of information about the specific system will normally be available. For a process plant, the process concept has to be settled before the analysis is initiated. At that point in time, the most important chemicals and reactions are known, together with the main elements of the process equipment (vessels, pumps, etc.). The HAZID analysis must be based on all safety-related information about the system, with respect to design criteria, equipment specifications, specifications of materials and chemicals, operational procedures, previous hazard studies of similar systems, and previous accident details if available.

The following input information should be available:

- Design sketches, drawings and data describing the system and sub-system elements for the various conceptual approaches under consideration.
- Functional flow diagrams and related data describing the proposed sequence of activities, functions and operations of the system elements during the contemplated life span.
- Background information relating to safety requirements associated with the contemplated testing, manufacturing, storage, repair and use locations, and safety-related experiences of similar previous programs or activities.

The HAZID analysis is conducted by identifying hazards and thereby potential accidental events that may lead to unwanted consequences. The analysis must also identify design criteria or alternatives that may eliminate or reduce the hazard. During the analysis, certain factors must be considered (AIChE 1992).

These factors are:

- Hazardous equipment and materials
  (e.g. fuels, highly reactive chemicals, toxic substances, explosives, high-pressure systems, and other energy storage systems)
- Safety-related interfaces between equipment and materials
  (e.g. material interactions, fire/explosion initiation and propagation, and control/shutdown systems)
- Environmental factors that influence the equipment and materials
  (e.g. storms, earthquakes, vibration, flooding, extreme temperatures, electrostatic discharge, and humidity)
- Operating, testing, maintenance and emergency procedures
  (e.g. human error, operator functions, equipment layout and/or accessibility, and personnel safety protection)
- Facility support
  (e.g. storage, testing equipment, training and utilities)
- Safety-related equipment
  (e.g. safety device, fire suppression, personal protective equipment).

Some hazards can be identified by the following (Rausand 2000):

- examining similar existing systems,
- reviewing existing checklists and standards,
- considering energy flows through the system,

- considering inherently hazardous materials,
- considering interactions between system components,
- reviewing previous hazard analyses for similar systems,
- reviewing operation specifications,
- considering all environmental factors,
- considering human/machine interfaces,
- considering usage mode changes,
- trying small-scale testing, and theoretical analyses,
- thinking through a worst-case scenario, what-if analysis.

The results from a HAZID analysis are usually presented in a specific HAZID worksheet, identifying the hazards, the causes, the potential consequences, and possible improvements and precautions. In most applications, it is relevant to start with the accidental events. A generic list of hazards may often be useful in supporting a brain-storming process to identify potential accidental events. A number of similar methods with other names are also used. Among these are preliminary hazard analysis (PHA), and rapid risk ranking (RRR).

The preliminary hazard analysis technique was developed by the US Army (MIL-STD-882C 1993), and has been successfully used within defence-related industries, and for safety analysis of engineering processes.

Hazard severity categories are defined to provide a qualitative measure of the worst credible mishap resulting from personnel error, environmental conditions, design inadequacies, procedural deficiencies, or system, sub-system or component failure or malfunction. The starting point of a HAZID worksheet is defining the potential accidental events. The worksheet has a column called 'hazard category'. In this column, the *severity* of the potential consequences is ranked. MIL-STD-882 requires that such a column for severity ranking be part of the HAZID (or PHA) worksheet. These hazard severity categories are defined in the Military Standard under the sub-title of 'hazard severity', and are presented in Table 5.1. An example of a HAZID worksheet is shown in Table 5.2.

In some cases, it may be relevant to include a more detailed severity ranking, e.g. distinguishing between human, environmental, and process or product consequences. Such a ranking depends on the actual context of the consequences, i.e. the risk assessment scale is given later in Table 5.7.

Table 5.1 Hazard severity ranking (MIL-STD-882C 1993)

Hazard severity	Description category	Mishap definition
I	Catastrophic	Death or system loos
II	Critical	Severe injury, severe occupational illness, or major system damage
III	Marginal	Minor injury, minor occupational illness, or minor system damage
IV	Negligible	Less than minor injury, occupational illness, or system damage

Table 5.2 Sample HAZID worksheet

System: Acid separation plant **Subsystem:** Precipitation tanks—piping **Drawing:** Al-ASP/PT 02-004				**Date:** Nov. 2006 **Page:** 11 of 32	
Reference	Accidental event	Probable causes	Major effects	Consequence/ severity	Corrective/ preventive action
HC piping	Gas leak/losses	CO_2 corrosion	Thin wall cracking	Safety catastrophic I	Ultrasonic corrosion probe
Slurry piping	Containment losses	Pipe wall penetration	Lack of inhibition	Environment critical II	Inhibitor piping injection check
Water piping	Containment losses	External erosion	Damage/water traps	Environment marginal III	Visual inspections

Table 5.3 Categories of hazards relative to various classifications of failure

Hazard category			
by cause	by effect	by consequence	by severity
Stress-related failure	Immediate functional	Critical safety	Catastrophic
Failure due to misuse	Gradual degradation	Critical operational	Critical
Failure due to damage		Major functional	Marginal
Failure due to weakness		Minor functional	Negligible
Failure due to wear-out		Hidden failure	
Maintenance failures		Non-operational	

Hazard consequences depend on the cause-effect nature of functional failure within a system as well as system states that define system hazards. The various combinations of the different defining categories of hazards (i.e. by cause, effect, consequence and/or severity), relating to the various *classifications of failure*, are presented in Table 5.3.

Hazards analysis can take the form of *backward analysis*, alternatively termed *causal analysis*, or of *forward analysis*, alternatively termed *consequence analysis*, depending on the need to identify causes or consequences respectively. Hazard analysis techniques that use backward search or analysis begin with a hazardous state and then determine the events that could lead to this state. The analysis starts from hazards identified at the process and/or systems level, and identifies their precursors further down in the systems hierarchical structure. Thus, in the case of backward analysis, the analysis is causal and the search is top-down. Conversely, in hazard analysis techniques that use forward search or analysis, in which the next sequences of events that could lead from a hazard to an accident or incident are identified, the analysis is consequential and the search is bottom-up.

Information that is derived from both backward and forward analysis, i.e. *cause-consequence analysis*, relates to recognisable failed system states that can be used to redesign the system to prevent or minimise the probability of occurrence and/or the severity of the hazard. In order to be able to apply causal analysis techniques, such as

fault-tree analysis (FTA), a more detailed specification or model of the behaviour of the system is required at the lower equipment levels of the systems hierarchy (e.g. assembly, sub-assembly and component levels). A high-level system design may appear to be safe, while the detailed design contains hazardous equipment interactions inherent within the system. However, these interactions may *not* be inherent within a single system and may arise only on the functional interface of equipment belonging to different systems, due to a complex integration of the various systems by design. The hazards and design constraints must be traced right down to the system components where feasible, so that assurance may be provided that the hazards have been eliminated or mitigated. Although theoretically this type of analysis can be comprehensively performed only on the detail design of the system, the practical approach is to progressively structure the design according to *systems hierarchical modelling* that is continually enhanced as the emerging systems breakdown structure (SBS) becomes more definite with each phase of the engineering design process. The use of object oriented programming (OOP) *simulation modelling* that provides graphic displays (preferably animated) of the various systems and equipment of the design enables realistic three-dimensional visualisation of the model appropriate to the system's domain. The OOP simulation must have the capability of backward and forward processing to accommodate both causal and consequence analysis respectively.

5.2.1 Forward Search Techniques for Safety in Engineering Design

Forward search techniques begin with an initiating event and trace it forwards in time or in effect. At the higher systems levels, the appropriate forward analysis technique is consequence analysis. As indicated previously in Sect. 3.3.2.1 dealing with failure modes and effects analysis, the *consequences of failure* are associated with the overall results that occur in the system or process as a whole, whereas the *effects of failure* are associated with the immediate results that initially occur within the assembly's or component's environment.

Thus, at the lower systems levels, the appropriate forward analysis techniques are failure modes and effects analysis (FMEA) and failure modes and effects criticality analysis (FMECA) or, in the case of safety analysis, event tree analysis (ETA), hazards and operability (HAZOP) studies, and failure modes and safety effects (FMSE) analysis. The difference between FMSE and FMECA is in the construct of the inductive FMSE spreadsheet that, in addition to the standard columns of an FMECA, includes safety-related aspects such as failure root causes, integrity measures, and inspection methods.

Hazard analysis thus conventionally includes the following deductive and inductive analysis techniques:

- Fault-tree analysis (FTA)
- Root cause analysis (RCA)

- Event tree analysis (ETA)
- Cause-consequence analysis (CCA)
- Hazardous operability studies (HAZOP).

These techniques have been developed for visualising the sequence of events in the operation of a complex engineered system and for estimating the probability of occurrence of the end result. They start either with an expected, unwanted effect (i.e. fault) and work backwards to the logical cause, or with a proposed cause (i.e. event) and proceed forwards through the relevant analysis steps, ending up with an end result effect and/or several effects. As indicated in Sect. 3.2.2, backward analysis techniques are *deductive*, whereas forward analysis techniques are *inductive*. These are often termed 'top-down' or 'bottom-up' procedures that emulate the diagrammatical arrangement representing the systems hierarchical structure that is used to define the path from effect to cause, or from cause to effect respectively.

Fault-tree analysis is a typical application of deductive analysis, in which the analysis begins with the system in a hazardous state and then works backwards one step at a time, during which irrelevant branches of possible causes can be omitted, or specific branches of greater significance can further be followed. However, applying a *reachability graph* to visualise the structural extent of backward analysis of process engineering systems, it becomes evident that the graph explodes quickly for complex systems, and in itself becomes complex (Leveson 1995). Many of the branches of the structural graph are either incomplete or impossible to pursue, requiring alternative analytic approaches, which will be considered in Sect. 5.3. Figure 5.1 shows the format for a fault-tree analysis that is a 'top-down' deductive analysis method, and here the branch points lead to the typical question:'*What are the conditions that lead to this point?*'.

Root cause analysis utilises the deductive logic tree approach, similar to fault-tree analysis, in establishing the root causes of a problem, whether it is a functional failure or a system state. Such a logic tree approach to problem solving is particularly useful for determining safety in detail engineering designs. By organising problem analysis results in an orderly manner as the design progresses, the time spent to find the root causes of possible problems is minimised. The method uses *factor trees* to guide the course of the analysis. These factor trees diagrammatically present the major functions to be considered in the design's project stages, and provide an excellent method for sorting out facts and zeroing-in on the root causes of problems.

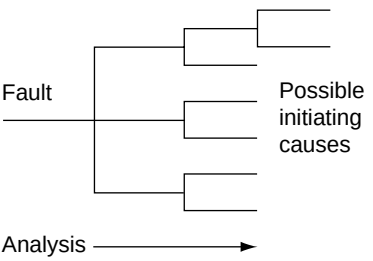


Fig. 5.1 Fault-tree analysis

Event tree analysis, unlike fault-tree analysis, uses inductive logic. This technique is a method for illustrating the sequence of outcomes that may arise after the occurrence of a selected initial event. It is mainly used in consequence analysis for pre-incident and post-incident application. The left side connects with the initiator, the right side with plant damage state; the top defines the systems; nodes (dots) call for branching probabilities obtained from the system analysis. If the path goes up at the node, the system succeeded, if down, it failed. Event trees have been applied in the nuclear industries for operability analysis of nuclear power plant as well as for accident sequence in the Three Mile Island nuclear power generator accident (INPO 84-027 1984).

Figure 5.2 shows an event tree format classified as a 'bottom-up' inductive analysis method. Here, the branch points follow a YES or NO criterion for a specific question of the type '*is valve V1 closed?*'.

Cause-consequence analysis is a combination of deductive analysis and of inductive analysis. This technique combines cause analysis (described by fault trees) and consequence analysis (described by event trees). The purpose of cause-consequence analysis is to identify chains of events that can result in undesirable consequences. With the probabilities of the various events in the CCA diagram, the probabilities of the various consequences can be calculated, thus establishing the risk level of the system. This technique was developed by RISO Laboratories in Denmark to be used in risk analysis of nuclear power stations (Aven 1992). It can also be adapted for process engineering in the estimation of the safety of protective systems.

Figure 5.3 shows a layout of a cause-consequence analysis that is both a 'top-down' *deductive analysis* and a 'bottom-up' *inductive analysis*.

These tree-based methods are used mainly to find cut sets leading to the undesired events. Fault trees and event trees have been widely used to quantify the probabilities of occurrence of accidents and other undesired events leading to the loss of life or economic losses in probabilistic risk assessment. However, use of fault tree and event tree analysis techniques is usually confined to static, logic modelling of accident scenarios, and does not cover risk assessment for dynamic systems (Siu 1994).

Methodologies for the analysis of dynamic systems include techniques such as *digraphs* or fault graphs, dynamic event logic, as well as Markov modelling, which will be considered later. In giving the same treatment to process failures and human errors in fault tree and event tree analysis, the conditions affecting human behaviour cannot be modelled explicitly. This affects the assessed level of dependency between events. Relatively new techniques such as human cognitive reliability (Gert-

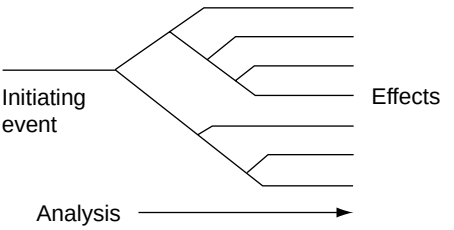


Fig. 5.2 Event tree

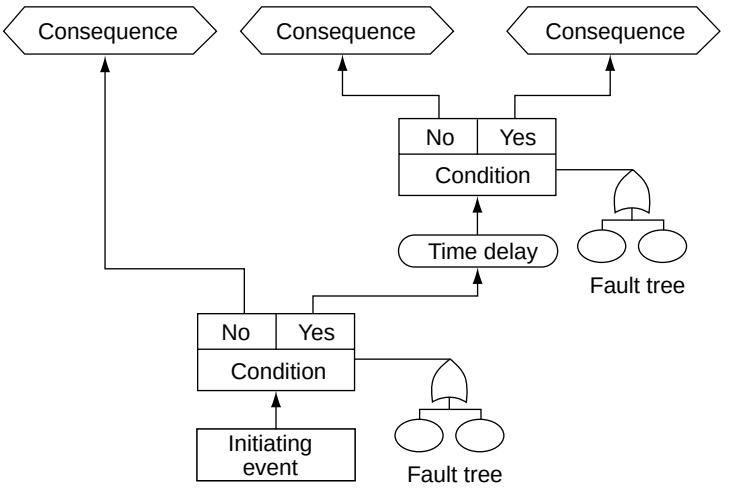


Fig. 5.3 Cause-consequence diagram

man et al. 1994), and programmable user modelling applications (Blandford et al. 1999) have emerged to reconcile deficiencies in the tree-based analysis techniques.

Furthermore, although the use of techniques are adequately suitable in designing for safety of process engineering designs, their use in designing for systems control is complicated by the large number of ways that computational control can address, or even contribute to, hazardous system states. This problem is solved by the use of a relatively new forward analysis technique called *deviation analysis* (Leveson 1995).

Deviation analysis (DA) is based on the underlying assumption that many accidents or incidents are the result of deviations in system variables, where a deviation is the difference between the actual and correct values appropriate for system control. The method originates from the forward analysis technique of software deviation analysis (SDA) in which hazardous behaviour in system control software is analysed. DA is an extension of the technique to system control hardware. Deviation analysis determines whether hazardous systems behaviour can result from a class of input deviations inclusive in the broad range of process characteristics such as capacity, input, throughput, output and quality. It is a means of determining system component robustness (or, in safety terminology, its *survivability*), or how it will behave in an imperfect environment.

Hazardous operability studies (*HAZOP*, short for *haz*ard and *op*erability), was first introduced by engineers from ICI Chemicals in the UK, in the 1970s. The method entails the investigation of *deviations* from the design intent for a process engineering installation by a design team with expertise in different areas such as engineering, operations, maintenance, safety and chemistry. The team is guided in a structured process, by using a set of guidewords to examine deviations from normal process conditions at various key points (nodes) throughout the process. The guidewords are applied to the relevant process parameters—for example, flow, tem-

perature, pressure, composition, etc.—in order to identify the *causes and consequences* of deviations. Typical terms used in a HAZOP are the following (Kletz 1999):

- *Node*: a specific location in the process in which (the deviations of) the process *intention* are evaluated.
- *Intention*: description of how the process is expected to behave at the node; this is qualitatively described as an activity (e.g. feed, reaction, sedimentation) and/or quantitatively in the process parameters, like temperature, flow rate, pressure, composition, etc.
- *Deviation*: a way in which the process conditions may depart from their intention.
- *Parameter*: the relevant parameter for the condition(s) of the process; e.g. pressure, temperature, composition, etc.
- *Guideword*: a short word to describe a deviation of the intention. The mostly used guidewords are NO, MORE, LESS, AS WELL AS, PART OF, OTHER THAN and REVERSE. In addition, guidewords like TOO EARLY, TOO LATE, INSTEAD OF, etc. are used, the latter mainly for batch-like processes. The guidewords are applied, in turn, to all parameters, in order to identify unexpected and yet credible deviations from the intention.
- *Cause*: the reason(s) why the deviation could occur. Many causes could be identified for one deviation.
- *Consequence*: the results of the deviation, in case it occurs. Consequences may comprise both process hazards and operability problems, like plant shutdown or quality decrease of the product. Many consequences can follow from one cause and, in turn, one consequence can have several causes.
- *Safeguard*: facilities that help to reduce the occurrence frequency of the deviation or to mitigate its consequences. There are five types of safeguards:

  a) Facilities that *identify* the deviation. These comprise, among others, alarm instrumentation and human operator detection.
  b) Facilities that *compensate* the deviation, e.g. an automatic control system that reduces the feed to a vessel in case of overfilling (increase of level). These usually are an integrated part of the process control.
  c) Facilities that *avoid* the deviation from occurring.
  d) Facilities that *prevent* deviation from escalating (e.g. trips). These facilities are often interlocked with several units in the process, and controlled by logical computers.
  e) Facilities that *relieve* the process from the hazardous deviation. These comprise, for instance, pressure safety valves (PSV) and vent systems.

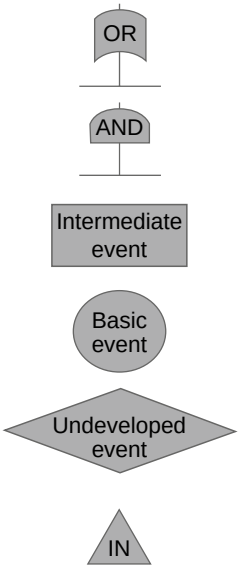
- *Recommendation*: activities identified during a HAZOP study for follow-up. These may comprise technical improvements in the design, modifications in the status of drawings and process descriptions, procedural measures to be developed or further in-depth studies to be carried out.

5.2.1.1 Fault-Tree Analysis for Safety in Engineering Design

The concept of *fault-tree analysis (FTA)* was originated by Bell Telephone Laboratories in the 1960s as a technique to perform a safety evaluation of the Minutemen Intercontinental Ballistic Missile Launch Control System. A fault tree is a logical diagram that shows the relation between system failure, i.e. a specific undesirable event in the system, and failures of the components of the system. It is a technique based on deductive logic. An undesirable event is first defined and causal relationships of the failures leading to that event are then identified. Fault trees can be used in qualitative or quantitative risk analysis. The difference between the two is that the qualitative fault tree is linguistic in structure and does not require use of the same rigorous logic as does the formal quantitative fault tree (cf. Fig. 5.4).

FTA is a deductive technique that focuses on a particular accident or failure, and provides a method for determining causes of that event. Fault-tree diagrams use logical operators, principally the OR and AND gates. The terminology is derived from electrical circuits, the term 'gate' referring to the control of a signal or electrical current. The term OR denotes a choice between two or more signals, either of which can 'open' the gate. The AND term refers to the requirement that both signals are necessary before there is an output from the gate. Figure 5.4 shows the logic and event symbols used in FTA.

Fault-tree analysis for safety in engineering design is conducted in several steps, from defining the problem to constructing the fault tree, analysing the fault tree, and documenting the results, specifically:



OR gate
The output event occurs if any of the input events occur

AND gate
The output event occurs only when all input events occur

Intermediate event
A fault that results from the interactions of other fault events

Basic event
A component failure that requires no further development

Undeveloped event
A fault that is not examined further because information is unavailable

Transfer IN/OUT symbols
IN indicates the tree is developed further at a corresponding OUT symbol

Fig. 5.4 Logic and event symbols used in FTA

Step 1. Defining the Problem

The engineering design team selects:

- the top event,
- the boundary conditions,
- system physical bounds,
- the level of systems resolution,
- initial conditions,
- events that are not allowed,
- existing conditions,
- conditional assumptions.

Defining the top event is one of the most important aspects of the first step. The top event is the accident (or undesired event) that is the subject of the FTA. The top event is often identified through other hazard analysis studies (such as HAZID). Top events should be precisely defined for the system or plant being evaluated, because analysing broadly scoped or poorly defined top events can often lead to an inefficient analysis.

For example, a top event of 'gas leaks in the plant' is too general. Instead, an appropriate top event would be 'gas leak in the HC piping of the acid separation plant precipitation tank B'.

The physical system boundaries encompass the system's equipment, the equipment's interfaces with other processes, and the utility/support systems that are to be included in the FTA. The design team should also specify the level of systems resolution for the fault-tree events. For example, a motor-operated valve can be included as a single item of equipment (i.e. component) or it can be described as several hardware items (i.e. parts, e.g. the valve body, valve internals, and motor operator). The systems resolution of the FT should be limited to the detail needed to satisfy the analysis objective, and should parallel the resolution of the available information.

The initial equipment configuration or initial operating conditions describe the system in its normal, unfailed state. Events that are not allowed are, for the purposes of the FTA, events that are considered to be unlikely or that are not to be considered in the analysis, for some exclusive reason. For example, wiring failures may be excluded from the analysis of an instrument system, or cabling may be excluded from the analysis of power generating units. Existing conditions within which the system functions are estimates (and assumptions) of the possible operational conditions that may arise within the system and its equipment, either as a result of the system's inherent complexity, or as a result of the complex integration of various systems.

Step 2. Constructing the Fault Tree

The FTA begins at the *top event* and proceeds, level by level, until all fault events have been traced to their basic contributing causes (i.e. basic events). At each level,

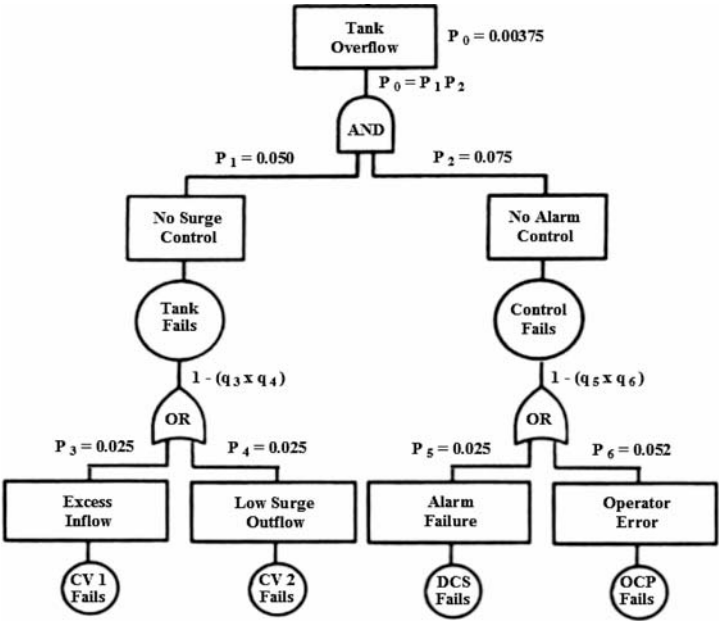


Fig. 5.5 Safety control of cooling water system

the immediate, necessary and sufficient causes are defined that would result in the intermediate or top event under consideration. The analysis continues at each level, until basic causes or the analysis boundary conditions are reached.

Returning to the simple fault tree of a cooling water system depicted in Fig. 3.19 of Sect. 3.2.2.6 dealing with fault-tree analysis in reliability assessment, assume that the systems design included provision for a back-up surge tank with an appropriate control alarm in the event the tank over-flowed, indicating problems with the cooling water feed. These problems would typically be:

Excess inflow.
Low surge outflow.
Control alarm failure.
Operator error.

Figure 5.5 shows an example of the cooling water surge tank fault tree with two levels below the top event.

Step 3. Analysing the Fault Tree

The analysis ‘solves’ the fault tree by identifying combinations of failures that can lead to accidents. These are called minimal cut sets (MCS). The minimal *cut sets* for the example shown in Fig. 5.5 would be:

- 'No surge control' and 'No alarm control'
- 'Excess inflow' and 'Alarm failure'
- 'Excess inflow' and 'Operator error'
- 'Low surge outflow' and 'Alarm failure'
- 'Low surge outflow' and 'Operator error'.

If the states of each of the control valves (CV1 and CV2) are in failure mode (i.e. failed closed and failed open), then further low-level cut sets can be defined, and the fault tree needs to be modified (additional rectangular boxes above each CV circular box) to include the *failed states*:

- 'CV1 fails open' and 'Alarm failure'
- 'CV1 fails closed' and 'Alarm failure'
- 'CV2 fails open' and 'Alarm failure'
- 'CV2 fails closed' and 'Alarm failure'.

Failure probabilities can now be assigned. The probabilities that are allocated to the events can be combined to estimate the probability of the top event. The probability of two events, the one with probability p_1 and the other with probability p_2, occurring together are:

$$P(\text{AND}) = p_1 \times p_2 \tag{5.1}$$

and q_1 and q_2 are the complements of p_1 and p_2 respectively:

$$q_1 = 1 - p_1$$
$$q_2 = 1 - p_2$$

Then: q_1 is 'NOT p_1' and: q_2 is 'NOT p_2'.

The probability of event 1 not occurring is thus q_1 and the probability of event 2 not occurring is q_2. Thus, for event 1 OR event 2 to occur, the probability of the combination that either does *not* occur—that is, that one of the two occurs—is given by the following expression:

$$P(\text{OR}) = 1 - (q_1 \times q_2) \tag{5.2}$$

The concept of this expression can be clarified by the following example. In Fig. 5.5, the probabilities of the equipment failures in the circles are derived from expert judgement, and the activities in the rectangular boxes are calculated from frequencies further down the tree.

The probability for no surge control is calculated as:

$$\begin{aligned} P(\text{OR valves}) &= 1 - [(1 - 0.025) \times (1 - 0.025)] \\ &= 0.050 \end{aligned}$$

The probability for no alarm control is calculated as:

$$\begin{aligned} P(\text{OR alarm}) &= 1 - [(1 - 0.025) \times (1 - 0.052)] \\ &= 0.075 \end{aligned}$$

The probability for the top event shown in the figure (tank overflow) is:

$$\begin{aligned} P(\text{AND})\ \text{tank} &= 0.050 \times 0.075 \\ &= 0.00375 \end{aligned}$$

Although the example is hypothetical, it closely resembles a real-world scenario in which it is interesting to note that the safety alarm control system's reliability is lower than that of the surge system it is meant to control! This is due to operator error where operator judgement is jeopardised by failure in the operator control panel (OCP)–which, in many processes, is often the case. The failure of an item of equipment will result in its replacement, which reduces the failure frequency, and which then changes the risk probabilities all the way up the tree.

The use of computer models is necessary to maintain the fault-tree analysis up to date. It is common in large process plants, however, for the maintenance group not to communicate these improvements to the reliability engineers who continue to use outdated high-risk numbers. Similarly, experiences of ineffective operation will usually initiate improved training, so that operator errors are less frequent and the reliability of the whole system is improved.

Step 4. Documenting the Results

The analysis should provide a description of the system, a discussion of the problem definition, a list of assumptions, the fault-tree model(s) that were developed, lists of minimal cut sets, and an evaluation of the significance of the MCSs and any recommendations that arise from the FTA.

Probability evaluation of fault trees is considered in most technical papers and books about safety and hazard analysis. However, some approximation discrepancies are evident, especially in the basic theory of assigning probabilities to the fault-tree gates—specifically, the OR gate.

The probability expression for the statistically independent input events for the OR gate has been given as, (Dhillon 1983):

$$\begin{aligned} P(\text{OR}) &= P(a+b+c+\ldots \text{etc.}) \\ P(\text{OR}) &= P(a)+P(b)+P(c)+\ldots \text{etc.} \\ a,b,c,\ \text{etc.} &= \text{input events}\,. \end{aligned} \tag{5.3}$$

In the example of Fig. 5.5, this is equivalent to:

$$\begin{aligned} P\,(\text{OR}) &= p_3+p_4 \ \text{ or } p_5+p_6 \\ &= 0.050 \quad \text{or } 0.077 \end{aligned}$$

Considering the complements of p_1 and p_2, namely q_1 and q_2, results in:

$$\begin{aligned} P\,(\text{OR}) &= 1-(q_3 \times q_4) \ \text{ or } 1-(q_5 \times q_6) \\ &= 0.049375 \quad \text{or } 0.0757 \end{aligned}$$

5.2.1.2 Root Cause Analysis for Safety in Engineering Design

Root cause analysis is predominantly a technique for determining the origin of causes of failure in engineered installations *after* completion of their design. However, the approach can also be used to identify potential root causes of failure, particularly failures with critical safety consequences, during the engineering design process *before* systems manufacture, installation and/or construction. The fundamental need for design engineers to consider how their designs operate in the field and, more importantly, how they fail is imperative to successfully achieving integrity in engineering design. This will ultimately result in engineering designs that satisfy both functional and integrity requirements, using sound engineering judgement, rather than 'crystal ball' prediction techniques.

Although there is a wealth of knowledge and data concerning systems performance of existing engineered installations, in general this is not utilised to the extent that information may be obtained for use in new designs, especially in complex integrations of designs. To this end, more formal and systematic methods should be introduced during the engineering design process.

Although specific methods and tools are available to facilitate designing for reliability, for example, their use is often limited to reliability engineers, with the design engineers of other disciplines frequently adopting an intuitive approach to considering reliability in design. As the design process becomes increasingly sophisticated with higher-level design tasks of complex integrations of similarly complex systems, it has become essential that design engineers formally investigate the integrity of these designs, particularly at each interface of the integrated systems.

Examining and understanding the *root cause* of failure of a design's functional operation can aid in designing for safety and designing-out unreliability. In selecting equipment from an existing design to meet a new requirement within different systems integration, it is important that design engineers look beyond the standard reliability metric of the existing design, and review in particular the root causes of failure and significant factors affecting the equipment's reliability and safety. In the past, there has been an over-reliance on the use of prediction methods. For example, the original reliability prediction handbook of the USA Department of Defence (DoD), MIL-HDBK-217, contained failure rate models for the various part types used in electronic systems, and concentrated mainly on the use of prediction methods that did not provide engineers with any knowledge of what might fail in service (MIL-HDBK-217F 1998).

A methodology aimed at integrating reliability enhancement practices into the engineering design process has been developed as part of a UK government and aerospace industry initiative. As a result, the Reliability Enhancement Methodology and Modelling (REMM) project was funded in part by the UK's Department of Trade and Industry through the Civil Aviation Research and Development program and by industrial partners involved (Marshall et al. 1998). The main objectives of the project are to develop a methodology that supports reliability enhancement in engineering design and to develop a model that facilitates reliability assessment throughout a *system's life cycle*. REMM is primarily used within the aerospace

environment but the methodology and model developed are equally applicable to other high-reliability system designs, such as in process, chemical and mechanical engineering design projects. A number of simple practical analyses for use by design engineers, during the early stages of systems realisation, have been developed as part of the REMM methodology. These analyses are aimed at improving high-level decision-making using simple graphical representations of reliability data, such as analyses of root causes, trends, and manufacturing data.

These graphical representation analyses include:

- Root cause analysis and classification of events into high-level failure categories, providing the means to determine those factors that have most effect on the system's service reliability and, hence, which elements should be tackled as a priority.
- Root cause and trend data across specific criteria such as equipment type, periods of time (e.g. particular manufacturing time-line points), application or use, providing further understanding of the nature of the failure that may be characteristic of the environment in which it is operating.
- Manufacturing data analysis, providing valuable insight into the factors that affect service reliability. Correlation between manufacturing methods and service requirements can often illuminate small changes in design and manufacturing process that result in significant effects on service reliability.

Root cause analysis also utilises the deductive logic tree approach, similar to fault-tree analysis (FTA), in establishing the root causes of functional failure or of a system state. Such an approach to problem solving is particularly useful for determining safety in engineering designs.

The approach of establishing the root causes of functional failure in systems design is intended to achieve the following:

- To organise and control design integrity problem identification.
- Provide a visual checklist to ensure all pertinent areas are covered.
- Allow for a standardised approach to safety problem identification.
- Serve as a documented guide for design integrity problem reviews.

The most common root cause analysis methods cover topics from events and causal factor analysis to change analysis, barrier analysis, management oversight and risk assessment, human performance evaluation, standard problem solving and basic decision-making.

These methods are considered in the *common root cause analysis* approach developed by the Office of Nuclear Energy, US Department of Energy in their DOE guideline DOE-NE-STD-1004-92, and 'Root cause analysis: guidance document' (DOE-NE-STD-1004-92. 1992).

Common Root Cause Analysis Methods

- *Events and causal factor analysis* identifies the time sequence of a series of tasks and/or actions and the surrounding conditions that can lead to a failure occurrence. The results are displayed in an events and causal factor chart that gives a picture of the relationships of the events and causal factors.
- *Change analysis* is used when the problem is obscure. It is a systematic process that is generally used for a single failure occurrence and focuses on elements that change.
- *Barrier analysis* is a systematic process that can be used to identify physical, and procedural barriers or controls that should prevent the occurrence of failure.
- *Management oversight and risk tree (MORT) analysis* is used to identify inadequacies in barriers/controls, specific barrier and support functions, as well as management functions. It identifies specific factors relating to a possible failure occurrence and identifies factors that permit these factors to exist.
- *Human performance evaluation* identifies those factors that influence task performance. The focus of this analysis method is on operability, work environment, and management factors, as well as man-machine interface studies to improve performance.
- *Problem solving and decision-making* provides a systematic framework for gathering, organising and evaluating information, and applies to all phases of a possible failure occurrence investigation (Kepner et al. 1981).

By organising problem analysis results in an orderly manner as the design progresses, the time spent to find the root causes of possible problems is minimised. The method consists of using *factor trees* to guide the course of the analysis. Factor trees diagrammatically present the major areas to be considered in the various stages of an engineering design project, such as:

- Systems and equipment design.
- Manufacturing and installation.
- Process start-up and ramp-up.
- Operations and maintenance.

To conduct a root cause analysis specifically in the systems and equipment design stage, a series of charts can be developed representing those functional areas to be investigated, and the various factors to be considered when investigating the functional areas for causes of potential failure problems. These root cause factors for the systems and equipment design area include the following:

- Origin of design criteria.
- Utility inputs prior to design.
- Equipment specifications.
- Constraints on the design.
- Actual design solution and test.

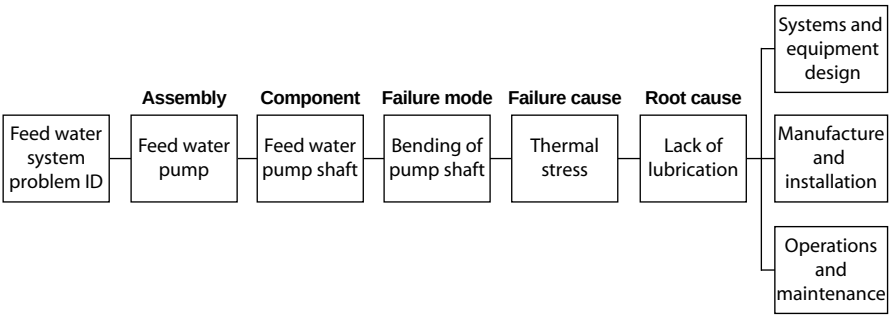


Fig. 5.6 Outage cause investigation logic tree expanded to potential root cause areas

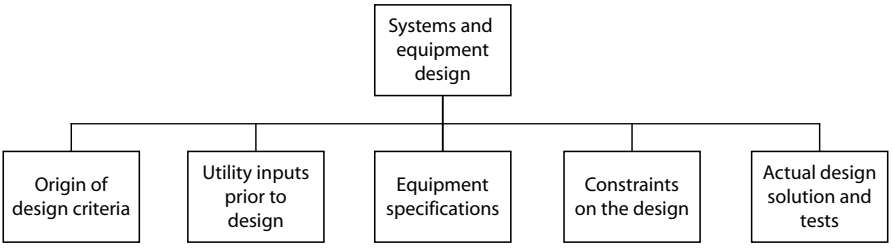


Fig. 5.7 Root cause factors for the systems and equipment design area

Each of these factors is developed into a factor tree chart indicating functional areas to be explored with the equipment's design. A thorough examination of preliminary information should eliminate the need for going through all the factor trees and all the associated questions concerning the potential root causes of design integrity problems.

In the following Figs. 5.6 and 5.7, a graphic example is given of a potential outage in a power generation unit due to root cause failure in the boiler feed water pump, expanded to the potential root cause areas of equipment design, manufacture and maintenance. Figure 5.8 gives a layout of the factor tree for the origin of design criteria.

5.2.1.3 Event Tree Analysis for Safety in Engineering Design

As indicated before, *event tree analysis (ETA)* is an inductive logic method for identifying the various accident and/or incident sequences that can generate from a single initiating event. The approach is based on the derivation of a sequence of hazardous events (accidents and incidents) that are then quantified in terms of their probability of occurrence. The events delineating these sequences are usually characterised in terms of:

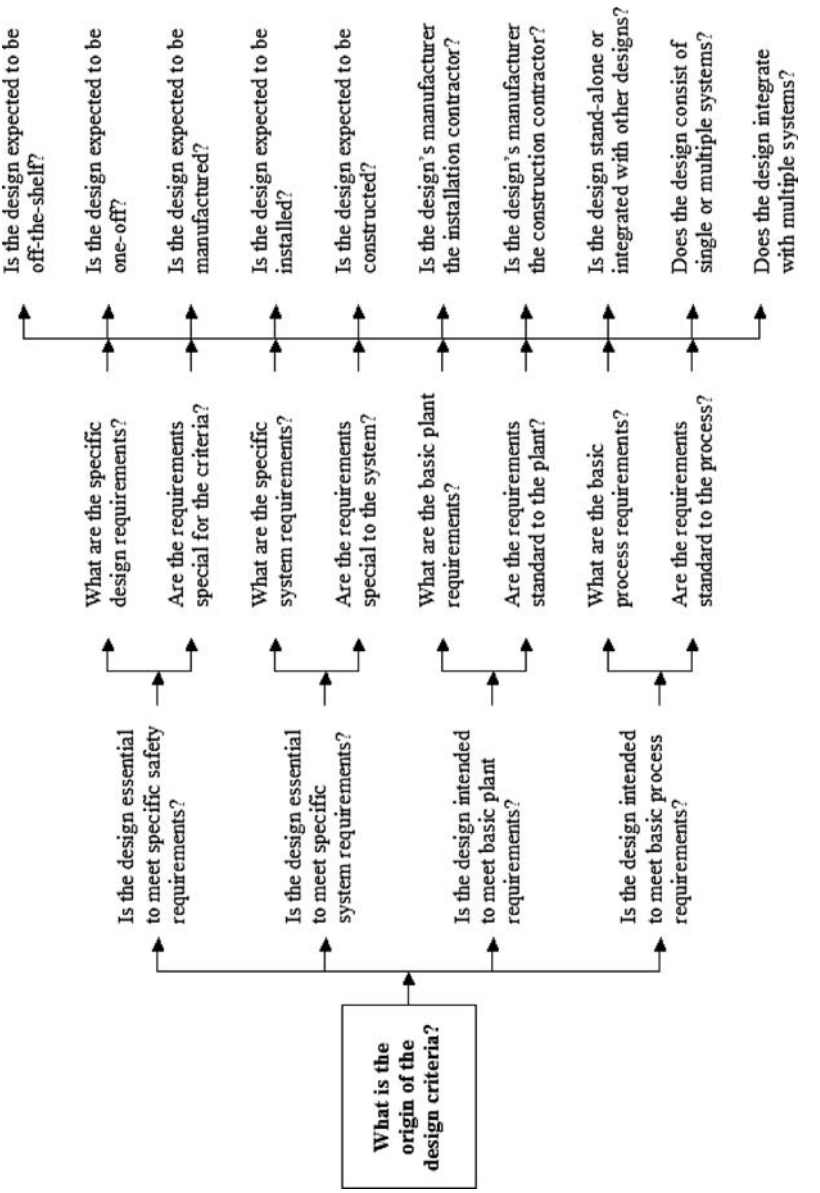


Fig. 5.8 Factor tree for origin of design criteria

- The intervention of protection systems that are supposed to take action for the mitigation of hazardous events (*system event tree*);
- The fulfilment (or not) of safety functions (*functional event tree*);
- The occurrence of physical phenomena (*phenomena event tree*).

Functional event trees are an intermediate step to the construction of *system event trees*. Following the initiating event, the safety functions that need to be fulfilled are identified; these will later be substituted by the corresponding safety and protection systems in the schematic design phase. The system event trees are used to identify the hazardous event sequences that may develop *within* the process engineering design that would require protection and safety systems. The *phenomena event trees* describe the evolution of hazardous event phenomena *outside* the process (i.e. fire, contaminant dispersion, etc.).

Event tree analysis may be qualitative, quantitative, or both, depending on the objectives of the analysis. In the application of hazards analysis, event trees may be developed independently or follow on from fault-tree analysis. An ETA is usually carried out in six steps (AIChE 1985):

1. Identification of a relevant initiating event that may give rise to unwanted consequences.
2. Identification of the safety functions that are designed to deal with the initiating event.
3. Construction of the event tree.
4. Description of the resulting hazardous event sequences.
5. Calculation of probabilities/frequencies for the identified safety consequences.
6. Compilation and presentation of the results from the analysis.

Step 1. Identification of a Relevant Initiating Event

An event tree begins with a defined hazardous event (accident and/or incident), termed the *initiating event*, the preciseness of the definition being essential for further analysis. The initiating event may be an internal or external failure, or even human error, and may have been identified by other risk analysis techniques like preliminary hazard analysis (PHA) or HAZID. To be of interest for further analysis, the initiating event must give rise to a number of safety consequence sequences. If the initiating event gives rise to only one consequence sequence, then fault-tree analysis is a more suitable technique to analyse the problem. The initiating event is often identified and anticipated as a possible critical event already in the early schematic design phase. In such cases, barriers and safety functions have usually been introduced to deal with the event.

Initiating events may be defined slightly different. For example, in the safety analysis of the cooling water system of an oxidation reactor, ‘loss of cooling water to the reactor’ may be chosen as a relevant initiating event. Alternatively, ‘rupture of cooling water pipeline’ may be chosen as the initiating event. Both of these are equally correct. It therefore follows that there is one event tree for each initiating

event considered. This aspect obviously poses a limitation on the number of initiating events that can be analysed in detail. Thus, one of the initial activities of event tree analysis is to group similar initiating events. Initiating events that are grouped in the same class usually have similar characteristics that lead to similar consequences and warrant the same safety functions. Only one typical initiating event for each class is investigated in detail.

Step 2. Identification of the Safety Functions

Once an initiating event is defined, all the safety functions that are required to mitigate the hazardous event must be defined and organised according to their time of intervention. The safety functions (safety systems, procedures, operator actions, etc.) that respond to the initiating event may be thought of as the system's defence against the occurrence of the initiating event. All safety functions that have an impact on the safety consequences of an initiating event must be identified in the sequence in which they are assumed to be activated. For each safety function, the set of possible success and failure states must be defined and enumerated, each state giving rise to a branching of the event tree.

The safety functions are classified in the following groups (AIChE 1985):

- Safety systems that automatically respond to the initiating event (e.g. automatic shutdown systems).
- Alarms that alert the operator(s) when the initiating event occurs (e.g. fire alarm systems).
- Operator procedures following an alarm.
- Barriers or containment methods intended to limit the effects of the initiating event.

The possible event chains, and sometimes also the safety functions, will be affected by various hazard-contributing factors (events or states) such as:

- Ignition or no ignition of a gas release.
- Explosion or no explosion.
- Time of the day.
- Wind direction.
- Meteorological conditions.
- Liquid/gas release containment.

Step 3. Construction of the Event Tree

The event tree displays the chronological development of event chains, starting with the initiating event and proceeding through successes and/or failures of the safety functions that respond to the initiating event. The safety consequences are clearly defined events that result from the initiating event. The diagram is usually drawn

from left to right, starting from the initiating event. Each safety function or hazard-contributing factor is called a node in the event tree, and is formulated either as an event description or as a question, usually with two possible outcomes ('true' or 'false'—'yes' or 'no'). At each node, the tree splits into two branches, an upper branch signifying that the event description in the box above that node is 'true', and a lower branch signifying that it is 'false'. The outputs from one event lead to other events. The development is continued to the resulting safety consequences.

For example, if the initiating event is the explosion of a process environment impregnated with flammable dust, coupled with the possible sparking of fire, the first function required would be that of quenching the fire with the appropriately installed sprinkler system and, finally, the setting off of a fire alarm. Following the initiating event 'explosion', fire may or may not break out. A sprinkler system and an alarm system have been installed that may or may not function. The quantitative analysis of the event tree is considered later. The functions are structured in the form of headings in the functional event tree, as shown in Fig. 5.9.

In Fig. 5.9, the calculation of the frequencies for the identified safety consequences are:

$$\begin{aligned}
\text{Fire control, with alarm} &= 10^{-2}/\text{year} \times 0.80 \times 0.99 \times 0.999 = 7.9 \times 10^{-3} \\
\text{Fire control, no alarm} &= 10^{-2}/\text{year} \times 0.80 \times 0.99 \times 0.001 = 7.9 \times 10^{-6} \\
\text{No control, with alarm} &= 10^{-2}/\text{year} \times 0.80 \times 0.01 \times 0.999 = 8.0 \times 10^{-5} \\
\text{No control, no alarm} &= 10^{-2}/\text{year} \times 0.80 \times 0.01 \times 0.001 = 8.0 \times 10^{-8} \\
\text{No fire} &= 10^{-2}/\text{year} \times 0.20 \qquad\qquad = 2.0 \times 10^{-3}
\end{aligned}$$

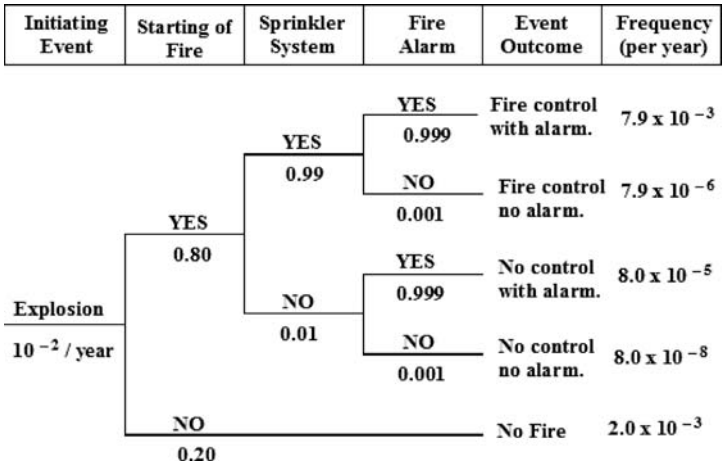


Fig. 5.9 Event tree for a dust explosion (IEC 60300-3-9)

Step 4. Description of Resulting Hazardous Event Sequences

The next step in the qualitative part of the analysis is to describe the different event sequences arising from the initiating event. One or more of the sequences may represent a safe recovery and a return to normal operation or an orderly shutdown. The sequences of importance, from a safety point of view, are those that result in accidents. The structure of the event tree diagram, clearly showing the progression of events relating to the accident, helps in specifying where additional procedures or safety systems will be most effective in protecting against these accidents. The resulting safety consequences must be described in a clear and unambiguous way. Once the safety consequences have been described, a criticality analysis is conducted, and the safety consequences ranked according to their criticality. Such a criticality ranking is based on the *risk* of the safety consequence, in terms of its *severity* and *probability of occurrence*. This is considered later in Sect. 5.2.1.6. Sometimes, it is beneficial to split the end safety consequences (outcomes) of the event tree, such as the assessment of 'estimated disabling injury frequency' and 'estimated reportable hazard frequency', into different categories of safety consequences, namely:

- *Life risk*—when the occurrence of critical functional failures can be expected to result in a risk of loss of life every time.
- *Loss risk*—when the occurrence of critical functional failures can be expected to result in a risk of loss of limb every time.
- *Health risk*—when the occurrence of critical functional failures can be expected to result in the risk of a health hazard every time.
- *People risk*—when the occurrence of critical functional failures can be expected to result in the risk of an accident affecting people working in the area every time.
- *Environment risk*—when the occurrence of critical functional failures can be expected to result in the risk of an accident affecting the environment every time.
- *Process risk*—when the occurrence of critical functional failures can be expected to result in the risk of an accident affecting the production process every time.
- *Product risk*—when the occurrence of critical functional failures can be expected to result in the risk of an accident affecting the related product every time.

In the example event tree for a dust explosion shown in Fig. 5.9, the following simplified categories are used:

- loss of lives
- environmental damage
- material damage.

The safety consequences may be ranked within each of these simplified categories. For the categories 'environmental damage' and 'material damage', typical sub-categories are used such as N (negligible), L (low), M (medium), and H (high). What is meant by these categories has to be defined in each particular case. If the safety consequences cannot be placed into a single group, a probability distribution may be given for various sub-categories, such as for the category 'loss of lives'. Thus, for this category, the sub-categories 0, 1–2, 3–5, 6–20, etc. are proposed. The

outcome of an event chain may be, for example, that 0 persons would be killed with probability 50%, 1–2 persons may be killed with probability 40%, 3–5 persons may be killed with probability 10%, and 6–20 persons may be killed with probability 2%. If, in addition, the frequency of the outcome can be estimated, then the *fatal accident rate (FAR)* associated to the specified initiating event can be calculated (Rausand 1999).

Quantitative Assessment of the Event Tree

If sufficient information is available for the initiating event and all the relevant safety functions and hazard-contributing factors, a quantitative analysis of the event tree may be carried out to give frequencies or probabilities of the resulting consequences. The probability of occurrence of the initiating event is usually modelled according to a homogeneous Poisson distribution, and a frequency that is measured as the expected number of occurrences per year (or a time unit). For each safety function, the conditional probability that it will function properly when the previous events in the event chain have occurred must be estimated. Some safety functions, like emergency shutdown (ESD) systems on offshore oil/gas platforms, may be very complex and will require a detailed analysis for the integrity of their design. The conditional reliability of a safety function will depend on a wide range of environmental and operational factors, such as stress-strength loads from previous events in the event chain, time since the last function test, etc.

In many cases, it will also be difficult to distinguish between 'functioning' and 'non-functioning'. For example, a fire pump may promptly start but stop prematurely before the fire is extinguished. The reliability assessment of a safety function may in most cases be performed by a fault-tree analysis or by an analysis based on a reliability block diagram. If the analysis is computerised, a link may be established between the reliability assessment and the appropriate node in the event tree, to facilitate automatic updating of the outcome frequencies and for sensitivity analysis. It may be relevant to study the effect on the outcome frequencies by changing the testing interval of a safety valve, for example. Graphically, the link may be visualised by a transfer symbol on one of the output branches from the node.

The probabilities of the various hazard-contributing factors (events/states) that enter into the event tree must also be estimated for the relevant contexts. Some of these factors will be independent of the previous events in the event chain, while others are not. However, most of the probabilities in the event tree are *conditional probabilities*. The probability that the sprinkler system in Fig. 5.9 will function after the initiating event is not equivalent to the probability that it will function on the basis of pilot tests under normal conditions. The possibility that the sprinkler system may have been damaged during the dust explosion and the first phase of the fire (i.e. before it is activated) must also be taken into account.

Considering the event tree in Fig. 5.9, let f_A denote the frequency of the initiating event A, 'explosion'. In this example, f_A is assumed to be equal to 10^{-2} per year, which means that an explosion will occur on average once every 100 years. Let B

denote the event 'start of a fire', and let $P(\mathrm{B}|\mathrm{A}) = 0.8$ be the conditional probability of this event when a dust explosion has already occurred. In the same way, let C denote the event 'sprinkler system on', following the dust explosion and outbreak of a fire. The conditional probability of C that the sprinkler system will function is $P(\mathrm{C}|\mathrm{BA}) = 0.99$. The event 'fire alarm activated' is denoted by D with probability $P(\mathrm{D}|\mathrm{BA}) = 0.999$.

In this example, the probability that the alarm will be activated by the event 'start of a fire' is assumed to be the same whether the sprinkler system is functioning or not. In most cases, however, the probability of this event would depend on the outcome of the previous event. Thus, let b, c and d denote the negation (non-occurrence) of the events B, C and D respectively, where $P(\mathrm{b}|xy)$ is equal to $1 - P(\mathrm{B}|xy)$, etc.

The frequencies (per year) of the end consequences may now be calculated as follows:

1. 'Fire control, with alarm'

$$f_{\mathrm{A}} \times P(\mathrm{B}|\mathrm{A}) \times P(\mathrm{C}|\mathrm{BA}) \times P(\mathrm{D}|\mathrm{BA}) = 10^{-2}/\mathrm{year} \times 0.80 \times 0.99 \times 0.999 = 7.9 \times 10^{-3}$$

2. 'Fire control, no alarm'

$$f_{\mathrm{A}} \times P(\mathrm{B}|\mathrm{A}) \times P(\mathrm{C}|\mathrm{BA}) \times P(\mathrm{d}|\mathrm{BA}) = 10^{-2}/\mathrm{year} \times 0.80 \times 0.99 \times 0.001 = 7.9 \times 10^{-6}$$

3. 'No control, with alarm'

$$f_{\mathrm{A}} \times P(\mathrm{B}|\mathrm{A}) \times P(\mathrm{c}|\mathrm{BA}) \times P(\mathrm{D}|\mathrm{BA}) = 10^{-2}/\mathrm{year} \times 0.80 \times 0.01 \times 0.999 = 8.0 \times 10^{-5}$$

4. 'No control, no alarm'

$$f_{\mathrm{A}} \times P(\mathrm{B}|\mathrm{A}) \times P(\mathrm{c}|\mathrm{BA}) \times P(\mathrm{d}|\mathrm{BA}) = 10^{-2}/\mathrm{year} \times 0.80 \times 0.01 \times 0.001 = 8.0 \times 10^{-8}$$

5. 'No fire'

$$f_{\mathrm{A}} \times P(\mathrm{B}|\mathrm{a}) = 10^{-2}/\mathrm{year} \times 0.20 = 2.0 \times 10^{-3}$$

It is evident that the frequency of a specific outcome (consequence) is simply obtained by multiplying the frequency of the initiating event by the probabilities along the event sequence leading to the outcome in question. If it is assumed that occurrences of the initiating event may be described by a homogeneous Poisson process, and that all the probabilities of the safety functions and hazard-contributing factors are constant and independent of time, then the occurrences of each outcome will also follow a homogeneous Poisson distribution.

Evaluation of the Event Tree

Once the final event tree has been constructed, the remaining task is to compute the probabilities of system failure. Each event (branch) in the tree can be interpreted as the top event of a fault tree that allows the evaluation of the probability of the occurrence of such event. The value thus computed represents the conditional probability of the occurrence of the event, given that the events preceding that sequence have occurred.

In the case of independent events, multiplication of the conditional probabilities for each branch in a sequence gives the probability of that sequence. This was illustrated in the example functional event tree for a dust explosion given in Fig. 5.9 (IEC 60300-3-9). Similarly, an illustration of independent event tree branching for a reactor safety study by the US Nuclear Regulatory Commission is given in Fig. 5.10 (NUREG 75/014 1975).

Once the system failure and success states have been properly defined, the states are then combined through the tree branching logic to obtain the various accident sequences that are associated with the given initiating event. Figure 5.10 shows a graphical example of a system event tree where the initiating event (I) is first depicted, and the system states are then connected in a stepwise, branching fashion.

System success or failure states are denoted by S_i and F_i respectively, where $i =$ the number of systems in the configuration. The accident sequences that result from the tree structure are shown in the last column. Each branch yields one particular accident sequence; for example, $(I)(S_1)(F_2)$ denotes the accident sequence in which the initiating event (I) occurs and system 1 is called upon and succeeds, (S_1), and system 2 is called upon but fails to perform its defined function, (F_2).

For larger event trees, this stepwise branching would simply be continued. The success and failure of a system must be defined under the condition that the

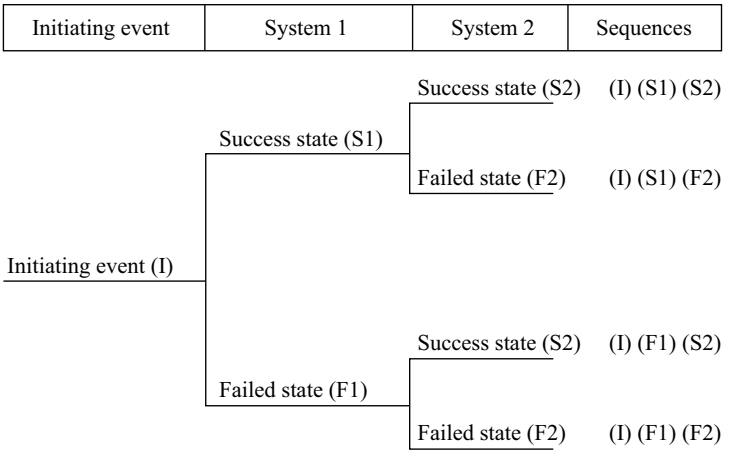


Fig. 5.10 Event tree branching for reactor safety study

initiating event has occurred. Likewise, the system states on a given branch of the event tree are conditional on the previous system states having occurred. In the case of dependent events, two approaches to accident sequence modelling are available.

The first approach is called *boundary condition event trees*, and consists of decomposing the system so as to identify the supporting parts or functions upon which some components and systems are simultaneously dependent. The functions appear explicitly as system event tree headings, preceding the dependent protection systems and components. Since dependent parts are extracted and explicitly treated as boundary conditions in the event tree, this approach leads to relatively small event trees.

For example, consider an initiating event that requires two systems, S_1 and S_2, to intervene and suppose that, to operate, S_1 needs the pumps of S_2. Then, one could *extract* the 'common part' and consider three systems: S_1, S_2^*, which is the S_2 system *without* the pumps common to S_1, and S_3, which includes the pumps used by both S_1 and S_2. It is obvious that S_3 is logically placed before S_1 and S_2 in the event tree, as schematically shown in Fig. 5.11, because it is the function that first responds to the initiating event because, to operate, S_1 needs the pumps of S_2.

The dependencies are then explicitly represented in the tree, and the branching associated with S_1 and S_2^* may be eliminated when S_3 is not functioning. Thus, all the conditional probabilities are made independent, and the probability of the accident sequences can be computed by simple multiplication. This approach considerably simplifies the computations but it requires a great deal of expertise by the analyst. In fact, since system interactions and dependencies are treated primarily within the inductive logic of the event tree, those dependencies not recognised by the analyst may not be incorporated into the analysis.

The second approach is called *fault-tree linking*. In this method, the dependencies from support systems or common parts are modelled in fault trees and thus, at the level of the event trees, the systems are inserted without the need to consider their

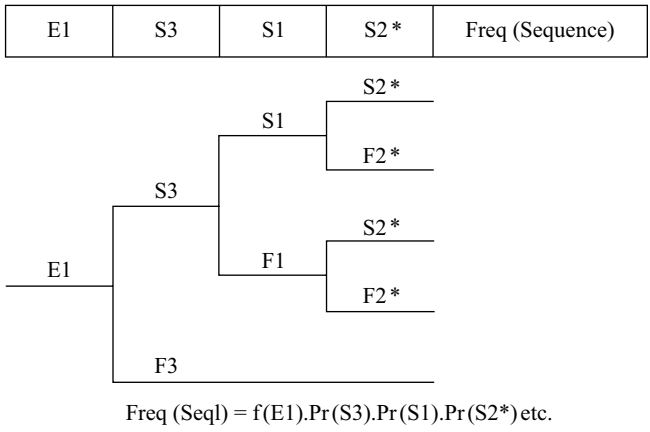


Fig. 5.11 Event tree with boundary conditions

structural dependencies. For each sequence of the event tree, the fault trees of the composing events are linked in one large fault tree that follows the logic depicted in the event tree, and the fault tree is then solved with the usual techniques to compute the probability of occurrence of that sequence.

Figure 5.12 shows the previous example of an initiating event that requires two systems, S_1 and S_2, to intervene, where both systems are explicit on the event tree without care to their dependence. The hazardous event (accident and/or incident) sequences in Fig. 5.12 may now be calculated using Bayes' theorem of *conditional probability*:

$$\begin{aligned}(I)(S_1)(S_2) &= P(S_2|S_1I)P(S_1|I)P(I)\\ (I)(S_1)(F_2) &= P(F_2|S_1I)P(S_1|I)P(I)\\ (I)(F_1)(S_2) &= P(S_2|F_1I)P(F_1|I)P(I)\\ (I)(F_1)(F_2) &= P(F_2|F_1I)P(F_1|I)P(I)\end{aligned} \tag{5.4}$$

If the probability of the sequence $(I)(S_1)(S_2)$ is to be evaluated, a fault tree is developed with the top event occurring when the initiating event I, and the failure of both systems S_1 and S_2 occur. In place of the events S_1 and S_2, the corresponding system fault trees can be substituted, thus obtaining a large fault tree that can be logically simplified (accounting for the existing dependencies) and evaluated so as to give the probability of the top event, i.e. the probability of the sequence of interest. With this method, the dependencies are properly treated even if the analysis had, a priori, no information that the dependency existed. This is particularly useful in evaluating systems for safety critical consequences during the engineering design stage when information concerning the dependencies of hazardous events is still

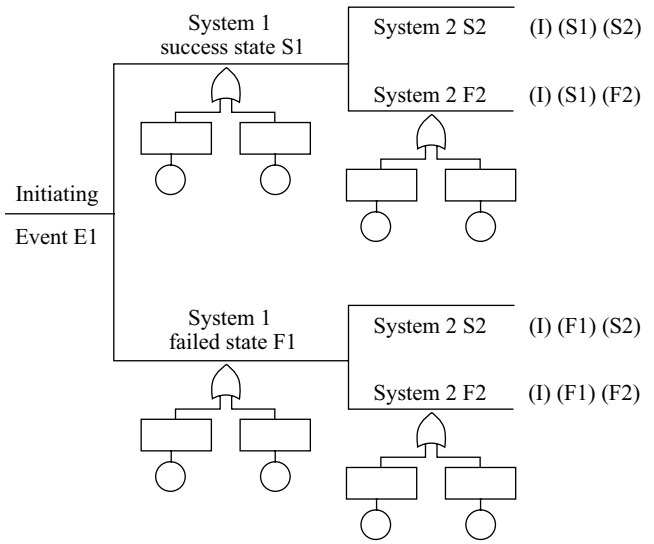


Fig. 5.12 Event tree with fault-tree linking

vague. Conversely, the resulting fault tree for an accident sequence may be rather large, necessitating more time for safety analysis during the design stage.

In summary, all the significant dependencies of hazardous events among systems are explicitly represented in the event trees with boundary conditions. The fault trees for the individual events are then simple and independent. However, great care must be taken in identifying all the existing dependencies. In the fault-tree link approach, dependencies of hazardous events are included in the fault trees for the various systems and, thus, are not dependent. The accident sequence in the linked fault tree is rather large and complex but all dependencies are treated automatically.

In Fig. 5.13, a simplified version of a functional event tree is illustrated for the case of a pipe rupture in the primary cooling circuit of a nuclear reactor. It is evident from these simplified event trees that for realistic systems, event tree analysis and, thus, safety analysis in engineering design can become quite complicated.

5.2.1.4 Cause-Consequence Analysis for Safety in Engineering Design

The *cause-consequence analysis (CCA)* method or, alternatively, the cause-consequence diagram (CCD) method is a tool for system safety and risk analysis. As with the fault-tree analysis method, the cause-consequence diagram documents the failure logic of the system. In addition to this, the cause-consequence diagram produces the exact failure probability in an efficient calculation procedure. The cause-consequence diagram technique, as applied to static systems, has been shown to yield the same result as those produced by the solution of the equivalent fault tree and binary decision diagram. On this basis, general rules have been devised for the construction of a cause-consequence diagram, given a static system. The use of the method in this manner has significant implications in terms of efficiency of conducting safety analysis, and can be shown to have benefits for determining safety in engineering design.

Safety analysis of industrial systems is carried out to reduce the risk of adverse events such as injury or death, as well as to aid in the protection of systems and facilities, by reducing the frequency or consequences of accidents and/or incidents. Since the early 1960s, various mathematical models have been used to perform reliability analysis in order to predict the likelihood that a system will function under a given demand. Each analysis model had different features that made it more appropriate to specific types of systems, and the most efficient analysis was to utilise the simplest technique. The most commonly employed technique to assess the probability of failure of industrial systems is *fault-tree analysis (FTA)*.

For systems containing independent failure events, it has been shown that the FTA technique produces a logical description of the failure process and yields, among other results, the system's unreliability. It has been highlighted, however, that this technique has limitations even when it is applied to systems containing independent failure events, in that the structural extent of backward analysis for this tree-based deductive method quickly becomes multi-branched for complex systems, and in itself becomes complex. Qualitatively, if the fault tree is complex, then finding the

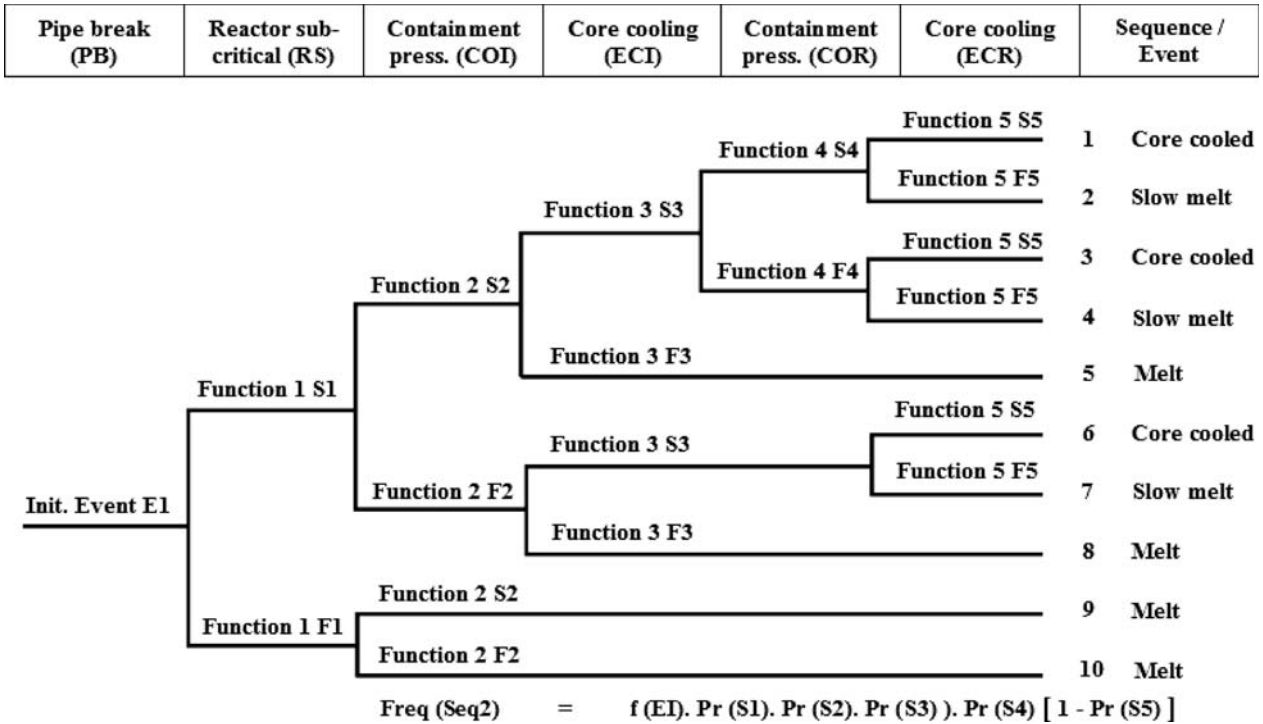


Fig. 5.13 Function event tree for loss of coolant accident in nuclear reactor (NUREG 75/014 1975)

minimal cut sets can be time-intensive. In addition, the top event probability, found via the inclusion-exclusion formula, may also be computationally time-consuming if the system contains a moderate number of minimal cut sets.

In the past, this problem was solved by using a simple approximation for the probability of occurrence of the top event. These approximations, however, can be inaccurate if the likelihood of component failure is large. The problem of inaccuracies due to approximation techniques has been alleviated by the development of the *binary decision diagram (BDD)* approach. BDDs are based on Bryant's trees (Bryant 1986) to obtain the exact top event probability efficiently by expressing the system failure modes as disjoint paths. The calculation of the top event probability is achieved by summing the probabilities of these disjoint paths. This analysis procedure makes the BDD technique more efficient than the traditional FTA technique. The BDDs, however, cannot be constructed from the system description, and are developed from the fault-tree representation of the system. During the conversion process, the BDD loses all the causality information that is represented in the fault-tree structure. In addition to this, an inefficient ordering of the basic events can result in an excessively large diagram that can prove difficult to analyse, reducing the efficiency of the method.

A technique has been developed that represents all system outcomes, given an initial event, on a diagram that contains a full textual description of the systems behaviour, and produces an exact quantification of system failure probability. This technique is based on the cause-consequence diagram (CCD) method developed at RISO Laboratories in Denmark in the 1970s to aid in reliability analysis of nuclear power plant (Villemeur 1991).

The cause-consequence diagram method involves the identification of the potential modes of failure of individual components and then relates the causes to the ultimate consequences for the system. The consequences evaluated include those that represent system failure as well as those that represent other systems behaviour. As all consequence sequences are investigated, the method can assist in identifying system outcomes that may not have been envisaged during the earlier design phases.

Cause-consequence analysis (CCA) is most frequently applied to systems where the system state changes with time (Nielsen et al. 1975). Application of cause-consequence analysis to a static system, and development of rules for the construction of a cause-consequence diagram representing a static system have been used in a high-integrity protection system (HIPS) to prevent the passage of a high-pressure surge in downstream vessels in a process engineering design (Ridley et al. 1996).

The Cause-Consequence Diagram Method

Cause-consequence diagramming is a technique that embodies both causal and consequence analysis. The technique provides a diagrammatic notation for expressing the potential consequences of an event (normally, a hazard) and the factors that influence the outcome. The basic notation is introduced in the context of the example in Fig. 5.14. In this diagram, the hazard is 'ignition'. The final outcomes (or so-called

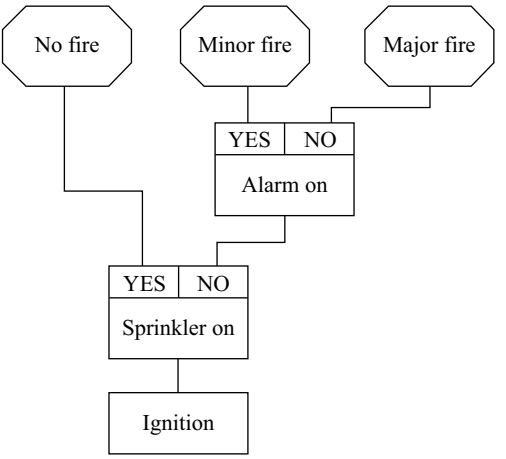


Fig. 5.14 Example cause-consequence diagram

significant consequences) are shown in octagons and vary from 'no fire', 'minor fire', to 'major fire'. The main factors that influence the outcomes are shown in 'condition vertices' (i.e. YES or NO branching), specifically 'alarm on' and 'sprinkler on'. The diagram shows that a major fire will occur as a result of the ignition hazard only if both the sprinkler and alarm system fail. If the frequency with which the hazard will occur can be estimated, and the probability that the sprinkler and alarm systems will fail on demand (and, importantly, to what degree these failures are correlated), then the frequency with which the hazard will give rise to this incident can be estimated. This is an essential step on the way to estimating the risk arising from the hazard.

Symbols Used for a Cause-Consequence Diagram

There are basically six types of symbols used for constructing a cause-consequence diagram. These symbols include the decision box, fault-tree arrow, initiator triangle, time delay box, OR gate, and consequence box, as illustrated in Table 5.4.

The *cause-consequence diagram* is thus developed from an initiating event, i.e. an event that starts a particular operational sequence, or an event that activates certain safety systems. The cause-consequence diagram is comprised of two conventional safety analysis techniques, the fault-tree analysis (FTA) method and the event tree analysis (ETA) method.

The *event tree analysis method* is used to identify the various paths that the system could take, following the initiating event, depending on whether certain subsystems/components function correctly or not.

The *fault-tree analysis method* is used to describe the failure causes of the subsystems considered in the event tree part of the diagram. This relationship is shown in Fig. 5.15.

Table 5.4 Cause-consequence diagram symbols and functions

SYMBOL	FUNCTION
q_i — YES | NO / Sprinkler on	The decision box represents the functionality of a component/system. The NO box represents failure to perform correctly, the probability of which is obtained via a fault tree or single component failure probability q_i
Ft1	Fault tree arrow represents the number of the fault tree structure which corresponds to the decision box
λ=	The initiator triangle represents the initiating event for a sequence where λ indicates the rate of occurrence
t = x hrs	Time delay 1 indicates that the time starts from the time at which the delay symbol is entered and continues up to the end of the time interval in the delay symbol
	OR gate symbol: Used to simplify the cause-consequence diagram when more than one decision box enters the same decision box or consequence box
	Consequence box represents the outcome event due to a particular sequence of events

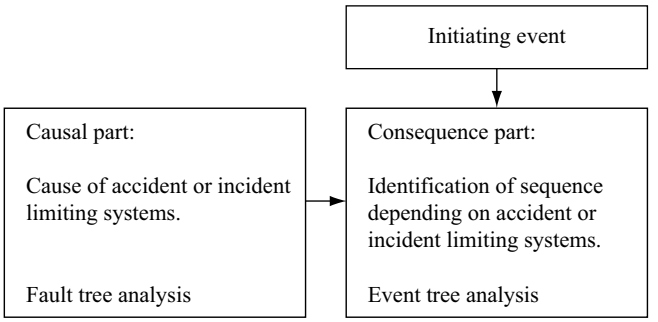


Fig. 5.15 Structure of the cause-consequence diagram

Rules for construction and quantification The cause-consequence diagram technique has been applied to a static safety system and found to yield results similar to those produced by a conventional fault tree (Ridley et al. 1996). On the basis of this study, general rules have been devised for the correct construction of the cause-consequence diagram, as given below. The use of the cause-consequence method in this manner has significant implications in terms of efficiency of reliability analysis, and can be shown to have computational benefits for analysing static safety systems.

Step 1. Component failure event ordering If the order of failure is irrelevant, which is typically the case in a static system, then the CCD can be initiated by considering *any* of the components in the system. The analysis of the CCD should yield identical results regardless of the component or variable ordering; however, the actual diagrams may vary in size. The first step of CCD construction is therefore deciding on the order in which component failure events are to be taken. To ensure a logical development of the causes of the system failure mode (i.e. initiating event), the ordering should follow the temporal action of the system, or the system's activation for the function required.

Step 2. Cause-consequence diagram construction The second stage involves the actual construction of the CCD. Starting from the initiating component, the functionality of each component or sub-system is investigated and the consequences of these sequences determined. If the decision box is governed by a sub-system, then the probability of failure will be obtained via a fault-tree diagram.

Step 3. Reduction If any decision boxes are deemed irrelevant (for example, the boxes attached to the NO and YES branches are identical, and their outcomes and consequences are the same), then these should be removed and the diagram reduced to a minimal form. Removal of these boxes will in no way affect the end result. This is illustrated in Fig. 5.16 where failure (F) can occur due to either of the two paths that terminate in the same failure function consequence, affecting either the NO or YES branches of component A.

On one path, the component (A) works, on the other it fails, proving that the state of component (A) represented by the decision box is irrelevant. When a redundant

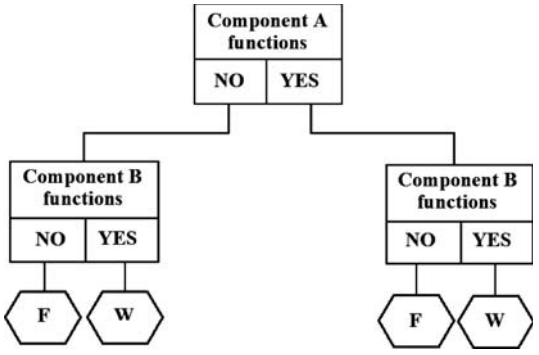


Fig. 5.16 Redundant decision box

decision box is identified, reduction is achieved by removing the box and replacing it with the next decision/consequence box. When no further redundancies exist, the cause-consequence diagram is deemed minimal.

Step 4. System failure quantification The probability of each consequence for a static system is determined by summing the probability of each set of events that lead to this particular outcome. Each sequence probability is obtained by simply multiplying the probabilities of the component events represented by the branch. This is possible because each sequence of events is mutually exclusive, and the probability of a component failure event is assumed independent.

Three-component systems The cause-consequence diagram approach for static systems can be demonstrated by a very simple system example. The approach shows that it has potential advantages in comparison to a conventional fault-tree analysis for larger systems. The system example contains three components A, B and C, and system failure is caused by either A and B failing together, or C failing alone. The system failure causes are illustrated as a fault-tree structure in Fig. 5.17.

The *cause-consequence diagram* can be constructed according to the following steps:

Step 1. Component failure event ordering The ordering chosen is that of A, B and C.

Step 2. Cause-consequence diagram construction The CCD is constructed by inspecting the failures of the components in that order (refer to Fig. 5.18).

Step 3. Reduction Boxes 3 and 4 are both irrelevant and are therefore removed. This process reduces the CCD, the final form being illustrated in Fig. 5.19 and, as no further redundancies exist, the diagram is minimal.

Step 4. System failure quantification The probability of system failure is equal to the sum of the probability of the three sequence paths that lead to the conse-

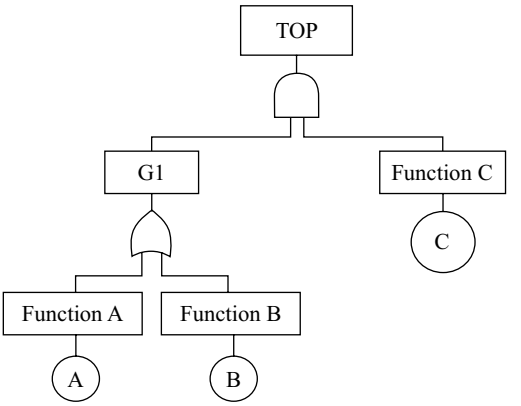


Fig. 5.17 Example fault tree indicating system failure causes

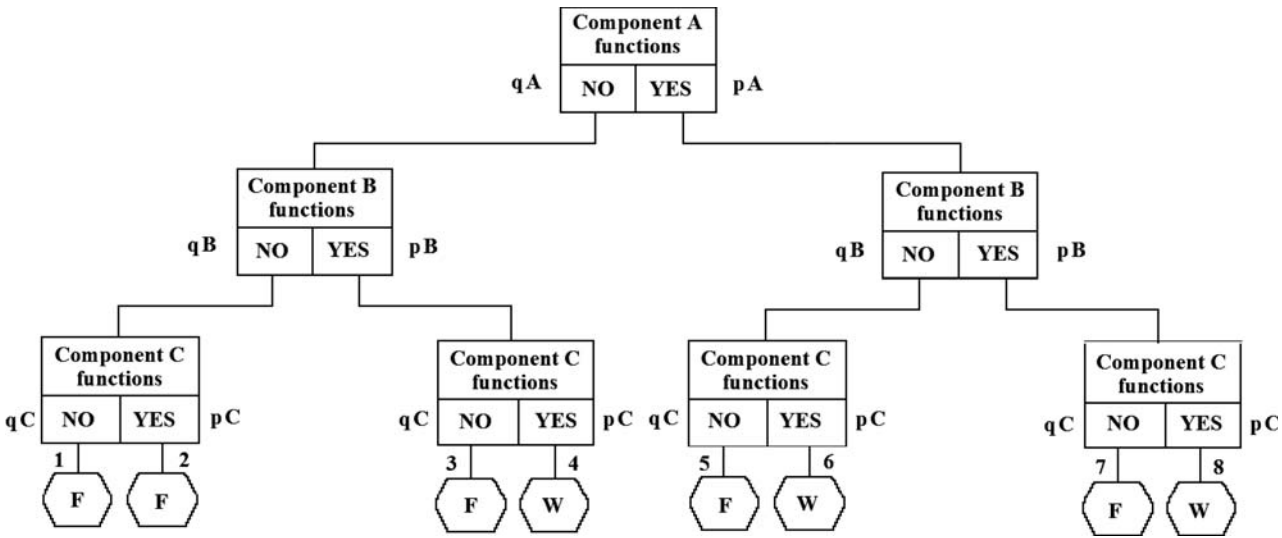


Fig. 5.18 Cause-consequence diagram for a three-component system

quence 'F'. Therefore, since the paths are mutually exclusive:

$$\begin{aligned}\text{Probability of failure} &= P(\text{path 1}) + P(\text{path 2}) + P(\text{path 4}) \\ &= q_A \cdot q_B + q_A \cdot (1 - q_B) \cdot q_C + (1 - q_A) \cdot q_C \\ &= q_A \cdot q_B + q_A \cdot q_C - q_A \cdot q_B \cdot q_C + q_C - q_A \cdot q_C \\ &= q_A \cdot q_B + q_C - q_A \cdot q_B \cdot q_C\end{aligned}$$

The fault-tree quantification calculates the top event probability to be identical to that obtained by the cause-consequence diagram approach. By studying the reduced form of the CCD, it can be noted that it is equivalent to the binary decision diagram (BDD) for the fault tree in Fig. 5.17 with the variable ordering A < B < C, as illustrated in Fig. 5.20. The top event probability can also be obtained directly from the BDD by multiplying the probabilities down the paths that lead to the terminal 1 node.

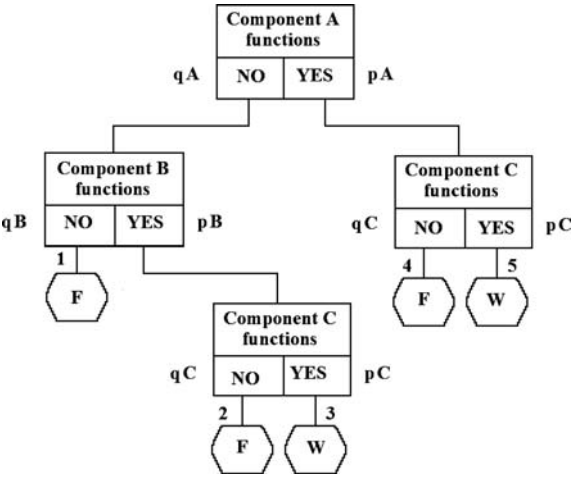


Fig. 5.19 Reduced cause-consequence diagram

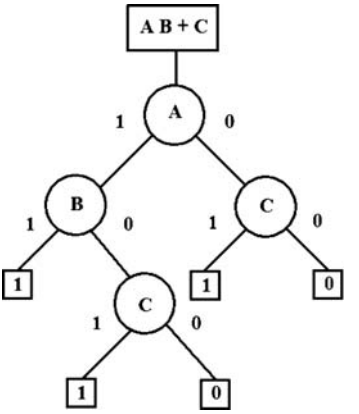


Fig. 5.20 BDD with variable ordering A < B < C

The cause-consequence diagram is reduced to a minimal form by, firstly, removing any redundant decision boxes and, secondly, manipulating any common failure events that exist on the same path. The common failure events can be extracted as common sub-modules or individual events. This process is equivalent to constructing the fault tree, converting it to a BDD, and identifying and extracting independent sub-modules. An algorithm has been developed that will produce the correct cause-consequence diagram and calculate the exact system failure probability for static systems with binary success or failure responses to the trigger event. This is achieved without having to construct the fault tree of the system and retains the documented failure logic of the system (Ridley et al. 1996)

The minimised cause-consequence diagram is then analysed using a BDD analysis procedure. Thus, exact, rather than approximate calculations are performed. The advantages of the cause-consequence diagram are:

- The diagram can be constructed directly from system description.
- Dependencies in the system can be incorporated in the analysis.
- The system is modularised to increase efficiency.
- Exact calculation procedures are adopted.

Repeated events The four-stage procedure developed to construct and analyse a cause-consequence diagram is capable of dealing with the events that occur in more than one fault-tree structure attached to the decision boxes in any sequence path. The CCD method can deal with repeated events in a more efficient way to that used for FTA (Ridley et al. 1996).

Using the CCD method, there is no need to obtain the Boolean expression of the top event and then manipulate it to produce a minimal form prior to analysis. The converse approach of the cause-consequence method deals with sequences of events that either occur (fail) or do not occur (work). The probability of a particular outcome is obtained by summation of the probabilities of all paths that lead to the outcome. Summation of the probabilities of the mutually exclusive paths in the reduced diagram yields a result similar to that obtained from the fault tree following Boolean reduction. An algorithm has been developed that can trace through a cause-consequence diagram, and identify and extract any repeated basic events in more than one fault-tree structure on the same sequence path (Bryant 1986; Ridley et al. 1996).

The procedural steps used in the extraction algorithm are the following:

1. Identify the fault-tree structures in the path under inspection.
2. Each fault tree in a path undergoes a modularisation process to identify independence. The identified independent sub-trees are then individually considered for further analysis.
3. The independent sub-trees for each fault-tree diagram are compared with one another and following, the identification of any common sub-trees or individual basic events, the cause-consequence diagram is modified.

4. The cause-consequence diagram is modified by applying the following rules:

    a. Following the identification of a common sub-tree or basic event, the common element is extracted and set as a new decision box at the highest point in the cause-consequence diagram with all dependencies below it.
    b. The cause-consequence diagram is then duplicated on each branch starting from the new decision box.
    c. Having developed a single decision box for the common sub-tree or basic event, the decision boxes that contained the common event prior to extraction require modification. The common event/s are set to 1 (TRUE) in the fault trees following the NO outlet branch from the new decision box, as this indicates failure, and 0 (FALSE) in the fault trees following the YES outlet branch to signify that the common event(s) are valid.
    d. After extraction of the common sub-tree or basic event, each fault tree that has been modified requires reorganisation. Each fault tree containing the extracted Boolean variable is inspected and the fault trees modified by setting the Boolean variable to represent the path taken in the cause-consequence diagram.
    e. The cause-consequence diagram is then reduced to a minimal form by removing any redundant decision boxes identified.

This procedure is repeated until all sequence paths have been inspected and no repeated sub-trees or basic events discovered.

For better clarity on the application of the procedural steps used in this extraction algorithm, an example of the technique is given in Sect. 5.2.4 dealing with safety and risk evaluation. The technique has been applied to a simple high-pressure protection system. The basic functions of the system are to prevent the passage of a high-pressure surge originating from upstream pumping of process material in order to protect process vessels located downstream of the surge.

5.2.1.5 Hazardous Operability Studies in Engineering Design

Hazardous operability (HAZOP) studies are based on the principle that a team approach to hazards analysis will identify more potential problems in process designs than would the combined results of individual designers of various disciplines and expertise who are working separately. The expertise is brought together during HAZOP sessions and, through a collaborative brainstorming effort, a thorough review is made of the process design under consideration.

The HAZOP study focuses on specific portions of the process called ‘nodes’. Generally, these are identified from the pipe and instruments diagram (P&ID) of the process before the study begins. A process parameter is identified (for example, *flow*), and an intention is created for the node under consideration. Then, a series of guidewords is combined with the parameter ‘flow’ to create *deviations*. For example, the guideword ‘no’ is combined with the parameter flow to give the deviation ‘no flow’. The team then focuses on listing all the credible causes of a ‘no flow’ de-

viation, beginning with the *cause* that can result in the worst possible *consequence* the team can think of at the time. Once the causes are recorded, the team lists the consequences, safeguards and any recommendations deemed appropriate. The process is repeated for the next deviation, and so on until completion of the node. The study then focuses on the next node and the process is repeated. HAZOP studies concentrate on identifying both hazards as well as operability problems. While the HAZOP study is designed to identify hazards through a systematic approach, more than 80% of the study's recommendations are operability problems, and per se not hazards. Although hazard identification is the main focus, operability problems are identified for their potential to lead to process hazards, or for their negative impact on the environment, or profitability of the engineered installation.

The definition of hazard is given as "any operation that could possibly cause a catastrophic release of toxic, flammable or explosive chemicals, or any action that could result in injury to personnel", whereas the definition of operability is given as "any operation inside the specific design under consideration that would cause a shutdown that could possibly lead to a violation of safety and health or environmental regulations, or negatively impact the profitability of the engineered installation".

a) Design Representations

A fairly wide range of design representations are in use in process engineering design and it is possible for any of these to be the basis of a HAZOP study. The use of mathematically formed representations for safety-related software systems is increasing and also these can be used for a HAZOP study. Examples of design representations include:

- block diagrams
- flow charts
- data flow diagrams
- object oriented design diagrams
- state transition diagrams
- timing diagrams
- logic diagrams
- electrical circuit diagrams.

The design representations used should cover all aspects of the system that could relate to hazards. If a single design representation does not, or cannot, cover all the relevant attributes or credible failures, then one or more other forms of representation should be used. The following issues are relevant in the decision of whether or not a further design representation is necessary (DEF STAN 00-58 2000):

- If dynamic behaviour is critical, such that hazards may result from incorrect sequencing, a representation such as a state transition diagram may be necessary.

- If the system has multiple states (such as start-up, normal operation, and shut-down), then representations of all of these should be available. Operating instructions or procedures should be included in the representation to be studied.
- If the timing of events is crucial, such that hazards could arise from timing deviations, a timing diagram is necessary.
- If, during a study, a question arises regarding the possibility of a hazard, and this cannot be answered by considering the attributes available on the design representation being studied, there is the likelihood that a further representation is necessary.

b) Entities and Their Attributes

It is the responsibility of someone familiar with the design, at the planning stage of a HAZOP study, to identify and document, for each component and interconnection on each design representation, the entities and their attributes, and also the attributes of any components to be studied. When the interconnection between two points is being studied, each type of flow should be identified as an entity in its own right, and every attribute relevant to each entity should be listed and studied, as it is common for there to be several types of data flow between two points. For example, there may be both information and control data.

c) Deviations from Design Intent

A HAZOP study may often concentrate on the interactions, and address components in detail only if an understanding of their failure modes is essential to the assessment of deviations from design intent on interconnections. If components are to be studied, then their associated attributes need to be identified. It should be noted that the term ‘components’ is used in the broadest sense and includes hardware, software, mechanical, electrical and electronic elements. The examination of components is not unique to HAZOP studies but this technique provides a systematic means of reviewing their possible failure causes and consequences. The deviations from design intent on the interactions are, however, the novel feature of HAZOP studies. Considering the interactions between components is useful as a preliminary technique if the failure modes of the components are not known at the early phases of the engineering design process, or if the failure modes are found to be very complex at the later detail design phase.

d) Guidewords and Interpretations

The principle of the use of guidewords is that, once a component or interconnection on the design representation has been selected for study, an entity on it (there may be one or more) and an attribute of the entity are chosen. A guideword is then applied to

Table 5.5 Standard interpretations for process/chemical industry guidewords

Guideword	Standard interpretation in process/chemical industry
No	No part of the design intention is achieved
More	A quantitative increase
Less	A quantitative decrease
As well as	All design intent achieved but with additional results
Part of	Only some of the intention is achieved
Reverse	Reverse flow in pipes and reverse chemical reactions
Other than	A result other than the original intention is achieved

the attribute. For example, if the guideword 'more' is applied to the attribute 'value', it may generate the questions 'what are the possible causes of the value of this entity being greater than the design intent?' or 'what are the consequences?'.

Inquiries are made into these questions and the results recorded. This process is repeated for each guideword in turn, and the whole process is then carried out for each other attribute of the entity being studied. Typical guidewords used in HAZOP studies are:

no, *more*, *less*, *as well as*, *reverse*, *other than*.

The choice of guidewords should be considered carefully, as a guideword that is too specific may limit ideas and discussion, and one that is too general may not focus the HAZOP study efficiently. Guidewords may be interpreted differently when applied to different design representations for different types of processes, as well as at different stages of a system's life cycle. When guidewords are chosen for a HAZOP study, their interpretations should be defined, as each guideword may have more than one interpretation in the context of its application to the design representation. The guideword interpretations in Table 5.5 are normally adequate for the process engineering industry (DEF STAN 00-58 2000).

Interpretations of attribute-guideword combinations Combinations of specific guidewords and attributes, in the context of the particular design representation, need interpretation according to standard guidelines as given in Table 5.5. A matrix may be a convenient way of expressing attribute-guideword combinations. Examples in Table 5.6 provide a matrix of interpretations of the guidewords in the context of design representations and attributes appropriate to those representations.

e) Selection of Process Parameters

The selection and application of *process parameters* in HAZOP studies of process engineering designs will depend on the type of process being considered, the equipment in the process, and the process intent. The most common specific process parameters that should be considered are *flow*, *temperature*, *pressure* and, where appropriate, *level*. In almost all instances, these parameters should be evaluated for

Table 5.6 Matrix of attributes and guideword interpretations for mechanical systems

Attribute	No	More	Less	As well as	Part of	Reverse	Other than
Generic meanings	No part of the intention is achieved	Quantitative increase	Quantitative decrease	All design intent with additional results	Only some of the intent is achieved	The logical opposite of the intention	Result other than original intention
Torque	No torque appears	Higher than expected	Lower than expected	N/A	N/A	Torque is reversed	Torque is cyclic
Load	No load	Higher than expected	Lower than expected	N/A	N/A	N/A	Load is in unexpected direction
Speed	No speed	Overspeed	Underspeed	N/A	N/A	N/A	Fluctuating
Force	No force	More than expected	Less than expected	N/A	N/A	N/A	In wrong direction
Temperature	No temp.	Higher than	Lower than	N/A	N/A	N/A	N/A
Containment	Complete failure of containment	N/A	N/A	N/A	Partial loss of containment	N/A	N/A
Material	Complete failure	N/A	Less of material	Corrosion is persistent	Fatigue, failure	N/A	Creep

every node. The team's comments concerning these parameters must be documented without exception. Additionally, the node should be screened for application of the remaining specific parameters such as those given in the list below. These should be recorded only if there is a hazard or operability problem associated with the parameter. A sample set of specific process parameters includes the following:

flow, temperature, pressure, composition, phase, level, relief, instrumentation, sampling, corrosion, erosion, services, utilities, maintenance, addition, safety, reaction, inserting, purging, contamination.

Specific process parameters should be considered when evaluating each node. If a particular parameter does not change from one node to the next, then it is not necessary to repeat all of the deviations that were considered in the previous node.

Guideword-parameter combinations—exploring deviations from design intent The HAZOP study creates deviations from the engineering design intent by combining *guidewords* (no, more, less, etc.) with process *parameters*, resulting in a possible deviation from the design intent. For example, when the guideword 'no' is combined with the parameter 'flow', the deviation 'no flow' results. The design team would then list all credible causes that will result in a 'no flow' condition for the specific node. Not all guideword-parameter combinations are meaningful, as the following examples indicate:

no flow	no temperature	no pressure	no reaction
more flow	more temperature	more pressure	as well as reaction
less flow	less temperature	less pressure	part of reaction
reverse flow	–	–	other than reaction

f) The Concept of Point of Reference

When defining nodes and performing a HAZOP study on a particular node, it is useful to use the concept of *point of reference (POR)* in the evaluation of deviations.

For example, in considering a node consisting of acidified gas piping up to the inlet tank of a reverse jet scrubber vessel, if the deviation 'no flow' is applied, then a dilemma results when considering the causes of 'no flow' due to pipe rupture of the acid inlet line (with safety and environmental consequences). The term 'no flow' is ambiguous, since there is still a flow of gas to the inlet tank but no flow through the acid piping to the inlet tank of the scrubber vessel. A *POR* should, therefore, be clearly established at the time the node is defined, at the downstream terminus of the node.

g) Screening for Causes of Deviations

It is necessary to be thorough in listing causes of deviations. A deviation is considered realistic if there are sufficient causes to consider that the deviation can occur.

However, only credible causes should be listed. Team judgment is used to decide whether to include events with a very low probability of occurrence. Expert judgment is required in determining what events have a low probability of occurring, so that credible causes are not overlooked. There are three basic types of causes:

- Human error, in the form of acts of omission or commission by an operator, designer or constructor, creating a hazard that could possibly result in a release of hazardous or flammable material.
- Equipment failure, in which a mechanical, structural or operating failure results in the release of hazardous or flammable material.
- External events, in which items outside the unit being reviewed affect the operation of the unit to the extent that the release of hazardous or flammable material is possible. External events include upsets on adjacent units affecting the safe operation of the unit (or node) being studied, loss of utilities, and exposure from weather and seismic activity.

The level of detail required in describing causes of a deviation depends on whether or not the cause occurs inside or outside the node.

For example, suppose that the inlet tank of the reverse jet scrubber includes a level controller as part of the node, where the level control valve results in a high-level condition in the closed mode. Since the valve and controller are part of the node, the causes should be stated in more detail because the valve may fail closed due to mechanical failure of the valve (internal event), or the valve may close due to loss of instrument air to the unit (external event). If the level controller was outside the node being studied, it would be sufficient to merely state 'level control valve LV-XXXX closes'. When the analysis considers the node in which the level controller is located, then more detail can be listed for the various causes.

h) Consequences and Safeguards

The primary purpose of a HAZOP study is the identification of scenarios that would lead to the release of hazardous or flammable material into the atmosphere, thereby exposing workers to injury. It is thus always necessary to determine, as exactly as possible, all *consequences* of any credible causes of a hazardous release of toxic material. This serves a twofold purpose, in that it aids in determining a risk ranking of multiple hazards, so that priority can be established in addressing the most severe hazards first; furthermore, it aids in determining whether a particular deviation results in an operability problem or hazard. If the HAZOP study team concludes from the consequences that a particular cause of a deviation results in an operability problem only, then further investigation should end in this case, and consider the next cause, deviation or node.

If the HAZOP study team determines that the cause will result in the release of hazardous or flammable material, then *safeguards* should be identified. Safeguards should be included whenever a combination of cause and consequence presents

a credible process hazard. The basis of what constitutes a safeguard can be summarised in the following criteria:

- Those systems and/or written procedures that are designed to prevent a catastrophic release of hazardous or flammable material.
- Those systems that are designed to detect and give early warning following the initiating cause of a release of hazardous or flammable material.
- Those systems and/or written procedures that mitigate the consequences of a release of hazardous or flammable material.

The HAZOP study team should use care when listing safeguards. Hazards analysis requires an evaluation of the consequences of failure of engineering and administrative controls, so a careful determination of whether or not these items can actually be considered safeguards must be made. In addition, the team should consider realistic multiple failures and simultaneous events when evaluating whether or not any of the above safeguards will actually function as such in the event of an occurrence.

i) Deriving Recommendations

Recommendations are made when the safeguards for a given hazard scenario, as judged by an assessment of the *risk* of the scenario, are inadequate to protect against the hazard. 'Action items' and 'information needs' are those recommendations that have been assigned for follow-up by one of the team members. Implementation of hazard analysis recommendations may follow the following guidelines:

- High-priority action items should be resolved within 4 months.
- Medium-priority action items should be resolved within 4–6 months.
- Lower-priority action items should be resolved following medium-priority items.

Review of all recommendations made in HAZOP studies must be made to determine relative priorities and determine a schedule of implementation. After each recommendation has been reviewed, all resolutions should be recorded in a tracking document and kept on file. Recommendations include design, operating or maintenance changes that reduce or eliminate deviations, causes and/or consequences. Recommendations identified in a hazard analysis are considered to be preliminary in nature.

5.2.1.6 Risk Analysis in Engineering Design

Risk analysis methodology used for determining the integrity of engineering design are grouped into two categories: *hazards identification* and *risk estimation*. This level of risk analysis is usually for making an assessment of equipment criticality during preliminary design through the use of a *risk priority number (RPN)* technique

(Bowles et al. 1994). Although the technique has been described in Sect. 3.2.2.5, some of the basic features are repeated here in summary.

This method prioritises risk by calculating a risk priority number for a component failure mode using three factors:

- Failure mode occurrence probability.
- Failure effect severity.
- Failure detection probability.

The risk priority number is computed by multiplying the rankings on a scale from 1 to 10 assigned to each of these three factors, and is expressed by the relationship:

$$\mathrm{RPN} = (\mathrm{OR})(\mathrm{SR})(\mathrm{DR}) \tag{5.5}$$

where:

RPN = the risk priority number
OR = the occurrence ranking
SR = the severity ranking
DR = the detection ranking.

Risk estimation, as adopted by the European Community (EC 1996) for use in risk assessment, is defined in the following format:

> *Risk, related to an identified hazard, is a function of the probability of its occurrence with respect to the frequency and duration of exposure to the hazard, and the means of avoiding it, and the severity of the accident or incident that can result from the hazard.*

Thus, risk can be quantified as the product of the level of severity of the risk (i.e. disaster or loss), with its probability of occurrence (i.e. chance).

This can be formulated as the following:

$$\mathrm{Risk} = \mathrm{Severity} \times \mathrm{Probability} \tag{5.6}$$

From the definition, severity is the disaster or loss incurred. The measure of severity can be quantified in two events: accidents and incidents. The measure of probability can be quantified in the form of appropriate statistical probability distributions or measures of statistical likelihood. In this regard, an accident is an undesired event that results in disastrous physical harm to a person. An incident is an undesired event that could result in a loss. In the context of safety, this loss is in the form of an asset loss, which implies damage to equipment or property. Risk is thus an indication of the *degree of safety*, determined on the basis of two considerations, the first according to design criteria, and the second according to operational performance:

- The *estimated* degree of safety. This is *assessed* according to the contribution of:
  - the 'estimated disabling injury frequency' arising from functional failure of the item,

  - the ‘estimated reportable hazard frequency’ arising from functional failure of the item,
  - the ‘estimated physical condition’ of the item related to its safety.

- The *actual* degree of safety. This is *measured* according to the contribution of:

  - the ‘actual disabling injury frequency’ arising from functional failure of the item,
  - the ‘actual reportable hazard frequency’ arising from functional failure of the item,
  - the ‘actual physical condition’ of the item related to its safety.

The assessment of ‘estimated disabling injury frequency’ considers *severity criteria* such as:

- *Life risk*—when the occurrence of critical functional failures can be expected to result in a risk of loss of life every time.
- *Loss risk*—when the occurrence of critical functional failures can be expected to result in a risk of loss of limb every time.
- *Health risk*—when the occurrence of critical functional failures is expected to result in the risk of a health hazard every time.

The assessment of ‘estimated reportable hazard frequency’ considers severity criteria such as:

- *People risk*—when the occurrence of critical functional failures can be expected to result in the risk of an accident affecting people working in the area every time.
- *Environment risk*—when the occurrence of critical functional failures can be expected to result in the risk of an accident affecting the environment every time.
- *Process risk*—when the occurrence of critical functional failures can be expected to result in the risk of an accident affecting the production process every time.
- *Product risk*—when the occurrence of critical functional failures can be expected to result in the risk of an accident affecting the related product every time.

The assessment of ‘estimated physical condition’ considers severity criteria such as:

- *Loss risk*—when the item’s physical condition can be expected to result in process losses in the system that will result in critical functional failures becoming imminent.
- *Damage risk*—when the item’s physical condition can be expected to result in physical damage to related items that will result in critical functional failures becoming imminent.
- *Defects risk*—when the item’s physical condition can be expected to result in physical defects arising in the item or its parts that will result in critical functional failures becoming imminent.

The various severity criteria described above are rated by designating a probability value from 0.1 to 1.0, for each criterion relevant to each failure mode, according to a *risk assessment scale*. The severity criteria is designated a value ranging from 10 to 1. The most severe degree of safety (disabling injury—life risk) is valued at 10, and no safety risk is valued at 1.

The probability value is assessed for different categories called 'actual', 'probable' and 'possible'. These probability values range from:

0.95 to 1.00 for the category actual
0.50 to 0.95 for the category probable
0.01 to 0.50 for the category possible.

The estimated risk is thus rated according to the *risk assessment scale* shown in Table 5.7, using the following probability qualifiers:

Actual occurrence: 0.95 to 1.00
Probable occurrence: 0.50 to 0.95
Possible occurrence: less than 0.50.

Table 5.7 Risk assessment scale

Risk assessment scale Estimated degree of safety:	Risk assessment values: Degree of severity × Probability		
Severity criteria	Actual 0.95 to 1.00	Probable 0.50 to 0.95	Possible 0.01 to 0.05
(Disabling injury)	Deg. Prob. Risk	Deg. Prob. Risk	Deg. Prob. Risk
Life risk	10	10	10
Loss risk	9	9	9
Health risk	8	8	8
(Reported accident)			
People risk	7	7	7
Process risk	6	6	6
Product risk	5	5	5
(Physical condition)			
Damage risk	4	4	4
Defects risk	3	3	3
Loss risk	2	2	2
(No safety risk)	1	1	1
Overall risk	Total	Total	Total
Overall average	Average	Average	Average

Table 5.8 Initial failure rate estimates

Qualification	Failure rate ($\times 10^{-6}$)
Very low	< 1
Low	1 to 10
Fair	10 to 100
High	100 to 1,000
Very high	$> 1{,}000$

Once an overall total and an overall average value of risk has been assessed, a *safety criticality rank* can be defined as follows:

$$\text{Criticality rank} = \text{Risk} \times \text{Failure rate} \tag{5.7}$$

If the failure rate for the item cannot been determined, qualifying values for *initial failure rate estimates* can be used (Table 5.8).

5.2.1.7 Summary of Safety and Risk Analysis in Engineering Design

Up to this point, the various conventional deductive and inductive analysis techniques for safety hazards and risk analysis have been considered without giving much attention to their specific application in each engineering design phase. Some of the more appropriate techniques that relate to the progressive phases in the engineering design process are the following:

- *Design cost risk analysis.*
  Design cost risk analysis consists of identifying independent variables relating to the system or equipment attributes such as mass, size, volume, material thickness, etc. plus the cost of ensuring the required reliability and safety relative to the selected attributes. The independent variables, also called cost drivers, are selected through statistical analysis, and form the basis of cost estimating relationships (CERs).
- *Operational risk analysis.*
  Operational risk analysis considers risk in their operating environment. As a result, it is necessary and useful to develop a safety hypothesis, expressed as a risk equation, which relates system throughput capacity to risk. Such a risk equation has its roots in financial risk management and has been expanded to measure the mean expected loss risk, which is more suitable for process systems in general. Such a measure not only quantifies risk but also clarifies system safety principles during conceptual design. Early identification of specific risk costs and safety benefits of different design alternatives enables avoidance or mitigation of hazards that could result in operational losses.

- *Operability analysis—formally, hazards and operability (HAZOP) analysis.*
  Operability analysis considers safety issues throughout an engineered installation's life cycle, from design, manufacture, installation, assembly and construction, through to start-up and operation. The later that hazardous operating modes are detected in this development process, the more serious and expensive they become to avoid or mitigate through the required plant modifications. Extensive and systematic examination of safety aspects has to be carried out carefully and at the earliest possible opportunity in the engineering design stage.
- *Point process analysis—formally, Markov chain point processes.*
  Point process analysis is intended to model a probabilistic situation that places points on a time axis. For safety analysis, these points are termed accident or incident *events*.
- *Fault-tree analysis (FTA).*
  Fault-tree analysis is the most frequently used in the assessment of safety protection systems for systems design. For potentially hazardous process engineering systems, it is required statutory practice to conduct a quantitative assessment of the safety features at the engineering design stage. The design is assessed by predicting the probability that the safety systems might fail to perform their intended task of either preventing or reducing the consequences of hazardous events.
- *Root cause analysis (RCA).*
  Root cause analysis (RCA) considers multiple failures arising from a common cause. This was first studied on a formal basis in the nuclear power industry. In order to obtain sufficiently high levels of reliability and safety in critical risk control circuits, redundancy was introduced. In applying redundancy, several items can be used in parallel with only one required to be in working order.
- *Cause-consequence analysis (CCA)—failure modes and safety effects analysis.*
  Cause-consequence analysis for safety systems design explores the system's responses to an initiating deviation from pre-determined norms (such as the limits of safe operating parameters), and enables evaluation of the probabilities of unfavourable outcomes at each of a number of mutually exclusive loss levels, depending upon the extent of deviation from these norms.
- *Hazards analysis (HAZAN)—probabilistic risk analysis.*
  Hazards analysis considers identifying potential hazards that may be caused either by the nature of the process or the intended systems configuration. A thorough safety and hazards analysis is compulsory during the engineering design and development stages, for official approval to commence with construction.

These techniques are considered in detail below, within the appropriate conceptual, preliminary or detail design phases of the engineering design process.

5.2.2 Theoretical Overview of Safety and Risk Prediction in Conceptual Design

Safety and risk prediction attempts to identify initial problems or preliminary hazards, and to estimate the risks related to the severity of their consequences and related probabilities of occurrence. Safety and risk prediction is considered in the *conceptual design* phase of the engineering design process, and includes concepts of modelling such as:

i. Cost risk models in designing for safety
ii. Process operational risk modelling
iii. Hazard and operability studies.

5.2.2.1 Cost Risk Models in Designing for Safety

Cost estimates during the early stages of engineering design are crucial. They influence the go, no-go decisions concerning the development of engineering projects. In many cases, from 70 to 80% of a design's cost is committed during the concept phase (Mileham et al. 1993).

Making a wrong decision concerning designing for reliability and safety can be extremely costly later in the development project. System modifications and process alterations become more expensive as the project progresses into manufacture, installation and construction. However, the difficulties of cost estimating at the conceptual design phase are well recognised (Meisl 1988). The two major obstacles that need to be addressed in estimating costs at the conceptual design phase are, first, working with a limited amount of available data concerning the new design and, second, identifying the requirements that determine how cost estimates are derived, including assumptions and risks. The task in overcoming these obstacles, particularly in estimating risk costs for safety in engineering design, is concerned with the choice of cost estimating methods, some of which include the following:

- Traditional cost estimating.
- Parametric estimating.
- Feature-based costing.
- Qualitative cost estimating.

a) Traditional Cost Estimating

In traditional costing, there are two main estimates: a 'first sight' or 'first round' estimate, which is done in the early design phases, and a detailed estimate, done later to calculate costs precisely. The former of these cost estimating methods is based largely on the experience of the estimator. For example, it is not uncommon for a 'first round' project estimate to be based upon a past similar project, or purely

on costing experience. Although useful for a rough order of magnitude estimate, this type of estimating is too subjective in engineering designs of large integrated systems, and more quantified and justified estimates are essential (Roy et al. 1999).

For detailed estimates, risk cost is based upon a knowledge of the cost of operations and the cost of failure repair. Typically, such a cost model would incorporate the following

$$TC = C_{\mathrm{i}} + C_{\mathrm{o}} + C_{\mathrm{r}} \tag{5.8}$$

where:

TC = total cost (safety life-cycle cost)
C_{i} = initial cost (design and manufacture)
C_{o} = operating cost
C_{r} = risk cost.

The risk cost component of this safety life-cycle (SLC) costing of a process engineering design can be expressed in terms of two cost components:

- the average cost of failure C_{f}, and
- the expected life of the system L_{t}

$$C_{\mathrm{r}} = \frac{C_{\mathrm{f}} \cdot L_{\mathrm{t}}}{\mathrm{MTBF}} \tag{5.9}$$

where:

MTBF = mean time between failures.

The risk cost component of the average cost of failure, C_{f}, can in turn also be expressed in terms of two cost components:

- the cost of failure loss, and
- the cost of failure repair

$$\begin{aligned} C_{\mathrm{f}} = {} & [C_{\mathrm{s}}(\mathrm{MTTR} + T_{\mathrm{m}}) + C_{\mathrm{l}}] \\ & + [C_{\mathrm{m}}(\mathrm{MTTR} + T_{\mathrm{m}}) + C_{\mathrm{d}} + C_{\mathrm{p}}] \end{aligned} \tag{5.10}$$

where:

T_{m} = repaired system response time
C_{s} = cost of loss of service
C_{l} = cost of incident/accident loss
C_{m} = cost of failure repair
C_{d} = cost of failure delay
C_{p} = cost of parts replacement
MTTR = mean time to repair.

The expected life of the system L_{t}, expressed as a ratio against the mean time between failures (MTBF), is in effect the expected number of failures over the life span of the system, which is a measure of the system's reliability, R. This reasoning

is based on the understanding that MTBF is a measure of the average time until the occurrence of failure.

Thus

$$R = \frac{L_t}{\text{MTBF}} \tag{5.11}$$

$$C_r = C_f \cdot R \tag{5.12}$$

Because risk cost is based upon a detailed knowledge of the cost of system operations and repair, the method is not useful during the conceptual design phase of project development. In order to estimate costs during this phase, other approaches are required.

b) Parametric Estimating

A widely used method for estimating costs at the early stages of process development is known as *parametric estimating (PE)*. Typically, for most systems in process engineering, mass relates to the cost of its manufacture. That is, as the weight of a pressure vessel increases, due to an increase in size (volume) or in thickness of material, so does the cost of manufacturing it. Furthermore, this particular relationship is often described as linear.

Using relatively simple algebra, it is possible to derive a formula to determine a mathematical relationship for cost to mass (or size). The linear equation $y = ax + b$ is used to describe the line of best fit for points representing this relationship and, once described, it is then possible to use the formula to predict the cost of other similar pressure vessels, based on their size or weight alone. Within the field of cost estimating, this relationship is known as a *cost estimating relationship (CER)*. This is a rather simplistic illustration describing the main principles of parametric estimating. As CERs become more complex, involving several variables, more complex mathematical equations are used to describe the relationships. When CERs become too complex for mathematical equations to solve, cost algorithms are developed, such as *genetic algorithms (GAs)* for determining the extent of the risk cost associated with designing for reliability and safety. An example of the use of such an algorithm is in optimising a risk cost function in the allocation of component redundancy to a safety control system (Coit et al. 1996).

Parametric estimating can be used throughout the life cycle of an engineered installation. However, it is used mainly during the early stages of development (i.e. conceptual design phase), and for *design to cost (DTC)* analyses, which is considered later. The techniques are acceptable for both military and industrial application (PCEI 1999).

However, parametric estimating does have its disadvantages—for example, CERs of many conceptual designs are too simplistic to forecast costs. Furthermore, parametric estimating is based primarily on statistical assumptions concerning cost driver relationships to cost, and estimations should not completely rely upon statistical analysis. Hypotheses based on experience, common sense and engineering

knowledge should come first, and then the relationship should be tested with statistical analysis. Most CER studies apply parametric estimating for quantitative criteria in design, but not for vague or unknown criteria requiring qualitative or expert judgment. Current research in this area has demonstrated the validity of the approach (Roy et al. 1999).

Design to cost The objective with *design to cost (DTC)* is to make the design converge to an acceptable cost, rather than to let the cost converge to design. DTC activities, during the conceptual and early design phases, are those of determining the trade-offs between cost and performance for each of the concept alternatives.

DTC can produce massive savings on risk cost before system development begins. The general approach is to set a cost goal, then allocate the goal to the elements of the design, including designing for reliability and designing for safety. The design must then be confined to the alternatives that satisfy the cost constraint (Michael et al. 1989).

However, this is only possible once a risk cost algorithm has been developed that can be used to determine the impact of these elements of the design such as designing for reliability and safety. These algorithms are used primarily to monitor the impact of design decisions on risk cost, rather than the converse, throughout the engineering design process. It is thus the cost engineers who are responsible for establishing sufficient information on cost in the early stages of systems development that will enable the design engineers to make meaningful decisions.

c) Feature-Based Costing

A relatively new form of PE is that of *feature-based costing (FBC)*. This has become popular due to the rise and sophistication of computer aided tools in engineering design. The growth of CAD/CAM technology and that of 3D modelling tools have largely influenced the development of feature-based costing. Researchers have for some time investigated the integration of design, process planning and manufacturing for costing using a feature-based modelling approach (Wierda 1991).

However, feature-based costing has not yet been fully established or developed with respect to costing safety in engineering design. Nonetheless, there are several good reasons for examining the use of features as a basis for risk costs during the early design phases where certain equipment (i.e. assemblies, sub-assemblies and components) have already been identified. Such equipment can essentially be described as a number of associated features, i.e. holes, flat faces, edges, folds, etc.

It follows that each equipment feature has cost implications, since the more features the equipment has, the more manufacturing it will require, and the greater its safety risk with respect to operational reliability, durability and robustness. Therefore, choices regarding the inclusion or omission of a feature impact the risk costs of equipment, especially process control equipment.

d) Qualitative Cost Estimating

Fuzzy logic, possibility theory and artificial neural networks present the next generation in computerising the human thought processes. Many researchers and practitioners are fast developing and investigating the use of *artificial intelligence (AI)* systems and applying these to cost estimating. For risk cost estimating purposes, the basic idea of using neural networks is to provide data to a computer so that it can computationally learn which safety attributes mostly influence the cost. This is achieved by training the system with data from past case examples with respect to the cost of losses due to hazardous failure, the estimated frequency of the initiating event, and the severity and probability of the consequences. The neural network then approximates the functional relationship between the attribute values and the risk cost. Safety attribute values such as estimate values of frequencies and/or probabilities are input to the network, which applies the approximated function obtained from the training data and computes a prospective risk cost. Relatively recent work has demonstrated that, under certain conditions, neural networks produce better costing predictions than do conventional regression costing methods. However, in cases where appropriate CERs can be identified, regression models have significant advantages in terms of accuracy, variability, model creation and model examination (Smith et al. 1997).

Artificial neural networks (ANN) require a large case base in order to be effective, which is not always the case with safety attributes of equipment in process engineering systems. In addition, the case base needs to be comprised of similar equipment in common applications, and new designs need to be of a similar nature, in order for the cost estimate to be effective. Thus, neural networks cannot cope easily with uniqueness or innovation in engineering design. With regression analysis, safety and risk issues in the design can be argued logically, and an audit trail of the development of the risk cost estimate can be established. This is because a CER equation is developed that is based on common sense and logic. In many cases, when considering neural networks, the resultant equation does not appear logical even if it was extracted by examining the weights, architecture, and nodal transfer functions that are associated with the final trained model. The artificial neural network truly becomes a 'black box' CER. This is disadvantageous if a detailed list of the reasons and assumptions behind the risk cost estimate is required. The black box CER also limits the use of risk analysis, which is a prime benefit of parametric estimating, and which will be considered now in greater detail.

e) Parametric Costing and Risk Analysis

This sub-section provides fundamental knowledge concerning the tools and techniques currently used within the area of parametric costing and risk analysis within the conceptual design phase. The method of *parametric cost estimating (PCE)* is commonly used to estimate the cost of new engineering designs. It provides a technique for predicting cost based on historical relationships between cost and one or

more predictor variables such as *cost estimating relationships (CERs)*. The method uses a statistical approach, and is commonly used for risk cost estimation during the conceptual design phase (Rush et al. 2000).

Cost Estimating Relationship (CER) Development

Cost estimating relationships (CERs) can range from simple heuristics (rules of thumb) to complex relationships involving multiple variables. The principal function of CERs is to provide equations or graphs that summarise historical cost data from which future cost estimates can be made. A general methodology for developing CERs includes activities such as data collection, testing a CER's logic, statistical analysis, CER significance tests, and validation. The collection of data is often a very critical and time-consuming activity, requiring more effort to be devoted to assembling a quality database than to any other task in the CER development process. After a database is developed, the next step is the mathematical formulation of a hypothesis and then to test the mathematical form of the CER in order to determine its logic. This involves identifying potential cost driving variables and identification of cost relationships.

In order to test and validate a CER, the statistical analysis technique of *multiple regression* is used to test the hypothesis. Although widely accepted, PCE is based on statistical *assumptions* concerning cost driver relationships to cost, particularly risk cost, and should therefore not be completely reliant upon statistical analysis but based also on experience, common sense and engineering knowledge. Because estimating is based on assumptions concerning the likely risk cost of an as yet undeveloped design, the preferred approach is to combine the statistical techniques of parametric estimating with statistical risk analysis.

The introduction of *risk cost analysis* ensures that the consequences of risks are correctly taken into account to be able to quantify risk cost early in the design stage of the life cycle of a system.

f) Risk Cost Analysis

The first step in analysing risk cost is identification of the CER variables. This is readily available from the results of the parametric cost estimating method. The risk cost consists of independent variables relating to the system or equipment attributes such as mass, size, volume, material thickness, etc. included in the CERs, plus the cost of ensuring the required reliability and safety relative to the selected attributes. The independent variables, also called cost drivers, are selected through statistical analysis, and form the basis of the CER.

The risk cost can be expressed in terms of the following principal cost components: the parametric cost estimates, and the cost of ensuring reliability and safety

$$\mathrm{RC} = C_0 + [C_1(\mathrm{mass}) + C_2(\mathrm{material})] + C_\mathrm{s} \tag{5.13}$$

where:

RC = risk cost
C_0 = initial cost constant (set to zero for cost comparisons)
C_1 = cost constant multiplied with the CER variable of mass
C_2 = cost constant multiplied with the CER variable of material
C_s = cost variable for ensuring required reliability and safety.

The cost of ensuring the required reliability and safety relative to the selected attributes can be formulated as

$$C_s = C_f R \tag{5.14}$$

where:

C_f = cost of failure relative to the selected attributes
R = risk of a failure incident occurring.

The risk of a failure incident occurring can be formulated as

$$R = p \cdot c \tag{5.15}$$

where:

p = the probability of the event occurring
c = the consequence of the risk on the estimate.

5.2.2.2 Process Operational Risk Modelling

Complex process systems, especially complex integrations of systems, increasingly have to cope with risk in their operating environment. As a result, it is necessary and useful to develop a safety hypothesis, expressed as a risk equation, which relates system throughput capacity to risk. Such a risk equation has its roots in financial risk management and has been expanded to measure the mean expected loss risk, which is more suitable for process systems in general. Such a measure not only quantifies risk but also clarifies system safety principles during conceptual design. Early identification of specific risk costs and safety benefits of different design alternatives enables avoidance or mitigation of hazards that could result in operational losses.

a) Overview of the Risk Hypothesis and Risk Equation

From Eqs. (4.23) and (4.24) in Sect. 4.2.1.2, a process system is considered to be a functional unit that converts inputs to outputs, and which may be composed of sub-systems connected either in series or in parallel, enabling the system to convert a set of process inputs, I_p, to a set of process outputs, O_p, per unit time, so that O_p is equivalent to the system throughput, T_p, where the yield is 100%.

Equation (4.23) is reviewed here as the following expression

$$\text{Process throughput } T_{\text{proc}}^{C} = \frac{\text{Material in process}}{\text{Processing time}} = \text{Rated capacity } (C_{\text{r}}) \tag{5.16}$$

The term *throughput capacity* relates engineering process throughput T_{p} to rated capacity C_{r}. If T_{p} is the maximum value for O_{p}, then T_{p} is seen as the throughput capacity of the system, measured as the units of output per unit time when the system is operating at rated capacity. In general, if the system is operating at a fraction f of throughput capacity T_{p}, due to process fluctuations, where f is an average constant (i.e. 0.95), then the *reduced* throughput, U, can be determined.

The reduced throughput, U, can be expressed as

$$U = f \times T_{\text{p}} \tag{5.17}$$

In reality, the system will be exposed to unpredictable fluctuations in throughput capacity and, over a period of time t, the mean and, thus, expected throughput capacity will be $\mathcal{T}_{\text{p}}$, where

$$\mathcal{T}_{\text{p}} = \left[\sum_{t=0}^{n} U_{\text{t}}\right] / n \tag{5.18}$$

where:

$\mathcal{T}_{\text{p}}$ = mean throughput capacity
n = number of time periods.

In real loss-deviation time periods, the actual capacity values can be expressed as the series

$$S_{T_{\text{p}}} = \{\mathcal{T}_{\text{p}} - L_1, \mathcal{T}_{\text{p}} - L_2, \ldots \mathcal{T}_{\text{p}} - L_n\} \tag{5.19}$$

where $L_1, L_2, \ldots, L_n$ are loss deviations from the average T_{p}.

The expected or average T_{p} actually rarely occurs, if at all. In reality, it is the unpredictable sequence of losses (L_1, or $L_2, \ldots, L_n$) with respect to an average or expected throughput capacity $\mathcal{T}_{\text{p}}$, in a given time period, which is used in the measure of risk of loss of throughput. Two meaningful measures of risk may be used, the traditional standard deviation measure, and a new measure, the mean expected loss that in many cases is more suitable for systems in general.

b) Risk Measures

Risk measures are statistical measures, such as the standard deviation risk (*SD-risk*) with respect to the mean throughput capacity $\mathcal{T}_{\text{p}}$; if twice the standard deviation is used, then an even stronger risk measure is obtained, the two-standard deviations risk (*2-SD-risk*) measure. A new measure more suitable for process systems in general, termed the mean expected loss risk (*MEL-risk*) with respect to hazard-free T_{p}, is proposed (Bradley 2001).

In general, risk of loss L of throughput capacity has two components, namely the probability of a hazard occurring, and the size of the loss in throughput with respect to some standard level of throughput. A MEL-risk of loss L means that the average loss, with respect to the mean throughput capacity $\mathcal{T}_p$ in a period where the hazard does not occur, is exactly L.

The standard deviation measure of possible loss with respect to the mean throughput capacity, $\mathcal{T}_p$, is the SD-risk measure. This measure is obtained by determining the standard deviation of the mean s of all the deviations $(L_1, L_2, \ldots, L_n)$ from the mean throughput capacity $\mathcal{T}_p$.

An SD-risk of s means that, in the next time unit, there is:

- a 50% chance or probability of a loss from the expected throughput capacity $\mathcal{T}_p$,
- a 34.1% chance of a loss between 0 and s from the expected $\mathcal{T}_p$,
- a 15.9% chance of a loss $> s$.

For a two-standard deviations measure, there is a 47.7% chance of a loss between 0 and $2s$ with respect to $\mathcal{T}_p$. This implies that there is a 13.6% chance of a loss between s and $2s$, and a 2.3% chance of a loss $>2s$, both losses with reference to the mean throughput capacity $\mathcal{T}_p$. In specifying an SD-risk, the standard deviation of the variations in throughput must be specified, as well as the standard level of throughput.

A 2-SD-risk of $2s$ means that, in the next time unit, there is:

- a 50% chance or probability of a loss from the expected throughput capacity $\mathcal{T}_p$,
- a 47.7% chance of a loss between 0 and $2s$ from the expected $\mathcal{T}_p$,
- a 2.3% chance of a loss $> 2s$.

It is assumed that the losses in each time unit are distributed normally, and the percentages are obtained from a normal distribution function table. These percentages will inevitably be different if the distribution departs from normal. The SD-risk measure is widely used in financial risk analysis, particularly for stock and bond portfolio management, since stock and bond prices follow a random pattern that gives rise to a near-normal distribution of price changes (Beaumont 1986).

Where there is exposure to future loss, which can be made up of two loss components, namely a *certain loss* and a *probable loss*, the SD-risk measure considers only the *probable loss*, which in effect is the true risk. This is better explained with the aid of an example: assume a system has a mean throughput capacity $\mathcal{T}_p = 400$ if there was no future loss exposure. Suppose that the system has exposure to a future loss in $\mathcal{T}_p$ with a mean of 100 and a standard deviation of 14 where the *least* loss is always greater than 70. This implies a *certain loss* of 70 plus a loss that makes up the balance with a mean of 30. This balance can, however, be as small as 0 (left side of the mean) and as large as 60 (right side of the mean), with a standard deviation of 14. The future loss thus has a *certain loss* of 70 and a *probable loss* of 30, with a standard deviation about the mean of the loss variations of 30 that is equal to 14. This standard deviation about the mean of the *probable loss* is the SD-risk. The system has a *certain loss* of 70 and a *probable loss* with a mean of 30 and an SD-risk of 14.

To deal with the problems that arise in arbitrary systems, where variations in throughput depart significantly from the normal distribution and the distribution of losses is not normal, an additional risk measure becomes essential. This is the mean expected loss risk (*MEL-risk*). Suppose that for a system exposed to risk, there is at least one hazard-free time period in which, by chance, the hazard does not occur, and where the loss with respect to the mean throughput capacity $\mathcal{T}_\mathrm{p}$ is L in this hazard-free time period, and where a loss exceeding L is not probable (but a loss less than L is probable). Thus, in the best-case scenario, the total hazard-free throughput capacity is $\mathcal{T}_\mathrm{p} - L$. Then all other throughput capacities, each in a time period where the hazard *does* occur in varying degrees of intensity, i.e. $\mathcal{T}_\mathrm{p} - L_1, \mathcal{T}_\mathrm{p} - L_2, \ldots, \mathcal{T}_\mathrm{p} - L_n$, may be considered as exhibiting losses, or loss deviations, with respect to the value of $\mathcal{T}_\mathrm{p}$ in the hazard-free time period. The mean of these loss deviations from $\mathcal{T}_\mathrm{p}$ in a hazard-free time period may be used as a measure of the *risk*. This measure of expected loss in the future with respect to the throughput capacity for a hazard-free time period is the mean expected loss risk (*MEL-risk*). Thus, a MEL-risk of loss L means that the average loss, with respect to the mean throughput capacity $\mathcal{T}_\mathrm{p}$ in a time period where the hazard does *not* occur, is exactly L. In specifying a MEL-risk, the mean deviation of the variations in throughput must be specified, as well as the standard level of throughput. A MEL-risk of loss L is two standard deviations from the mean $\mathcal{T}_\mathrm{p}$. The definitions of the loss variance, standard deviation or SD-risk, and two standard deviations or MEL-risk of loss L from the mean $\mathcal{T}_\mathrm{p}$ are considered by their formulation.

The *variance (V)* is the square of the differences between the losses and their average

$$V = (1/n) \cdot \sum (L_k - A_L)^2 \tag{5.20}$$

where:

L_k = the loss L_k ($k = 1$ to n) for n losses
A_L = the average (or mean) $(1/n)\sum L_k$.

The *standard deviation (SD)* is the *spread* about the average (or mean)

$$\begin{aligned} \mathrm{SD}^2 &= (1/n) \cdot \sum (L_k - A_L)^2 \\ \mathrm{SD} &= \sqrt{(1/n) \cdot \sum (L_k - A_L)^2} \end{aligned} \tag{5.21}$$

SD is the *root mean square deviation* between the losses and their average (SD^2 is the difference between the average of the squares and the square of the average), and can be computed as

$$\text{MEL-risk} = \sqrt{(1/n) \cdot L_k^2 - \left\{(1/n) \cdot \sum L_k\right\}^2} \tag{5.22}$$

A1—standard deviation, SD_1

$$\begin{aligned} \mathrm{SD}_1 &= \sqrt{(1/n) \cdot L_k^2 - \left\{(1/n) \cdot \sum L_k\right\}^2} \\ \mathrm{SD}_1 &= \text{SD-risk} \end{aligned} \tag{5.23}$$

A2—standard deviation, SD_2

$$\begin{aligned} SD_2 &= \sqrt{(1/n)\cdot L_k^2 - \left\{(1/n)\cdot \sum L_k\right\}^2} \\ SD_2 &= \text{MEL-risk} \end{aligned} \tag{5.24}$$

where:

L_k = the loss $L_k (k = 1 \text{ to } n)$ for n losses.

There are two extreme cases with regard to $\mathcal{T}_p$ for a hazard-free period of time (Bradley 2001):

(i) Explicit hazard-free case:
In the explicit case, the hazard-free throughput capacity $\mathcal{T}_p - L$ cannot be exceeded beyond the value of L. This throughput capacity remains in a time period when no hazard occurs. However, a hazard is certain to occur sometime. Thus, over a period of time, there will be a distribution of n losses about the mean and, in at least one of the n time periods, there will occur a loss deviation L with respect to the mean throughput capacity $\mathcal{T}_p$. However, no loss deviation below L will ever occur. The concept of a hazard-free throughput capacity level $\mathcal{T}_p - L$ implies:

(1) that no variation in throughput capacity can occur leading to a throughput capacity below the hazard-free level, and
(2) that the only variations in throughput capacity that can occur must lead to a throughput capacity at or below the hazard-free level.

This ensures that all *probable losses* are included in, and *certain losses* excluded from, the MEL-risk measure.

(ii) Implicit hazard-free case:
In the implicit case, the values in each time period fluctuate about the mean throughput capacity $\mathcal{T}_p$, and the distribution of the deviations from the mean follows some reasonably bell-shaped distribution, where large but usually improbable loss deviations from the mean throughput capacity $\mathcal{T}_p$ occur, and where no explicit hazard-free throughput capacity can be determined. In such a case, a hazard-free throughput capacity $\mathcal{T}_p - L$ may be defined where the loss L is two standard deviations from the mean.
For this case, the MEL-risk is defined as the mean expected loss with respect to $\mathcal{T}_p - L$ for the hazard-free period, with a value equivalent to two standard deviations of the mean throughput capacity $\mathcal{T}_p$.

MEL-risk can therefore be viewed as the hazard-free deviation, either explicit or implicit, from the throughput capacity $\mathcal{T}_p$, and is also equal to the average loss to be expected in a future hazard-free time period, with respect to throughput capacity T_p.

5.2.2.3 Hazard and Operability Studies for Risk Prediction

Safety issues have to be considered throughout an engineered installation's life cycle, from design, manufacture, installation, assembly and construction, through to start-up and operation. The later the hazardous operating modes are detected in this development process, the more serious and expensive they become to avoid or mitigate in terms of the required plant modifications. Thus, an extensive and systematic examination of safety aspects has to be carried out carefully and at the earliest possible opportunity in the engineering design stage. To meet these essential demands, a thorough safety and hazards analysis is compulsory during the engineering design and development stages, for official approval to commence with construction.

The initial step of such an analysis is *process hazard identification (PHI)*, which aims at identifying potential hazards that may be caused either by the nature of the process or the intended systems configuration. Further steps in this analysis are achieved by a variety of methods such as what-if analyses and safety checklists, usually incorporated in a more formal *hazard and operability study (HazOp)* conducted as early as possible in the conceptual and/or preliminary design phases. However, investigations in these early design phases identify faults only in the basic plant layout because no detailed specifications of process behaviour, or of the required controller equipment, may yet be available. Therefore, in the later detail engineering phase, further examination of the dynamic behaviour of systems is necessary to determine fail safe control by programmable logic controllers (PLCs) or distributed control systems (DCSs).

The technique of HazOp has been used and developed over approximately four decades for identifying potential hazards and operability problems caused by deviations from the design intent of both new and existing process plants. Because of the high profile of process plant accidents, emphasis has often been placed upon the identification of hazards but, in so doing, potential operability problems have been neglected. Yet, it is in the latter area that benefits of a HazOp study are usually the greatest. With respect to 'design intent', all industrial processes are designed for a purpose. Process systems are designed and constructed to achieve desired objectives. In order to do so, each item of equipment must consistently function according to specified criteria. These criteria can be classified as the 'design intent' for each particular item.

As an example, in the cooling water system of Fig. 5.5, consider now the cooling water circuit piping in which the pumps are installed. A simplified statement of the design intent of this small section of the reactor cooling system would be 'to continuously circulate cooling water at an initial temperature of X °C and at a rate of Y l per hour'. It is usually at this low level of design intent that a HazOp study is directed. The use of the word 'deviation' now becomes easier to understand. In the case of the cooling water circuit, a deviation or departure from the design intent would be a cessation of circulation, or the water being at an excessively high initial temperature. It is important to note the difference between a deviation and its cause. In this case, failure of the pump would be a cause, not a deviation, and a bent shaft due to insufficient lubrication would be a possible root cause. Essentially, the HazOp

procedure involves taking a full description of a process system and systematically questioning every part of it to establish how deviations from the design intent can arise. Once identified, an assessment is made as to whether such deviations and their consequences can have a negative effect upon the safe and efficient operation of the system. If considered necessary, remedial action is then taken.

An essential feature in this process of questioning and systematic analysis is the use of *keywords* to focus attention on deviations and their possible causes. In Sect. 5.2.1.5, keywords consisted of guidewords, attributes and process parameters. In the early conceptual phase of engineering design, when many equipment attributes and process parameters have not yet been defined but it is considered expedient to conduct a preliminary HazOp study, these keywords are simplified by grouping into two subsets:

- Primary keywords, which focus attention upon a particular aspect of the design intent or an associated process condition or parameter
  (e.g. isolate, vent, open, clean, drain, purge, inspect, maintain, start-up and shutdown).
- Secondary keywords, which are combined with a primary keyword to suggest possible deviations
  (e.g. no, less, more, also, other, early, late, reverse, fluctuation).

The usefulness of a preliminary HazOp study thus revolves around the effective application of these two subsets of keywords—for example, (pressure/*maintain*) (pressure/*less*) as primary and secondary keyword combinations.

a) Primary and Secondary Keywords

Primary keywords reflect both the process design intent and operational aspects of the system being studied. Typical process-oriented words are very similar to the process parameters of Sect. 5.2.1.5, as the words employed will generally depend upon the process being studied, whether at systems level or at a more detailed component level. However, the technique is hazard *and* operability studies; thus, added to the above primary keywords might be relevant operational words such as those given in Table 5.9.

This latter type of primary keyword is sometimes either overlooked or given secondary importance. Improper consideration of the word ‘isolate’, for example, can result in impromptu and sometimes hazardous means of taking a non-essential

Table 5.9 Operational primary keywords

Isolate	Drain
Vent	Purge
Open	Inspect
Clean	Maintain
Start-up	Shutdown

item of equipment offline for repairs because no secure means of isolation has been provided. Sufficient consideration of the words 'start-up' and 'shutdown' are particularly important, as most hazardous situations arise during these activities. For example, during commissioning it is found that the plant cannot be brought on-stream because no provision for safe manual override of the safety system trips has been provided, or it may be discovered that it is necessary to shut down an entire system just to re-calibrate or replace a pressure gauge.

Secondary keywords are similar to the HazOp guidewords of Sect. 5.2.1.5 and, when applied in conjunction with a primary keyword, they suggest potential deviations or problems. Although they tend to be a standard set, the following list is taken from Table 5.5 with a review of their meanings in line with industrial processes (Table 5.10).

Table 5.10 Operational secondary keywords: standard HazOp guidewords

Secondary keywords (standard HazOp guidewords)

Word	Meaning
No	The design intent does not occur (e.g. flow/no) or the operational aspect is not achievable (isolate/no)
Less	A quantitative decrease in the design intent occurs (e.g. pressure/less)
More	A quantitative increase in the design intent occurs (e.g. temperature/more)
Reverse	The opposite of the design intent occurs (e.g. flow/reverse)
Also	The design intent is completely fulfilled but, in addition, some other related activity occurs (e.g. flow/also, indicating contamination in a product stream, or level/also meaning material in a tank or vessel that should not be there)
Other	The activity occurs but not in the way intended (e.g. flow/other could indicate a leak or product flowing where it should not, or composition/other might suggest unexpected proportions in a feedstock)
Fluctuation	The design intention is achieved only part of the time (e.g. an airlock in a pipeline might result in flow/fluctuation)
Early	Usually used when studying sequential operations; this would indicate that a step is started at the wrong time or done out of sequence
Late	Usually used when studying sequential operations; this would indicate that a step is started at the wrong time or done out of sequence

b) HazOp Study Methodology

In simple terms, the HazOp study process involves systematically applying all relevant keyword combinations to the system in question, in an effort to uncover potential problems. The results are recorded in columnar format under the following headings:

node, attributes/parameters, deviations, causes, consequences, safeguards, action.

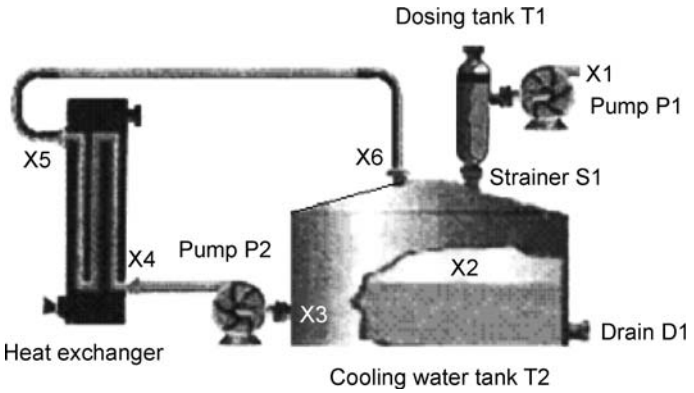


Fig. 5.21 Example of part of a cooling water system

In considering the information to be recorded in each of these columns, an example of part of the cooling water system depicted in Fig. 3.18 of Sect. 3.2.2.6 dealing with fault-tree analysis is illustrated in the simple schematic below (Fig. 5.21).

HazOp Study for Part of Cooling Water System

Process from–to nodes
$X_1 \rightarrow X_2$.

Attributes

Pump P_1	flow, pressure
Dosing tank T_1	flow, level
Strainer S_1	flow
Cooling water tank T_2	flow, level.

Deviation
The keyword combination being applied (e.g. no/flow).

Cause
Potential causes that would result in the deviation occurring (e.g. 'strainer S_1 blockage due to impurities in dosing tank T_1' might be a cause of flow/no).

Consequence
The consequences that would arise, both from the effect of the deviation (e.g. 'loss of dosing results in incomplete precipitation in T_2') and, if appropriate, from the cause itself (e.g. 'cavitations in pump P_1, with possible damage if prolonged').

The recording of consequences should be explicit. An important point to note, particularly for hazard and operability modelling (included later in this paragraph), is that when assessing the consequences, credit for protective systems or instruments that are already included in the design should not be considered.

Safeguards
Any existing protective devices that either prevent the cause or safeguard against the adverse consequences must be recorded. For example, the recording 'local pressure gauge in discharge from pump might indicate problem was arising' might be considered. Safeguards need not be restricted to hardware but, where appropriate, credit can be taken for procedural aspects such as the use of a standard work instruction (SWI) and job safety instructions (JSI).

Action
Where a credible cause results in a negative consequence, it must be decided whether some action should be taken. It is at this stage that consequences and associated safeguards are considered. If it is deemed that the protective measures are adequate, then no action need be taken, and words to that effect are recorded in the 'action' column.

Actions fall into two groups:

- Actions that remove the cause.
- Actions that mitigate or eliminate the consequences.

Whereas the former is to be preferred, it is not always possible, especially when dealing with equipment malfunction. However, removing the cause first should always take preference and, only where necessary, the consequences mitigated. For example, to return to the example cause 'strainer S_1 blockage due to impurities etc.', the problem might be approached in a number of specific remedial ways:

- Ensure that impurities cannot get into T_1, by fitting a strainer in the offloading line. Consider carefully whether a strainer is required in the suction to pump P_1. Particulate matter might pass through the pump without causing any damage, and it might be necessary to ensure that no such matter gets into T_2. If the strainer can be dispensed with altogether, the cause of the problem might be removed.
- Fit a differential pressure gauge across the strainer, with perhaps a high alarm to give clear indication that a total blockage is imminent.
- Fit a strainer, with a regular schedule of changeover and cleaning of the standby unit.

Having gone through the steps involved in recording a single deviation, the technique can now be inserted in the context of a qualitative hazard and operability computational model. Such a model is quite feasible, as the HazOp study method is an iterative process, applying in a structured and systematic way the relevant keyword (guideword-parameter) combinations in order to identify potential problems.

The example serves to highlight several points of caution when formulating actions:

Thus, it is not always advisable to automatically opt for an engineered solution, adding additional instrumentation, alarms, trips, etc. Due regard must be taken of the reliability of such devices, and their potential for spurious operation causing unnecessary downtime. In addition, the increased operational cost in terms of maintenance, regular calibration, etc. should also be considered. It is not unknown for

an over-engineered solution to be less reliable than the original design because of inadequate testing and maintenance. Furthermore, it is always advisable to take into account the level of training and experience of the personnel who will be operating the plant. Actions that call for elaborate and sophisticated protective systems are often wasted, as well as being inherently hazardous, if operators do not understand how they function.

c) Hazard and Operability Modelling

A crucial step in support of a HazOp analysis is to find a suitable *discrete event system (DES)* representation for the physical system behaviour, generally described by continuous dynamics. However, systems modelling approaches have to be adapted to the information that is available at certain points in the design stage.

To create a model that is appropriate for PHI, a method must be developed that qualitatively maps the dynamics in state transition systems. This type of model is ideal for HazOp but is often not sufficient for controller verification, especially if thresholds of timeouts have to be considered. Thus, the initial model, derived in the early design phases, must be refined by adding quantitative information so that a timed discrete event system is obtained for controller verification in the detail engineering design phase. As a basis for a concept to check the safety of a process system in different design stages, the physical systems behaviour is mapped into state transition systems given as a 6-tupel

$$\mathrm{TS} = (S, S_0, I, O, \phi, \theta) \tag{5.25}$$

where:

TS = state transition system
S = finite set of states
S_0 = set of initial states, where $S_0 \subseteq S$
I = finite input
O = finite output.

Furthermore:

$\phi : S \cdot I \rightarrow 2^S$ denotes the state transition function
$j : S \cdot I \rightarrow \theta$ denotes the state output function.

Application of the model (in computerised form) in a HazOp study relates system behaviour, mapped into state transition systems, to the HazOp guidewords of ‘none’, ‘more of’, ‘less of’, ‘reverse’, ‘part of’, ‘more than’, ‘other than’, etc. This type of DES is appropriate to represent the system’s behaviour qualitatively. However, to introduce quantitative information into the TS, time-dependent transitions must be augmented, which will be considered later.

d) Qualitative Modelling for Hazard Identification

In a typical model-based PHI, as it is established in the process industries, a team of experts systematically examines a system's related process flow diagram (PFD) and currently available piping and instrumentation diagram (PID).

To analyse failures and all conceivable deviations from the desired operation, the HazOp guidewords 'none', 'more of', 'less of', 'reverse', 'part of', 'more than', 'other than', etc. are used to qualitatively describe the dynamic behaviour of the system. If an inadequacy or a potential hazard is identified, appropriate countermeasures have to be added. Current topics of research to formalise this procedure are based on fuzzy modelling (Wang et al. 1995) or expert systems (Vaidhyanathan et al. 1996).

In the conceptual engineering phase, further information about the detail of the process, such as secondary reactions, equipment operations, and final mass and energy balances, is still vague. All data are eventually summarised in the PID and supplemented by additional information about the purposes of controllers and safety devices—but no exact specifications and detailed numeric data about the physical functions are yet available. Thus, the interaction between the system's physical behaviour and the controller actions can be modelled only qualitatively (to the degree of abstraction used in a HazOp study based on the guidewords). However, even a qualitative model must have features to express causality and the temporal order of actions. The procedure of creating a model according to Eq. (5.25) is carried out by the following four steps:

1. For each systems unit of a plant (reactor, pressure vessel, etc.) or item of equipment of a system (tank, pump, etc.)—depending on the level of resolution of the process at the particular design phase—the set V of process variables $v \in V$ describing physical behaviour is identified. This set typically comprises process quantities such as temperature, pressure, level, input flow and output flow.
2. Second, a set Q_j of qualitative states is introduced for each process variable v_j, e.g. the states 'critically low', 'low', 'normal', 'high' and 'critically high' for a process variable 'pressure'. The set of states in Eq. (5.25) follows from $S = Q_1, Q_2, \ldots, Q_j$. Usually, the set of initial states S_0 corresponds to the system's normal operation mode.
3. The third step, a crucial one, is to define the interactions of the process variables that are given as transitions between states in S depending on triggering signals. Thus, for each pair of states, $\sigma_1, \sigma_2 \in S$, the analyst decides whether a physical effect $i_k \in I$ exists that can cause a transition between the states

$$\begin{aligned} \phi_1 k_2 &: (\sigma_1, i_k) \rightarrow \sigma_2 \\ \phi_2 k_1 &: (\sigma_2, i_k) \rightarrow \sigma_1 \end{aligned} \tag{5.26}$$

   In this case, the enabling/enforcing effect is included into TS.
4. The modeller has to examine if the triggering input signal i_k has any further effect on the process behaviour. If there is an effect, then an output signal $O_1 \in O$

that specifies this behaviour is introduced as

$$\phi_1 k_1 : (\sigma_1, i_k) \rightarrow O_1 \tag{5.27}$$

An important aspect of creating the DES is that, in accordance with the HazOp study, even unlikely triggering events and their consequences must be modelled. A discrete model derived like this is not only suitable for PHI but can also be used as a basis for later model refinement in the detail design phase. Relying upon a safe system function defined in the early engineering design phases, one task of the later detail design phase is to design supervisory controllers that ensure the exclusion of dangerous operating modes.

To solve this task, model-based verification is used, which includes the following:

- A DES model of the system, including all possible physical behaviours, is generated.
- The controller specifications are transformed into a DES representation, and the combination of both yields a discrete model of the controlled system.
- The avoidance of dangerous states is verified or falsified by *reachability analysis*.

e) Quantitative Representation of Uncontrolled Processes

An analysis aiming to check whether a supervisory controller always ensures safe systems operation must satisfy the following questions:

- If a system's state moves in the direction of a critical situation, does the controller always react with an appropriate countermeasure to avoid this situation?
- Has the threshold of a process variable (or a threshold of time) at which a countermeasure is applied been chosen correctly, to avoid the critical state?

In principle, a transition system obtained from qualitative modelling, such as (Eq. 5.25), is sufficient to answer the first question. However, an examination of controller thresholds asks for a model comprising also numerical data for thresholds, and information about the duration for which a discrete state is active.

In this case, the DES of (Eq. 5.25) is extended to a timed transition system given as 7-tupel

$$\mathrm{TTS} = (S, S_0, I, O, \phi, \theta, \tau) \tag{5.28}$$

where:

TTS = timed transition system
S = finite set of states
S_0 = set of initial states, where $S_0 \subseteq S$
I = finite input
O = finite output
τ = finite set of clocks.

Furthermore

$\phi : S \cdot I \cdot \psi\tau) \rightarrow 2^S$ denotes the state-time transition function
$j : S \cdot I \cdot \psi\tau) \rightarrow \theta$ denotes the state-time output function.

In contrast to the TS of (Eq. 5.25), the TTS contains a finite set of clocks τ, and the state transition function $\phi : S \cdot I \cdot \psi(\tau)$ depends on logical propositions $\psi(\tau)$ over the clock variables.

f) Checking Safety by Reachability Analysis

Based on the discrete models generated as described in Eqs. (5.25) and (5.28), a comprehensive investigation of the system's safety is possible. The concept of *reachability analysis (RA)* is appropriate for checking safety in different design phases, since it is applicable to models of both degrees of abstraction (i.e. qualitative – Eq. 5.25, and quantitative – Eq. 5.28).

If SC denotes the set of critical states, a complete search over all possible runs of the DES shows whether a path from an initial state $s \in S_0$ to a critical state contained in SC exists – in this case, a hazard is identified, and respectively the correspondence of controller implementation and specification is falsified. Obviously, the analysis of the refined model of (Eq. 5.28) is more costly because the time constraints $\psi(\tau)$ have to be considered in determining the transitions. Thus, to minimise the computational effort, model refinement should be limited to the necessary.

For preliminary hazards identification (PHI), alternative strategies can be considered. Following the HazOp study method, design failures can be identified by forward simulation of the state transition model of (Eq. 5.25). In fact, such a simulation imitates the application of guidewords, since a possible deviation from normal operation can be assumed by generating the corresponding input signal, and the propagation of its effect is investigated as a sequence of transitions in the model. However, such a hazard identification approach relies on the user's intuition in choosing the right starting scenario, as well as one of several non-deterministic choices during the simulation.

The application of hazard and operability modelling during the conceptual design phase, including preliminary hazards identification (PHI) and reachability analysis (RA) in a specific industrial process engineering example, is considered in detail in Sect. 5.3.1.

5.2.3 Theoretical Overview of Safety and Risk Assessment in Preliminary Design

Safety and risk assessment attempts to estimate the expected safety risk and criticality for each individual *system* or *assembly* at the upper systems levels of the systems breakdown structure (SBS). Safety and risk assessment ranges from estimations of

the safety risk of relatively simple systems with series and parallel *assemblies*, to estimations of the safety risks of multi-state systems with random failure occurrences. Safety and risk assessment is considered in the *schematic* or *preliminary design* phase of the engineering design process, and includes basic concepts of modelling such as:

i. Markov point processes in designing for safety.
ii. Fault-tree analysis for safety systems design.
iii. Common cause failures in root cause analysis.

5.2.3.1 Markov Point Processes in Designing for Safety

A *point process* is intended to model a probabilistic situation that places points on a time axis. For safety analysis, these points are termed accident or incident *events*. To express these points mathematically in an *event space* Ω, the following notation is used: if A is a set of events in Ω, then N_A is the number of events in the set A, while if t is a positive real number, then $N(t)$ is the number of events on $(0,t]$. Thus, for example
if:

$$N(t) = N(0,t]$$

then:

$$N(a,b] = N(b) - N(a)$$

and:

$$N\{a\} = \text{the number of events at the point } a. \tag{5.29}$$

A point process has no simultaneous event (i.e. more than one accident and/or incident cannot occur simultaneously on the same equipment at the same time) if each step of $N(t)$ is of unit magnitude (where t is measured in units of time such as seconds, minutes, hours, days, etc.), with complete certainty (i.e. probability $= 1$) (Thompson 1988).

a) Point Process Parameters

In developing parameters of a point process, let $M(t)$ be the expected value or mean of $N(t)$. Thus

$$M(t) = \bar{E}N(t) \tag{5.30}$$

where:

$M(t)$ = a non-decreasing continuous function
E = expected value.

Taking derivatives

$$\mu(t) = \mathrm{d}/\mathrm{d}t[M(t)] = M'(t) \tag{5.31}$$

where:

$\mu(t)$ = instantaneous rate of change of the expected value of the number of events with respect to time t.

The instantaneous rate of change, $\mu(t)$, is termed the event or *incident rate* of the process. Thus, in modelling a system or its equipment for reliability and/or safety with respect to hazards (or events in a point process) during the schematic or preliminary design phase, the incident rate of the process is, in effect, the failure rate of the system. However, it must be expressly noted that this concept of incident rate differs from the failure rate of the age distribution of equipment. Obviously, equipment ages with use over a period of time, and becomes more prone to failure (i.e. wear-out failure characteristic of the failure hazard curve of Fig. 3.19). This is the hazard rate function, $r(t)$, considered in Sect. 3.2.3 (refer to Eqs. 3.29 to 3.33), and expressed as

$$r(t) = \lim_{\Delta t \to 0} \frac{P(t \le Z < t + \Delta t)}{\Delta t} \tag{5.32}$$

$$= \frac{\lim_{\Delta t \to 0} F(t)}{1 - F(t)} \tag{5.33}$$

where

$$F(t) = \int_{x=t}^{t+\Delta t} f(x)\,\mathrm{d}x \tag{5.34}$$

The rates $r(t)$ and $\mu(t)$ are quite different, in that the pattern of $r(t)$ follows the *wear-out* shape of the failure hazard curve (bathtub or U-shaped curve), whereas the pattern of $\mu(t)$ is linear and follows the *random failure* or *useful life* shape of the failure hazard curve. Another function of point processes, in addition to the incident rate $\mu(t)$, is the *intensity function*. If there are no simultaneous events, then the incident rate equals the intensity (Thompson 1988, cited Leadbetter 1970).

The intensity of point process events (accidents or incidents) can be expressed as

$$h(t) = \lim_{\Delta t \to 0} \frac{P(N(t + \Delta t)) \ge 1}{\Delta t} \tag{5.35}$$

where:

$h(t)$ = probability of one more event in the interval $t + \Delta t$.

b) Markov Chains and Critical Risk

Critical risk theory hypothesises that, out of k risks, at least one will be critical with respect to the severity of their consequences. The theory is based on predicting a change in these consequences as a result of removing or adding a risk (Thompson 1988). For example, it attempts to predict a change in the *useful life* expectancy of a cooling water tank, if an ant-corrosion agent was added to the tank's contents; or to predict the probability of an increase in random occurrence events (failures) in electric pump motors due to pump seal deterioration as a result of the addition of an anti-corrosion agent to the cooling water circuit.

Critical risk theory assimilates a stochastic process where the transition probabilities from an earlier to a later state depend only on the earlier state, and the times involved. This is typical of Markov chains. Thus, critical risk implies that initially a system or an item of equipment is in an operable state 0 and, after a time period T, the system or equipment undergoes a state change or transition from being operable to being inoperable (i.e. failed) as a result of some consequence due to *critical risk C*.

For a *critical risk C*, where $C = 1,2,3,\ldots,k$, time and cause of failure are subject to chance. Only transitions from state 0 to one of the different states $0,1,2,3,\ldots,k$ are possible, in which the states $1,2,3,\ldots,k$ are considered to be *absorbing* (once in the system, they are never removed).

Let $P_{ij}(\tau,t)$ be the probability of transition from state i at time τ to state j at time t. Assume that the *intensity functions* $h_i(t)$ exist, and satisfy the following expressions

$$P_{00}(t,t+\Delta t) = 1 - \sum_{i=1}^{k} h_i(t)\Delta t + 0(\Delta t) \tag{5.36}$$

$$P_{0i}(t,t+\Delta t) = h_i(t)\Delta t + 0(\Delta t) \tag{5.37}$$

$$i = 1,2,3,\ldots k\,.$$

This yields the *Kolmogorov differential equations* (Oksendal 1985):

$$\frac{\mathrm{d}}{\mathrm{d}t}P_{00}(0,t) = -P_{00}(0,t)\cdot h(t) \tag{5.38}$$

$$h(t) = \sum_{i=1}^{k} h_i(t) \tag{5.39}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}P_{0i}(0,t) = P_{0i}(0,t)\cdot h(t) \tag{5.40}$$

$$i = 1,2,3,\ldots k\,.$$

c) Review of Kolmogorov Differential Equations

It is useful at this point to review the *Chapman–Kolmogorov equation*, which states that

$$P_{ij}(s+t) = \sum_k P_{ik}(s) \cdot P_{kj}(t) \tag{5.41}$$

or, in matrix terms

$$P(s+t) = P(s) \cdot P(t) \tag{5.42}$$

Note that $P(0) = I$, which is the *identity matrix*. For integer t, it follows that $P(t) = P(1)^t$ but then t need not be an integer. Setting $t = \mathrm{d}s$ in the Chapman–Kolmogorov equation gives

$$\begin{aligned} P(s+\mathrm{d}s) &= P(s) \cdot P(\mathrm{d}s) \\ P(s+\mathrm{d}s) - P(s) &= P(s) \cdot [P(\mathrm{d}s) - I] \\ P'(s) &= P(s)\, Q \end{aligned} \tag{5.43}$$

where:

$Q = P'(0)$ is the matrix (called the Q-matrix or the generator matrix of the chain).

This is termed the *Kolmogorov forward equation*, which is one part of the Kolmogorov differential equations. The Kolmogorov forward equation can be derived as follows:

$$\begin{aligned} P[X(s+\mathrm{d}s) = j] &= \sum_k P[X(s+\mathrm{d}s) = j | X(s) = k] P[X(s) = k] \\ &= \sum_{k \neq i} P[X(s) = k] \cdot q_{ki}\, \mathrm{d}s + \left(1 - \sum_{k \neq i} q_{ki}\right) P[X(s = j)] \end{aligned}$$

If $q_{kk} = -\sum_i q_{ki}$ then:

$$\frac{\mathrm{d}}{\mathrm{d}s} P[X(s) = k] = \sum_k P[X(s) = k] \cdot q_{ki}$$

The *Kolmogorov backward equation* (Eq. 5.44) is obtained by inserting $s = \mathrm{d}t$ into the previous Chapman–Kolmogorov equation:

$$P \neq (t) = QP(t) \tag{5.44}$$

To appreciate the difference between the forward and backward equations, there are two different ways of evaluating the linear birth-and-death process (or, in this case, the operable and failed states). It is theoretically possible to solve the Kolmogorov equation, giving the solution:

$$P(t) = \mathrm{e}^{Qt} = \sum_n t^n \cdot Q^n / n\,!$$

However, this solution is not very useful because Q^n is difficult to evaluate; a simpler method is the use of matrices, utilising the Q-matrix, or the generator matrix of the chain.

d) The Q-Matrix

The row sums of the Q-matrix are always zero. For example, in the case of a linear birth-and-death process, the rate of transitions from x to $x+1$ is the birth rate $x\beta$ and, from x to $x-1$, the death rate $x\delta$. Therefore, with all other entries in the Q-matrix being zero:

$$q_{x,x-1} = x\delta, \quad q_{x,x+1} = x\beta, \quad \text{and} \quad q_{x,x} = -(\beta+\delta)x$$

Thus, the *Q-matrix* is represented in tabular form as:

Table 5.11 Values of the Q-matrix

0	0	0	0	–
δ	$-(\beta+\delta)$	β	0	–
0	2δ	$-2(\beta+\delta)$	2β	–
0	0	3δ	$-3(\beta+\delta)$	3β

The time until the next event, starting in x, has an exponential distribution with rate $\lambda_x = -q_{x,x}$, after which it changes state according to the transition matrix R. For calculating state change probabilities, the expected time to change to a particular state, especially the expected time to the first state change, is $1/\lambda_x$. State change problems such as 'find $h_x(t)$, the probability that X changes to state 0 before time t, starting from state x' can be treated in the following manner:

$$h_x(t) = \int_{0,t} \lambda_x \cdot \mathrm{e}^{-\lambda_x u} \left\{ q_{x0}/\lambda_x + \sum_{y\neq 0,x} \cdot q_{xy}/\lambda_x \cdot h_y(t-u) \right\} \mathrm{d}u$$

Substituting $v = t - u$:

$$h_x(t) = \int_{0,t} \mathrm{e}^{-\lambda_x v} \left\{ q_{x0} + \sum_{y\neq 0,x} \cdot q_{xy} \cdot h_y(v) \right\} / \mathrm{e}^{-\lambda_x u}$$

Differentiating, and setting $\lambda_x = -q_{x,x}$, the expressions obtained are easier to solve in specific cases:

$$h'(t) = Qh(t), h_0(t) = 1, h_x(0) = 0 \text{ for } x \neq 0$$

Returning to the Markov chain model, the Kolmogorov differential equations are

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}P_{00}(0,t) &= -P_{00}(0,t)\cdot h(t) \\ \frac{\mathrm{d}}{\mathrm{d}t}P_{0i}(0,t) &= P_{00}(0,t)\cdot h(t) \\ i &= 1,2,3,\ldots,k\,. \end{aligned} \tag{5.45}$$

These may be solved to yield the following relationships

$$P_{00}(0,t) = \exp\left[-\int\limits_{(0,t)} h(x)\,\mathrm{d}x\right] \tag{5.46}$$

$$P_{0i}(0,t) = \exp\left[-\int\limits_{(0,t)} h_i(x)\cdot P_{00}(0,x)\,\mathrm{d}x\right]$$

where the *survival function of the useful life expectancy* is expressed as

$$P_{00}(0,t) = F'(t) \tag{5.47}$$

The *hazard rate*, represented by the *intensity function*, is expressed as

$$h(t) = \sum_{i=1}^{k} h_i(t) \tag{5.48}$$

The *expected useful life* is expressed as

$$\mu = \int\limits_0^{\infty} F'(y)\,\mathrm{d}y \tag{5.49}$$

The joint probability of the *random failure occurrence* (useful life expectance), together with the *hazard rate*, is expressed as

$$\begin{aligned} P(Z \le z, C = i) &= P_{0i}(0,z) \\ P_{0i}(0,z) &= \int\limits_0^{z} F'(x)\cdot h_i(x)\,\mathrm{d}x \end{aligned} \tag{5.50}$$

The probability of failure resulting from critical risk C is expressed as

$$\begin{aligned} \prod_i &= P(Z \le \infty, C = i) \\ &= P_{0i}(0,\infty) \\ P_{0i}(0,\infty) &= \int\limits_0^{\infty} F'(x)\cdot h_i(x)\,\mathrm{d}x \end{aligned} \tag{5.51}$$

e) Critical Risk Theory in Designing for Safety

In applying critical risk theory to a series process engineering system, the following modelling approach is taken:

Assume the system consists of k independent components, each with expected useful life lengths of z_1, z_2, z_3, …, z_k, all of which must function for the system to be able to function, and where the useful life length of the system is Z.

Denoting the *survival function of the useful life expectancy* of Z by F', and of z_i by F_i' $(i = 1, 2, 3, \ldots, k)$, then

$$\begin{aligned} Z &= \min(z_1, z_2, z_3, \ldots, z_k) \\ F_i'(z) &= P_{00}(0, z_i) \end{aligned} \tag{5.52}$$

Then: $F'(Z) = \prod_{i=1}^{k} F'(Z)$.

The *hazard rate* represented by the intensity function can now be formulated

$$h(Z) = \sum_{i=1}^{k} h_i(Z) \tag{5.53}$$

The probability of failure resulting from critical risk is expressed as (Eq. 5.54):

$$P_{0i}(0, Z) = \int_0^{\infty} F'(Z) \cdot h_i(Z)\, \mathrm{d}z \tag{5.54}$$

Using the expression for the *hazard rate* $h_i(z)$ of useful life expectancy of Z_i, the *survival function of the useful life expectancy* of the series process engineering system is then expressed as

$$F_i'(Z) = \exp\left[-\prod_{i=1}^{K} \int_0^{z} \frac{f(z|C=i)}{F'(Z)}\, \mathrm{d}z \right] \tag{5.55}$$

f) The Concept of Delayed Fatalities

In assessing the safety of a complex process, critical risk may be considered as resulting in fatalities due to an accident. These fatalities can be classified as immediate or as delayed. It is the delayed fatalities that are of primary interest in high-risk engineered installations such as nuclear reactors (NUREG 75/014 1975; NUREG/CR-0400 1978). Critical risk analysis applies equally well to delayed fatalities as to immediate fatalities. To model the impact of delayed fatalities in the assessment of safety in engineering design, consider the effect of a new constant risk, with intensity $h(y)$, which is delayed for time d. The model parameters include the following expressions (Thompson 1988):

The intensity function for the new risk is:

$$\begin{aligned} h_{\text{new}}(y) &= 0 \; y \leq d \\ &= \lambda \; y > d \end{aligned}$$

The probability that the new risk is the critical risk (resulting in fatality) is (from Eq. 5.51)

$$\prod_i = P(y \leq \infty, C = i) \tag{5.56}$$

$$= P_{0i}(0, \infty)$$

$$P_{0i}(0, \infty) = \int_0^\infty F'(y) \cdot h_i(y) \, \mathrm{d}y \tag{5.57}$$

$$P_d(0, \infty) = \int_0^\infty \lambda \, \mathrm{e}^{-\lambda y} F'(y) \, \mathrm{d}y \tag{5.58}$$

$$= \lambda \int_d^\infty F'(y) \, \mathrm{d}y + (\lambda)$$

The *expected useful life* with the new risk delayed is expressed as (from Eq. 5.49)

$$\mu = \int_0^d F'(y) \, \mathrm{d}y + \int_d^\infty \mathrm{e}^{-\lambda y} F'(y) \, \mathrm{d}y \tag{5.59}$$

$$= \mu \int_d^\infty 1 - \mathrm{e}^{-\lambda y} F'(y) \, \mathrm{d}y$$

$$= \mu - \lambda \int_d^\infty F'(y) \, \mathrm{d}y + (\lambda)$$

The US Nuclear Regulatory Commission's Reactor Safety Study (NUREG 75/014 1975) also presents nuclear risk in comparison with the critical risk of other types of accidents. For example, the annual chances of fatality for vehicle accidents in the USA are given as 1 in 4,000, whereas for nuclear reactor accidents the value is 1 in 5 billion.

5.2.3.2 Fault-Tree Analysis for Safety Systems Design

For potentially hazardous process engineering systems, it is required statutory practice to conduct a quantitative assessment of the safety features at the engineering design stage. The design is assessed by predicting the probability that the safety

systems might fail to perform their intended task of either preventing or reducing the consequences of hazardous events. This type of assessment is best carried out in the preliminary design phase when the system has sufficient detail for a meaningful analysis, and when it can still be easily modified. Several methods have been developed for predicting the likelihood that systems will fail, and for making assessments on avoiding such failure, or of mitigating its consequence. Such methods include Markov analysis, fault-tree analysis, root cause and common cause analysis, cause-consequence analysis, and simulation. *Fault-tree analysis (FTA)* is the most frequently used in the assessment of safety protection systems for systems design.

a) Assessment of Safety Protection Systems

The criterion used to determine the adequacy of the safety system is usually a comparison with specific target values related to a system's probability to function on demand. The initial preliminary design specification is to predict its likelihood of failure to perform according to the design intent. The predicted performance is then compared to that which is considered acceptable. If system performance is not acceptable, then deficiencies in the design are removed through redesign, and the assessment repeated. With all the various options for establishing the design criteria of system configuration, level of redundancy and/or diversity, reliability, availability and maintainability, there is little chance that this approach will ensure that the design reaches its final detail phase with all options adequately assessed. For safety systems with consequence of failure seen as catastrophic, it is important to optimise performance with consideration of all the required design criteria, and not just adequate performance at the best cost. The target values should be used as a minimum acceptance level, and the design should be optimised with respect to performance within the constraints of the design criteria. These analysis methods are well developed and can be incorporated into a computerised automatic design assessment cycle that can be terminated when optimal system performance is achieved within the set constraints.

Safety systems are designed to operate only when certain conditions occur, and function to prevent these conditions from developing into hazardous events with catastrophic consequences. As such, there are specific features common to all safety protection systems—for example, all safety systems have sensing devices that repeatedly monitor the process for the occurrence of an initiating event. These sensors usually measure some or other process variable, and transmit the state of the variable to a controller, such as a programmable logic controllers (PLC) or distributed control system (DCS). The controller determines whether the state of the process variable is acceptable, by comparing the input signal to a set point. When the variable exceeds the alarm limit of the set point, the necessary protective action is activated. This protective action may either prevent a hazardous event from occurring, or reduce its consequence.

There are several design options with respect to the structure and operation of a safety system where, from a design assessment point of view, the *level of*

redundancy and *level of diversity* are perhaps the more important. The safety system must be designed to have a high likelihood of operability on demand. Thus, single component failures should not be able to prevent the system from functioning. One means of achieving this is by incorporating redundancy or diversity into the system's configuration. Redundancy duplicates items of equipment (assemblies, sub-assemblies and/or components) in a system, while diversity includes totally different equipment to achieve the same function. However, increased levels of redundancy and diversity can also increase the number of system failures. To counteract this problem, *partial redundancy* is opted for—e.g. *k* out of *n* sensors indicate a failed condition. It is specifically as a result of the *assessment* of safety in engineering design during the preliminary design phase that decisions are made where to incorporate redundancy or diversity, and if full or partial redundancy is appropriate.

b) Design Optimisation in Designing for Safety

The objective of design optimisation in designing for safety is to minimise system unreliability (i.e. probability of component failure) and system unavailability (i.e. probability of system failure on demand), by manipulating the design variables such that design criteria constraints are not violated. However, the nature of the design variables as well as the design criteria constraints engender a complexity problem in design optimisation.

Commonly with mathematical optimisation, an objective function defines how the characteristics that are to be optimised relate to the variables. In the case where an objective function cannot be explicitly defined, some form of the function must be assumed and the region defined over which the approximate function can be considered acceptable. Design criteria constraints fall into two categories: those that can be determined from an objective function relating to the design variables, which can be assessed mathematically, and those that cannot be easily expressed as a function, and can be assessed only through analysis. In the former case, a computational method is used to solve the design optimisation problem of a safety system. The method is in the form of an iterative scheme that produces a sequence of system designs gradually improving the safety system performance. When the design can no longer be improved due to restrictions of the design criteria constraints, the optimisation procedure terminates (Andrews 1994).

Assessment of the preliminary design of a safety system might require improvements to system performance. This could imply developing a means of expressing system performance as a function of the design variables

$$Q_{\text{system}} = f(V_1, V_2, V_3, \ldots, V_n) \tag{5.60}$$

where:
V_1, V_2, V_3, … , V_n are the design variables, typically including:

- the number of high-pressure valves,
- the number of pressure transmitters,

- the level of redundancy of valves,
- the number of transmitters to trip.

It is computationally difficult to develop a function Q that can consider all design options. However, with the use of a Taylor series expansion, the following expression is obtained

$$f(x+\Delta x) = f(x) + \mathbf{g}^{\mathrm{T}}\Delta x + \tfrac{1}{2}\Delta x^{\mathrm{T}} \cdot G \cdot \Delta x \tag{5.61}$$

where:

Δx = the change in the design vector
$\mathbf{g}$ = the gradient vector
G = the Hessian matrix.

The gradient $\mathbf{g}(x)$ is the first-order partial derivatives of $f(x)$

$$\mathbf{g}(x) = \left[\frac{\delta}{\delta x_1} f(x), \frac{\delta}{\delta x_2} f(x), \ldots \frac{\delta}{\delta x_n} f(x)\right] \tag{5.62}$$

The Hessian matrix $G(x)$ is a square symmetric matrix of second derivatives given as

$$G(x) = \begin{bmatrix} \dfrac{\delta^2 F}{\delta x_1 \delta x_1}, & \dfrac{\delta^2 F}{\delta x_1 \delta x_2}, & \cdots & \dfrac{\delta^2 F}{\delta x_1 \delta x_n} \\ \dfrac{\delta^2 F}{\delta x_2 \delta x_1}, & \dfrac{\delta^2 F}{\delta x_2 \delta x_2}, & \cdots & \dfrac{\delta^2 F}{\delta x_2 \delta x_n} \\ \vdots & \vdots & & \vdots \\ \dfrac{\delta^2 F}{\delta x_n \delta x_1}, & \dfrac{\delta^2 F}{\delta x_n \delta x_2}, & \cdots & \dfrac{\delta^2 F}{\delta x_n \delta x_n} \end{bmatrix} \tag{5.63}$$

Truncating (Eq. 5.61) after the linear term in Δx means that the function $f(x+\Delta x)$ can be evaluated provided that the gradient vector can be obtained, that is, $\partial f/\partial x$ for each design parameter. Since integer design variables are being dealt with, $\partial f/\partial x$ cannot be strictly formulated but, if consideration is taken of the fact that a smooth curve has been used to link all discrete points to give the marginal distribution of f as a function of x_i, then $\partial f/\partial x_i$ can be obtained. Partial derivatives can be used to determine how values of f are improved by updating each x_i by Δx_i. A fault tree can be developed to obtain $f(x+\Delta x)$ for each x_i provided $x_i + \Delta x_i$ is integer; finite differences can then be used to estimate $\partial f/\partial x_i$. This would require a large number of fault trees to be produced and analysed, which would usually result in this option not being pursued from a practical viewpoint.

Since truncating the Taylor series of (Eq. 5.61) at a finite number of terms provides only an approximation of $f(x+\Delta x)$, the solution space over which this approximation is acceptable also needs to be defined. This is accomplished by setting up a solution space in the neighbourhood of the design's specific target variable. This procedure results in an iterative scheme, and the optimal solution being approached by sequential optimisation.

c) Assessment of Safety Systems with FTA

Where design criteria constraints can be assessed only through analysis, fault-tree analysis (FTA) is applied. In the assessment of the performance of a safety system, a fault tree is constructed and analysed for two basic system failure modes: failure to work on demand, and spurious system trips. Fault trees are analysed in the design optimisation problem, to obtain numerical estimates of the partial derivatives of system performance with respect to each design variable. This information is required to produce the objective function coefficients. However, the requirement to draw fault trees for several potential system designs, representing the causes of the two system failure modes, would make the optimisation method impractical. Manual development of a new tree for each assessment would be too time-consuming. One approach in resolving this difficulty is to utilise computer automated fault-tree synthesis programs; at present, these have not been adequately developed to accomplish such a task. An alternative approach has been developed to construct a fault tree for systems design, using *house events* (Andrews et al. 1986).

House events can be included in the structure of fault trees, and either occur with certainty (event set to TRUE) or do not occur with certainty (event set to FALSE). Their inclusion in a fault-tree model has the effect of turning on or off branches in the tree. Thus, a single fault tree can be constructed that, by defining the status of house events, could represent the causes of system failure on demand for any of several potential designs. An example of a sub-system of a fault tree that develops causes of *dormant failure* of a high-pressure protection system, alternately termed a high-integrity protection system (HIPS), is illustrated in Fig. 5.22. In this example, the function of the HIPS sub-system is to prevent a high-pressure surge passing through the process, thereby protecting the process equipment from exceeding its individual pressure ratings. The HIPS utilises transmitters that determine when pipeline pressure exceeds the allowed limit. The transmitters relay a signal to a controller that activates HIPS valves to close down the pipeline. The design variables for optimisation of the HIPS sub-system include six house events (refer to Fig. 5.22) that can be summarised in the following criteria:

- what *type* of valve should be fitted,
- whether high-pressure valve *type* 1 should be *fitted*, or *not*,
- whether high-pressure valve *type* 2 should be *fitted*, or *not*.

The house events in the fault tree represent the following conditions:

H1 – HIPS valve 1 fitted
NH1 – HIPS valve 1 not fitted
H2 – HIPS valve 2 fitted
NH2 – HIPS valve 2 not fitted
V1 – Valve type 1 selected
V2 – Valve type 2 selected.

Considering first the bottom left-hand branch in Fig. 5.22 that represents ‘HIPS valve 1 fails stuck’, this event will depend on which type of valve has been selected

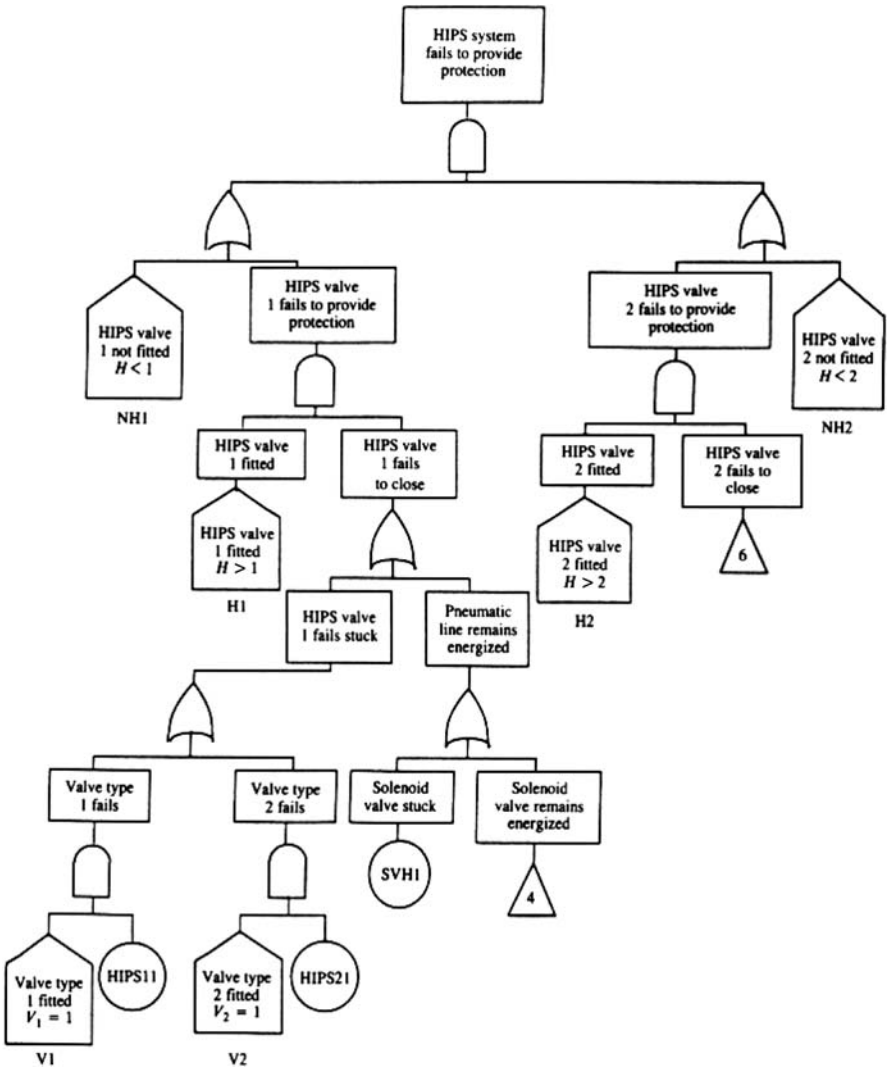


Fig. 5.22 Fault tree of dormant failure of a high-integrity protection system (HIPS; Andrews 1994)

in the design. If type 1 has been fitted, then V1 is set to TRUE. If type 2 is fitted, then V2 is set to TRUE. This provides the correct causes of the event being developed in function of which valve is fitted. One of either V1 or V2 must be set. Furthermore, if no HIPS option is included in the system design, then house events NH1 and NH2 will both be set (i.e. TRUE). Once these events are set, the output event from the OR gates into which they feed will also be true. At the next level up in the tree structure, both inputs to the AND gate will have occurred and, therefore, the HIPS system will not provide protection. Where HIPS valves are fitted, the appropriate house events

NH1 or NH2 will be set to FALSE, requiring component failure event to render the HIPS sub-system inactive.

By using house events in this manner, all design options can be represented in a single fault tree. Another fault tree can be constructed using the same technique to represent causes of spurious system failure for each potential design. The fault trees are then analysed to obtain numerical estimates of the partial derivatives of system performance with respect to each design variable. This information is required to produce the objective function coefficients in the design optimisation problem. The objective function is then derived by truncating the Taylor series at the linear term of the gradient vector, $\boldsymbol{g}$, and ignoring the quadratic term of the Hessian matrix. This truncation means that a finite number of terms provide an approximation of the objective function, with a valid representation of Q_{system} only within the neighbourhood of the target design variables. Additional constraints are therefore included to restrict the solution space in the neighbourhood of the design's specific target variables. The objective function is then evaluated in the restricted design space, and the optimal design selected.

5.2.3.3 Common Cause Failures in Root Cause Analysis

The concept of multiple failures arising from a common cause was first studied on a formal basis during the application of root cause analysis in the nuclear power industry. In order to obtain sufficiently high levels of reliability and safety in critical risk control circuits, redundancy was introduced. In applying redundancy, several items can be used in parallel with only one required to be in working order.

Although the approach increases system reliability, it leads to large increases in false alarms measured in what is termed the *false alarm rate (FAR)*. This is overcome, however, by utilising a concept termed *voting redundancy*; in its simplest arrangement, this is two out of three, where the circuit function is retained if two or three items are in working order. This not only improves reliability and safety but also reduces the FAR. Voting redundancy has the added advantage that a system can tolerate the failure of some items in a redundant set, allowing failed items to be taken out of service for repair or replacement (electronic control components such as sensors, circuit boards, etc. are usually replaced).

a) Defining CMF and CCF

It has become evident from practical experience in the process industry that, in many cases, the levels of reliability and safety that are actually being obtained have fallen short of the predicted design values. This is due largely to common root causes leading to the concurrent failure of several items. The concept of *common mode failures (CMF)* was developed from studies into this problem. It was subsequently recognised that multiple failures could arise from common weaknesses, where a particular item (assembly and/or component) was used in various locations on a plant.

Furthermore, the term *common cause failure (CCF)* was applied to the root causes of these failure modes, not only manifested at the parts and component level but also including the effects of the working environment on the item, e.g. the effects from the assembly, sub-system and system levels, as well as the process and environmental conditions. Consequently, the term *dependent failure* was used to include both CMF and CCF, although CMF is, in effect, a subset of CCF. Many terms relating to the integrity of systems and components were formally defined and included in a range of military standards, especially reliability and maintainability. However, it took some time before CMF and CCF were formally defined in the nuclear energy industry.

The UK Atomic Energy Authority (AEA) has defined CMF as follows (Edwards et al. 1979):

> "*A common-mode failure (CMF) is the result of an event which, because of dependencies, causes a coincidence of failure states of components in two or more separate channels of a redundancy system, leading to the defined system failing to perform its intended function*".

The UK Atomic Energy Authority has also defined CCF as follows (Watson 1981):

> "*A common-cause failure is the inability of multiple first in line items to perform as required in a defined critical time period, due to a single underlying defect or physical phenomena, such that the end effect is judged to be a loss of one or more systems*".

CCF can arise from both engineering and operational causes:

- *Engineering causes* can be related to the engineering design as well as manufacturing, installation and construction stages. Of these, engineering design covers the execution of the design requirement and functional deficiencies, while the manufacturing, installation and construction stages cover the activities of fabrication and inspection, packaging, handling and transportation, installation and/or construction. Plant commissioning is often also included in the engineering causes.
- *Operational causes* can be separated into procedural causes and environmental effects. The procedural causes cover all aspects of maintenance and operation of the equipment, while environmental causes are quite diverse in that they include not only conditions within the process (influenced partly by the process parameters and the materials handled in the process) but external environmental conditions such as climatic conditions, and extreme events such as fire, floods, earthquakes, etc. as well.

Typical examples of actual causes of CCF are (Andrews et al. 1993):

- Identical manufacturing defects in similar components.
- Maintenance errors made by the same maintenance crews.
- Operational errors made by the same operating crews.
- Components in the same location subject to the same stresses.

Since the earliest applications of CCF, two methods have been extensively used to allow for such events. These are the *cut-off probability method* and the *beta factor*

Table 5.12 Upper levels of systems unreliability due to CCF

Systems configuration	Minimum failure probability
Single instrument	10^{-2}
Redundant system	10^{-3}
Partially diverse system	10^{-4}
Fully diverse system	10^{-5}
Two diverse systems	10^{-6}

method. The cut-off probability method proposes limiting values of failure probability to account for the effect of CCF.

The basis of this is the assumption that, because of CCF, system reliability can never exceed an upper limit determined by the configuration of the system. These upper levels of systems *unreliability* were generically given as shown in Table 5.12 (Bourne et al. 1981).

The beta method assumes that a proportion, β, of the total failure rate of a component arises from CCF. It follows, therefore, that the proportion $(1-\beta)$ arises from *independent* failures. This can be expressed in

$$\lambda_t = \lambda_i + \lambda_{\text{ccf}} \tag{5.64}$$

where:

λ_t = the total failure rate
λ_i = the independent failure rate
λ_{ccf} = the common cause failure rate.

From this equation follows

$$\lambda_{\text{ccf}} = \beta \cdot \lambda_t$$

and:

$$\lambda_i = 1 - \beta \cdot \lambda_t \tag{5.65}$$

The results from the beta factor method must, however, be considered with some pessimism because they need to be modified for higher levels of redundancy than is needed for the simple one-out-of-two case. Although in theory CCF can occur, it does not follow that it will. The probability of failure of all three items of a two-out-of-three redundancy system due to CCF is likely to be lower than the probability of two failing (Andrews et al. 1993).

The cut-off method is thus extensively used where there are no relevant field data or even if any database is inadequate, and serves as a suitable guide in the preliminary design phase for determining the limiting values of failure probability to account for the effect of CCF. It is also quite usual in such circumstances to use the beta factor method, but this requires engineering judgment on the appropriate values for beta—in itself, this is probably no more accurate than using the cut-off method. A combination of both methods in the assessment of reliability and safety due to CCF in engineering design is best suited for application by expert judgment

in an *information integrated technology (IIT)* program considered in Sect. 3.3.3.4 and illustrated in Fig. 3.46.

The beta factor model is extensively used in predictions, in which the appropriate values for beta are selected on the basis of expert engineering judgement. The problem with the model, though, is the lack of any detailed data for generic systems, assemblies and components, to provide an adequate assessment of safety in engineering design—especially in the preliminary design phase. As a result, quite large beta factors have been applied without any justification, and caution needs to be exercised when selecting these beta values, otherwise the estimates will give unjustifiably pessimistic results with possible over-design of safety-related systems. A somewhat different approach to the beta factor model has thus been taken in which the beta values are not used but predictions are made directly from event data using expert judgment. This approach necessitates identifying the root causes of failure and the likelihood of generating simultaneous failures in similar equipment (Hughes 1987). Fundamentally, the basis of this approach, typical to IIT, is to represent the variability of a component failure probability by distributions that can be estimated directly from a relatively small database. However, some researchers have pointed out the deficiencies of expert engineering judgement as applied to common cause failures, and contend that analysis of such failures is a knowledge-based decision process and, therefore, is itself subject to error or uncertainty (Doerre 1987).

b) Problems with Applying CCF in Safety and Risk Analysis for Engineering Design

Problems with applying CCF in safety and risk analysis for engineering design assessment in the preliminary design phase can thus be reviewed.

These problems are summarised as (Hanks 1998):

- The lack of a suitable comprehensive database for CCF.
- Use of simple CCF models giving pessimistic results for redundancy systems.
- The assumption that similar components will be similarly affected.
- Errors in understanding the nature of CCF and applying the appropriate methodology.

Various alternative models that refine the beta factor method have thus been proposed, such as a binomial failure rate model that assumes that a component has a constant independent failure rate and a susceptibility to common cause shocks at a constant rate (NUREG/CF-1401 1980). This has been extended to include common cause shocks that do not necessarily result in catastrophic failure. A practical method of common cause failure modelling modifies the beta factor model to take account of the levels of redundancy and diversity in systems (Martin et al. 1987). It was previously noted that the simple beta model is pessimistic when applied to redundancy systems. This can also be the case when it is applied to a range of similar components even if they are installed in one system. To illustrate this problem,

an example is given based on a simplified high-pressure protection redundancy configuration relating to the high-integrity protection sub-system (HIPS) illustrated in Fig. 5.22. As indicated in Sect. 5.2.3.2, the function of the HIPS sub-system is to prevent a high-pressure surge passing through the process, thereby protecting the process equipment from exceeding its individual pressure ratings.

A schematic of the simplified configuration is given in Fig. 5.23. In this example, a possible source of CCF is the contamination of the upstream high-pressure line. In theory, all the regulators should be equally affected—however, much depends upon the design features of the main valves and their control systems. Contamination of the high-pressure line will affect the active control valve, A_1, in the operating stream. Whether it will affect the monitor valve M_1, and to what extent, depends on the way the control system functions. Both the regulators in the standby stream should be unaffected. In this example, there is a potential for CCF to occur but normal practice would be to assume that CCF applies equally to all the four identical valves—so, it will be seen that the result of any prediction would be pessimistic. Another problem in this case would be the total misunderstanding of how to apply CCF prediction methodology.

In Fig. 5.23, there are two control streams, one functioning as an operating stream with two identical regulators (pressure valves), M_1 and A_1, and the other functioning as a standby stream with two identical regulators, M_2 and A_2. Each stream's regulator configuration consists of a monitor valve, M_i, and an active control valve, A_i ($i = 1, 2$). The first regulator in the operating stream, the monitor valve M_1, is fully open in readiness to take over control should the pressure rise *above* a predetermined level due to failure of the active control valve A_1. The active control valve A_1 controls the outlet pressure. Similarly, the first regulator in the standby stream, M_2, is fully open and will function in a similar manner as valve M_1, should the standby stream be activated. The second regulator in the standby stream, A_2, is closed and will take over pressure regulation if either of the regulators in the operating stream fails to reduce the outlet pressure to *below* a predetermined level.

Common cause failures can arise from a wide range of potential problems, typically identified through factor tree charts and associated questions concerning the potential root causes of design integrity problems (as indicated in Sect. 5.2.1.2, and Figs. 5.6 through to 5.8).

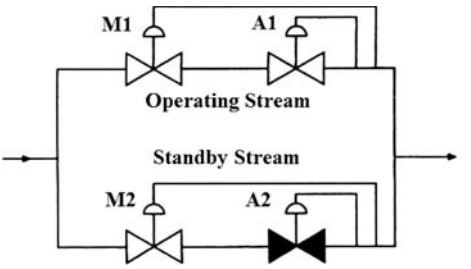


Fig. 5.23 Schematic of a simplified high-pressure protection system

Table 5.13 Analysis of valve data to determine CCF beta factor

Valve type	CCF beta factor
Ball, plug and gate valves	0.01–0.02
Relief valves, all types	0.05
Check and non-return valves	0.05
Cut-off valves	0.05
Large regulators/control valves	0.10–0.19
Small regulators/control valves	0.05–0.12
Actuators, all types	0.05–0.16

Minimising the effects of CCF is thus an on-going process, already from the early phases of engineering design to the in-service life. Any attempt to cut corners or costs will almost inevitably expose engineered installations to a higher level of CCF-induced failures, with resulting increased costs of failure maintenance, lost production and possible loss of life.

Design criteria databases do not usually include common cause failure data. One problem is that CCF data for a particular component can be specific to the application and, hence, require a whole series of design, operation and maintenance considerations for a particular process. Detailed analysis of valve data from a large database collected from maintenance and operational records has yielded useful information on the incidence of CCF (Hanks 1998). The data, summarised in Table 5.13, cover a wide range of valve types and applications, including actuators.

This quantification of the CCF beta value, indicating that a significant portion of the total failure rate of a component arises from common cause failures, has placed upper limit constraints on obtaining sufficiently high levels of reliability and safety in critical risk control circuits. As indicated previously, redundancy is one method in avoiding this problem, although the approach does lead to large increases in the *false alarm rate (FAR)*.

This is overcome, however, by utilising voting redundancy where several items are used in parallel and a selected amount are required to be in working order. The voting redundancy problem involves the simultaneous evaluation and selection of available components and a system-level design configuration that collectively meets all design constraints and, at the same time, optimises some objective function, usually system safety, reliability, or cost. In practice, though, each of these parameters may not be exactly known and there is some element of risk that the constraint will not actually be met or the objective function value may not be achieved.

5.2.4 Theoretical Overview of Safety and Risk Evaluation in Detail Design

Safety and risk evaluation determines safety risk and criticality values for each individual item of equipment at the *lower* systems levels of the systems breakdown structure (SBS). Safety and risk evaluation determines the causes and consequences of hazardous events that occur randomly, together with a determination of the frequencies with which these events occur over a specified period of time based on critical *component* failure rates. Safety and risk evaluation is considered in the *detail design* phase of the engineering design process, and includes basic concepts of modelling such as:

i. Point process event tree analysis in designing for safety.
ii. Cause-consequence analysis for safety systems design.
iii. Failure modes and safety effects evaluation.

5.2.4.1 Point Process Event Tree Analysis in Designing for Safety

The most extensive safety study to date is the US Nuclear Regulatory Commission's report "Reactor Safety Study" (NUREG-75/014 1975). In October 1975, the NRC issued the final results of a 3-year study of the risks from postulated accidents during the operation of nuclear power reactors of the type used in the USA. This report, known as the "Reactor Safety Study (RSS)", or by its NRC document number, WASH 1400, was the first comprehensive study that attempted to quantify a variety of risks associated with power reactor accidents. Since that time, about 40 reactors have been analysed using the same general methodology as WASH 1400 but with considerably improved computer codes and data.

The most recent and the most detailed of these studies has been the effort undertaken by the NRC to analyse five different reactors using the very latest methodology and experience data available. In June 1989, the second draft of this work, "Severe Accident Risks: An Assessment for Five U.S. Nuclear Power Plants" (NUREG 1150 1989), was issued for public comment. There is, however, widely held belief that the risks of severe nuclear accidents are small. This conclusion rests in part upon the probabilistic analysis used in these studies (NUREG/CR-0400 1978).

The analysis used in these studies provides a suitable example in order to better understand the application of point process event tree analysis in the evaluation of safety and risk in the detail design phase. The approach to safety evaluation, as researched in these two studies, considered the sources of the risk, its magnitude, design requirements, and risk determination through probabilistic safety evaluation (PSE). These points of approach, although very specific to the example, need to be briefly explained (Rasmussen 1989).

a) Determining the Source of Risk

During full power operation, a nuclear power reactor generates a large amount of radioactivity. Most of this radioactivity consists of fission products, resulting from the fission process, which are produced inside the reactor fuel. The fuel is uranium dioxide, a ceramic material that melts at about 5,000 °F. The fuel effectively contains the radioactive fission products unless it is heated to the melting point. At temperatures in this range, essentially all the gaseous forms of radioactivity will be released from the fuel. In addition, some of the more volatile forms of the solid fission products may be released as fine aerosols. If any of these forms were to be released into the atmosphere, they could be spread by prevailing winds.

b) Designing for Safety Requirements

Design requirements for safety in US nuclear plants mandate that the plants have systems to contain any radioactivity accidentally released from the fuel. The main system for accomplishing this is the containment building, an airtight structure that surrounds the reactor. In addition, all reactors have a system for removing aerosols from the containment atmosphere. In many reactors, this system consists of a water spray that can create the equivalent of a heavy rainstorm inside the containment building. Boiling water reactors (BWR) accomplish this function by passing released gases through a pool of water. The principal goal of the reactor safety philosophy is to prevent the accidental release of radioactivity. As a backup, systems are added that prevent the release of radioactivity to the atmosphere even if it were released from the fuel. Despite these efforts, one can always postulate ways in which these systems might fail to prevent the accidental release of radioactivity.

It is the task of probabilistic safety evaluation (PSE) to identify how this might happen, to determine how likely it is to happen and, finally, to determine the health effects and economic impacts of the radioactive releases upon the public.

c) Probabilistic Safety Evaluation (PSE)

- The first step in a PSE analysis begins by developing the causes and likelihood of heating the fuel to its melting point due to either external causes (earthquakes, floods, tornadoes, etc.) or internal causes. This analysis involves developing a logical relationship between the failures of plant components and operators, and the failure of system safety functions. The result of this analysis is an estimate of the probability of accidentally melting the fuel, a condition often called ‘core melt’. Of the plants analysed thus far, most have an estimated likelihood of core melt of between 1 in 10,000 and 1 in 100,000 per plant year.
- The second step in a PSE analysis is to determine the type and amount of radioactivity that might be released in the different accidents identified. These fractions of the various types of radioactivity released are called the ‘source terms’ for

the accident. The values from WASH 1400 are in most cases significantly larger than those from NUREG 1150. The lower values of NUREG 1150 are the result of new information gained from major research in the USA, Japan and Western Europe. These experiments, and the measurements at Three Mile Island confirm that the values used in WASH 1400 are too high.

- The final step in a PSE analysis is to calculate the effects of any radioactivity released in the accident. Sophisticated computer models have been developed to do this calculation. These models require input of the source terms, the population density around the site, and weather data for recent years from the plant site. The code then calculates thousands of cases to generate curves that give the magnitude of given risks versus their probabilities. The results of the calculations are in the form of fatality curves. The curves generally give the frequency in units per reactor year for events of a given size, and have a wide range of consequences, from quite small at high frequencies to quite large at very low frequencies.
- Curves of this shape are typical of all accidents where a number of factors affect the magnitude of the event. In the case of catastrophic accidents, clearly this refers to accidents of low probability near the high-consequence end of the scale. These extreme accidents come about only if the various factors affecting the magnitude of the consequences are all in their worst states. Thus, for example, the core must melt, then the containment must fail above ground level, the wind must be blowing towards an area of relatively high population density, inversion conditions must prevail, and civil protection efforts must fail to be effective.

Criticism of the Reactor Safety Study pointed to inadequacies in the statistical methodology, particularly the uncritical use of the log-normal distribution to derive probability estimates for the failure of individual nuclear safety systems (NUREG/CR-0400 1978). There is an inherent weakness to the approach, in that there is no way of being sure that a critical initiating event has not been overlooked. The logic event tree consists of the initiating event and the success or failure response of each of the applicable engineered safety features. After identifying the accident sequence, the probability of occurrence of each engineered safety system in the sequence must be evaluated. As no empirical data are available on which to base estimates of system failure rates, it is necessary to use techniques that generate system failure rates from comparative estimates of failures of similar equipment. The extended use of event trees to derive probability estimates for both the failure of individual nuclear safety systems, as well as the accident sequences was developed by the US Department of Defense and the US National Aeronautics and Space Authority (NASA; NUREG 1150 1989).

With the identification of potential accidents and the quantification of their probability and magnitude, accident sequences are identified by means of logic diagrams—in this case, logic event trees. The starting point for the development of these logic event trees is the identification of the event that initiates a potential accident (due to a catastrophic failure event) or potential incident (due to a critical failure event). A typical initiating event for the nuclear reactor safety example would be a pipe break that results in a loss of coolant. Initiating events are usually identified using technical information and engineering judgment, similar to an

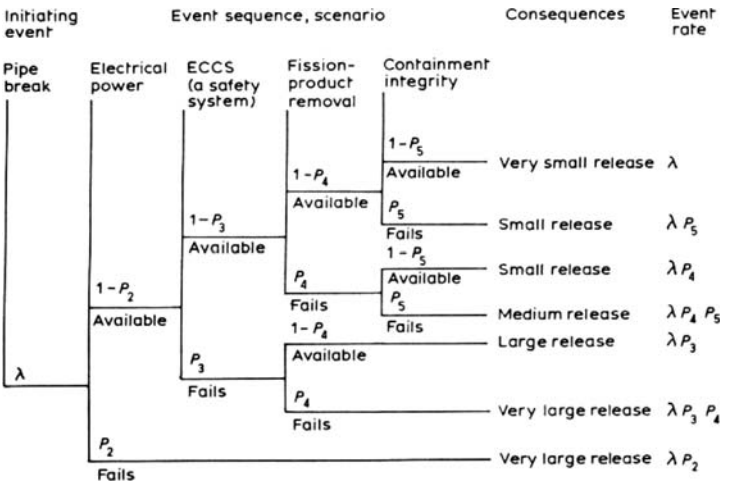


Fig. 5.24 Typical logic event tree for nuclear reactor safety (NUREG-751014 1975)

integrated information technology (IIT) program considered in Section 3.3.3.4 and in Fig. 3.41.

Figure 5.24 shows a typical logic event tree with an initiating event of a pipe break in a nuclear reactor coolant line, with probability of occurrence of λ. The logic event tree is simplified in that only seven out of 2^4 possibilities need to be considered—for example, if electric power fails with an event rate of λP_2, then none of the engineering safety features will function. The output of the logic event tree is the release category consequences with their event rates. Since the probability of occurrence is small (i.e. equivalent to the concept of a *rare event*), the probabilities are approximated by omitting all $1 - P_i$ terms.

d) Point Process Consequence Analysis

The basic methodology of the Reactor Safety Study used an approach of determining a demand failure rate. This can briefly be explained as the control of the rate of reaction in an atomic power plant by the insertion of control rods. The times at which control is needed is termed the transient demand, and was assumed to occur in an operating time equivalent to a Poisson process. When a transient demand occurs, the conditional probability that the safety system does not function, resulting in the consequence of an accident, was determined. Based on the Reactor Safety Study, a method for evaluating consequences as a result of safety system failure in a catastrophic-event process such as a nuclear reactor has been researched (Thompson 1988).

Suppose the events initiating accident sequences (as in Fig. 5.24) occur in time according to a stochastic point process with an event rate of $\mu(t)$. Furthermore, let $N(t)$ denote the number of events up to time t, and T_i $(i = 1, 2, 3, \ldots, k)$ denote

the time at which the *i*th initiating event occurs. Suppose further that the *i*th initiating event yields the consequence C_i. Assume that C_i is a non-negative random variable with failure distribution function $P(C_i \leq c) = F(c)$, and survival function $P(C_i \geq c) = F'(c)$. The consequences can be assumed to be identically distributed and independent of one another, with the understanding that there are several kinds of risks, each with its own initiating event rate and consequence distribution. Finally, the evolution of consequences has been assumed to follow a point process. Actually, the consequences of many accidents and incidents are difficult to express numerically and most are vector valued in time. The basic methodology of the Reactor Safety Study in dealing with this problem was to conduct a separate study for each type of consequence, and to present the risk in terms of an event rate against a consequence in the form of a *risk curve*, as illustrated in Fig. 5.25. In mathematical terms, if $\mu_k(t)$ is the event rate at time t of consequences exceeding k, the critical number of consequences, then the risk curve is a graph for fixed t of $\mu_k(t)$ versus k. The event rate of consequences that exceed k is related to the process of initiating events and the distribution of consequences

$$\mu_k(t) = [1 - F(k)]\mu(t) \tag{5.66}$$

where:

$F(k)$ is the failure distribution function of C_i.

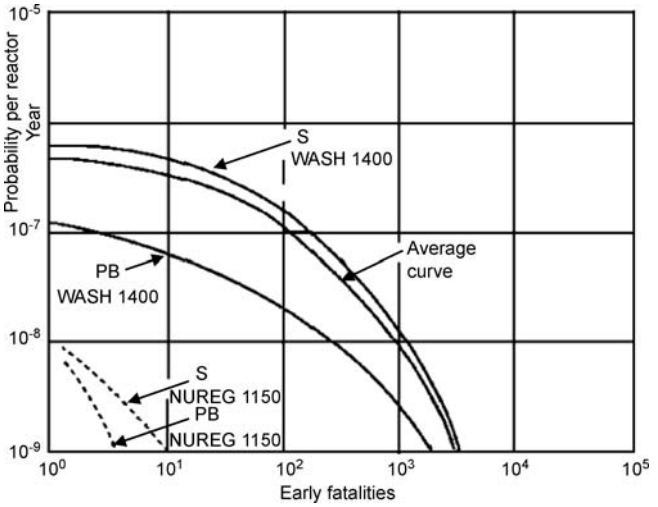


Fig. 5.25 Risk curves from nuclear safety study (NUREG 1150 1989) Appendix VI WASH 1400: c.d.f. for early fatalities

Different process systems designs have different consequence sequences. The consequence sequences, S, of a particular process, P, over a time period, t, can be expressed as

$$S_{\mathrm{P}}(t) = \begin{cases} 0 & N(t) = 0 \\ C_1 + C_2 + C_3 + \ldots C_{N(t)} & N(t) > 0 \end{cases} \tag{5.67}$$

Characteristics of $S_{\mathrm{P}}(t)$ are determined by $N(t)$ and the distribution of the consequences C_i, where the sequence of consequences constitutes a point process. Of specific interest is to determine the catastrophic event having the greatest consequence when one accident is too much in the sequence of consequences. This is done by defining an expression for S'_{p} in the catastrophic case

$$S_{\mathrm{P}}(t) = \begin{cases} 0 & N(t) = 0 \\ \max[C_1 + C_2 + C_3 + \ldots C_{N(t)}] & N(t) > 0 \end{cases} \tag{5.68}$$

Probability $S'_{\mathrm{p}}(t)$ being less than k, the critical number of consequences is given by

$$P(S'_{\mathrm{p}}(t) \leq k) = P' \tag{5.69}$$

where:

$$P' = \sum_{n=0}^{\infty} P[C_i \leq k; \quad i = 1, 2, 3, \ldots, n | N(t) = n]$$

If C_i is a non-negative random variable with the failure distribution function $P(C_i \leq c) = F(c)$, then

$$P(S'_{\mathrm{p}}(t) \leq k) = \sum_{n=0}^{\infty} [F(k)]^n \cdot P[N(t) = n] \tag{5.70}$$

$$= \psi_t[F(k)] \tag{5.71}$$

where:

$\psi_t =$ probability generating function of $N(t)$ (i.e. Bernoulli transform)

and if C_i is a non-negative random variable with the survival function $P(C_i \geq c) = F'(c)$, then

$$P(S'_{\mathrm{p}}(t) > k) = 1 - \psi_t[F(k)] \leq [1 - F(k) \cdot E\, N(t)] \tag{5.72}$$

Thus, for consequences exceeding k, the critical number of consequences (or the 'cut-off value' between acceptable and unacceptable consequences), the probability of the occurrence of an unacceptable consequence within time t will be less than $F'(k)E\, N(t)$, where $F'(k)$ is the survival function $P(C_i > k)$, and $E\, N(t)$ is the expected value or mean of $N(t)$, the number of events on $(0, t]$.

If system failure is now identified with obtaining an unacceptable consequence, then $F'(k)$ is the demand failure rate (such as the demand to control the rate of reaction in an atomic power plant by the insertion of control rods). This demand failure rate yields an upper bound for the probability of failure. Since k is the unacceptable critical number of consequences, the probability of a consequence exceeding

that value must be as small as possible—that is, $F'(k)$ will be near 0 with an upper bound when $F(k)$ is near 1.

The expected maximum consequence can be expressed as

$$EC'(t) = \int_0^\infty \{1 - \psi_t[F(k)]\}\,\mathrm{d}k \tag{5.73}$$

$$EC'(t) = \sum_{n=0}^{\infty} \int_0^\infty \{1 - [F(k)]^n\}\,\mathrm{d}k \cdot P[N(t) = n] \tag{5.74}$$

From Eqs. (5.72) and (5.74) we get:

$$EC(t) \cdot P[N(t) \geq 1] \leq EC'(t) \leq EC(t) \cdot N(t)$$

where:

$EC(t)$ = the expected value of consequence C in period t
$EC'(t)$ = the expected value of consequence C' in period t
$P[N(t) \geq 1]$ = the probability that the number of events ≥ 1.

The expected time to the first critical event with unacceptable consequence is given as

$$EV_k = \int_0^\infty \psi_t[F(k)]\,\mathrm{d}t \tag{5.75}$$

$$EV_k = \sum_{n=1}^{\infty} ET_n[F(k)]^n[1 - F(k)] \tag{5.76}$$

where:

T_n = time of occurrence of the nth initiating event.

The *probability generating function (p.g.f.)*, or Bernoulli transform ψ_t, needs to be defined in greater detail: Thus, given a random variable $N(t)$, its generating function $\psi_t(z)$ is expressed as

$$\psi_t(z) = \sum_{n=1}^{\infty} z^n P[N(t) = n] \tag{5.77}$$

$$\psi_t(z) = Ez^{N(t)} \tag{5.78}$$

$\psi_t(z)$ is a function in terms of z with the following properties:

- The p.g.f. is determined by and also determines C
- $\psi_t'(1)$ is the expectation of $N(t)$
- $\psi_t''(1)$ is the expectation of $N(t) \cdot [N(t) - 1]$.

Probability generating functions also provide for addition of independent random variables. For example, if $N(t)$ and $C(t)$ are independent, then the p.g.f. of

$\{N(t)+C(t)\}$ is obtained by multiplying the p.g.f.s of the random variables together

$$Ez^{N(t)} \cdot Ez^{C(t)} = Ez^{N(t)+C(t)} \tag{5.79}$$

where $N(t)$ and $C(t)$ are independent.

5.2.4.2 Cause-Consequence Analysis for Safety Systems Design

Cause-consequence analysis for safety systems design is fundamentally a *combinatorial symbolic logic technique*, utilising the symbolic logic of *fault-tree analysis (FTA)*, *reliability block diagramming (RBD)* and *event tree analysis (ETA)*. Each of these techniques has unique advantages and disadvantages. In most complex safety systems designs, it is beneficial to construct a model using one technique, then transform that model into the domain of another technique to exploit the advantages of both. Fault trees are generated in the failure domain, reliability diagrams are generated in the success domain, and event trees are generated in both the success and failure domains.

Methodology to transform any one of the above models into the other two, by translating equivalent logic from the success to failure or failure to success domains, is considered later. Probabilities are propagated throughout the logic models to determine the probability that a system will fail, i.e. its risk, or the probability that a system will operate successfully, i.e. its reliability. Probability data may be derived from available empirical data or, if quantitative data are not available, then subjective probability estimates may be used.

Cause-consequence analysis for safety systems design explores the system's responses to an initiating deviation from predetermined norms (such as the limits of safe operating parameters), and enables evaluation of the probabilities of unfavourable outcomes at each of a number of mutually exclusive loss levels, depending upon the extent of deviation from these norms. The deviation beyond a set limit is designated an event.

The analysis then begins with an initiating event and performs a forward (bottom-up) analysis using ETA. This technique provides data similar to those available with conventional event tree analysis; however, it affords two advantages over the conventional event tree—time sequencing of events is better portrayed, and discrete, staged levels of outcome are analysed. The cause portion of this technique is the safety system response to an undesired process state or condition. This process state is represented as a fault-tree TOP event and is normally, but not always, quantified by its probability of occurrence. The consequence portion of this technique yields a display of potential outcomes representing incremental levels of success or failure of the safety system. Each increment has an associated level of assumed or calculated probability, based on variations of responses of the safety system to the various process states or conditions. The cause has an associated probability, and each consequence has an associated severity and probability (NASA 1359 1994).

Cause-consequence analysis for safety systems design is particularly useful in analysing command-start and command-stop protective devices, emergency response systems, and engineered safety features. Cause-consequence analysis is fundamentally useful in evaluating design decision options concerning the effects and/or benefits of sub-tiered redundant or diverse countermeasures for safety systems design. This technique may be used to compliment a failure modes and effects analysis (FMEA) or, more specifically, a failure modes and safety effects (FMSE) analysis, otherwise known as probabilistic risk analysis (PRA).

a) Fault Tree, Reliability Block Diagram, and Event Tree Transformations

Fault trees, reliability block diagrams (RBDs) and event trees are all symbolic logic models. Fault trees are generated in the failure domain, reliability diagrams are generated in the success domain, and event trees are generated in the success and failure domains. These techniques transform any one of the above models into the other two by translating equivalent logic from the success to failure or failure to success domain. Fault trees offer comprehensive qualitative or quantitative analysis. RBDs offer a simplistic method to represent system logic, and event trees enable systems evaluation in both the success and failure domains. Prior to considering the methods for transforming a fault tree, RBD or event tree into either of the other two logic models, it is essential to first review reliability block diagrams (RBDs):

A *reliability block diagram (RBD)* is a deductive, top-down analysis, symbolic logic model, used to define the path from effect to cause, and generated in the success domain. Each RBD has an input and an output and flows left to right from the input to the output. Blocks may depict failure events or element functions within a system, though most RBDs typically depict system element functions only. A system element can be a sub-system, assembly, sub-assembly, component or part. Simple RBDs are constructed of series, parallel, or combinations of series and parallel elements, as indicated in Fig. 5.26 (NASA 1359 1994).

An RBD may contain a combination of series and parallel branches where each block represents an event or system element function. These blocks are connected in series if all elements must operate for the system to operate successfully, or they are connected in parallel if only one element needs to operate for the system to operate successfully.

Reliability is the probability of successful operation during a defined time interval and, conversely, unreliability is the probability of failure during a defined time interval. In a safety analysis context, RBDs indicate system reliability or unreliability, where each block may represent a system element function (operates successfully) or a failure event. Each element of a block diagram is assumed to function or to fail independently of the other elements. The overall system reliability can thus be determined from the relationships between element reliability and system reliability for series and parallel systems.

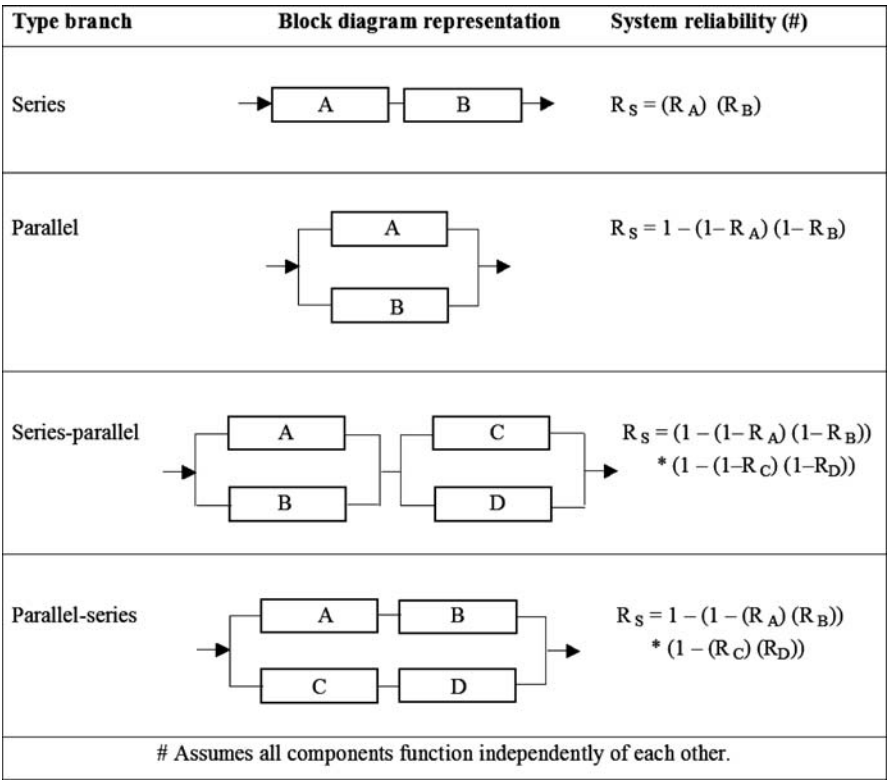


Fig. 5.26 Simple RBD construction

The relationships between element reliability and system reliability for series and parallel systems can be mathematically expressed as

$$\text{Series } R_S = \prod_i^n R_i = R_1 * R_2 * R_3 \ldots R_n \tag{5.80}$$

$$\text{Parallel } R_S = 1 - \prod_i^n (1 - R_i)$$

$$\text{Parallel } R_S = [1 - (1 - R_1)(1 - R_2)(1 - R_3) \ldots (1 - R_n)]$$

where:

R_S = system reliability
R_i = system element reliability
n = number of system elements that function independently.

Not all systems can be modelled with simple RBDs. Some complex systems cannot be modelled with true series and parallel branches. These systems must be modelled with a complex RBD. Such an RBD is presented in Fig. 5.27. In this example, if

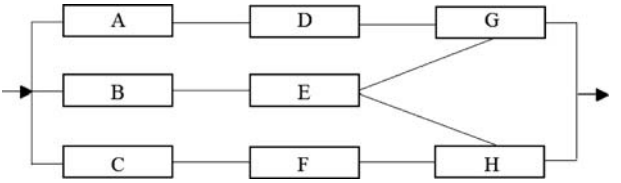


Fig. 5.27 Layout of a complex RBD (NASA 1359 1994)

element E fails, then paths B, E, G and B, E, H are not success paths; thus, this is not a true series or parallel arrangement.

An RBD enables evaluation of various potential design configurations. The required element reliability levels can be determined to achieve the desired system reliability. An RBD can also be used to identify design configuration elements in symbolic logic as a precursor to performing an FTA. The procedures to generate a simple RBD are as follows:

(1) Define the system into its SBS from the available functional diagram of the system.
(2) Construct a block diagram using the convention illustrated in Fig. 5.26.
(3) Calculate system reliability bands, R_{SL} (low) to R_{SH} (high), from each element's reliability band, $R_{i\mathrm{L}}$ (low) to $R_{i\mathrm{H}}$ (high), in the following manner:

a. For series systems with n elements that are to function independently

$$R_{\mathrm{SL}} = \prod_{i}^{n} R_{i\mathrm{L}} = R_{1\mathrm{L}} \cdot R_{2\mathrm{L}} \cdot R_{3\mathrm{L}} \cdot \ldots \cdot R_{n\mathrm{L}} \tag{5.81}$$

$$R_{\mathrm{SH}} = \prod_{i}^{n} R_{i\mathrm{H}} = R_{1\mathrm{H}} \cdot R_{2\mathrm{H}} \cdot R_{3\mathrm{H}} \cdot \ldots \cdot R_{n\mathrm{H}} \,.$$

b. For parallel systems with n elements that are to function independently

$$R_{\mathrm{SL}} = 1 - \prod_{i}^{n} (1 - R_{i\mathrm{L}}) \tag{5.82}$$

$$R_{\mathrm{SL}} = [1 - (1 - R_{1\mathrm{L}})(1 - R_{2\mathrm{L}})(1 - R_{3\mathrm{L}}) \ldots (1 - R_{n\mathrm{L}})]$$

$$R_{\mathrm{SH}} = 1 - \prod_{i}^{n} (1 - R_{i\mathrm{H}})$$

$$R_{\mathrm{SH}} = [1 - (1 - R_{1\mathrm{H}})(1 - R_{2\mathrm{H}})(1 - R_{3\mathrm{H}}) \ldots (1 - R_{n\mathrm{H}})]$$

c. For series-parallel systems, first determine the reliability for each parallel branch using the equations in step 3b. Then treat each parallel branch as an element in a series branch and determine the system reliability by using the equations in step 3a.
d. For parallel-series systems, first determine the reliability for each series branch using the equations in step 3a. Then treat each series branch as an

element in a parallel branch and determine the system reliability by using the equations in step 3b.

For systems that are composed of the four arrangements given above, the reliabilities for the simplest branches are first determined, which then become branches within the remaining block diagram. The reliability for the new branches are then determined. This process is continued until one of the above four basic arrangements remains, from which system reliability is calculated.

As an example, consider a high-pressure process with a high-integrity protection system (HIPS) containing two sub-systems designated S_1 and S_2. Sub-system S_1 has three sensor components and at least one of the three must function successfully for the sub-system to operate. Sub-system S_2 is an essential instrumentation path for the protection system. Sub-system S_2 has three components that all need to function successfully for the sub-system to operate. The estimated reliability band for each component is given in Table 5.14.

The components for sub-system S_1 are in a parallel branch with the components of sub-system S_2. In addition, the components for sub-system S_1 form a series branch, and the components for sub-system S_2 form a parallel branch. An RBD for the system is illustrated in Fig. 5.28.

Sub-system S_1 reliability
Low band value: $R_{S_1L} = 1-(1-0.80)(1-0.80)(1-0.80) = 0.992$
High band value: $R_{S_1H} = 1-(1-0.85)(1-0.85)(1-0.85) = 0.997$

Table 5.14 Sub-system component reliability bands

Component	Low	High
1 A	0.80	0.85
1 B	0.80	0.85
1 C	0.80	0.85
2 D	0.90	0.95
2 E	0.90	0.95
2 F	0.90	0.95

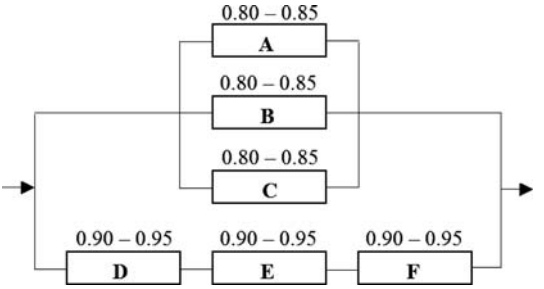


Fig. 5.28 Example RBD

Sub-system S_2 reliability
Low band value: $R_{S_2L} = (0.90)(0.90)(0.90) = 0.729$
High band value: $R_{S_2H} = (0.95)(0.95)(0.95) = 0.857$

System reliability
Low band value: $R_{SL} = 1 - (1 - 0.992)(1 - 0.729) = 0.998$
High band value: $R_{SH} = 1 - (1 - 0.997)(1 - 0.857) = 0.999$

The reliability band for the combined system is 0.998 to 0.999. This example is of particular interest in that the configuration for the HIPS produces an overall reliability range that is higher than any of the sub-system reliabilities. The reliability values for the parallel sub-system S_1 are higher than the reliability values for the series sub-system S_2, implying that the sensors configuration of the HIPS has higher priority than does the instrumentation path.

The application of an RBD in safety systems design provides several advantages in that it allows evaluation of design concepts when design changes can still be incorporated. Furthermore, it tends to be easier to visualise than other logic models, such as fault trees, because blocks representing elements in an RBD can be arranged in a manner representing how these elements function in the system. Since RBDs are easy to visualise, they can be generated prior to conducting FTA, and transformed into a fault tree.

The methods for transforming a fault tree, RBD or event tree into either of the other two logic models are as follows (NASA 1359 1994), starting with the RBD to fault tree transformation shown in Fig. 5.29.

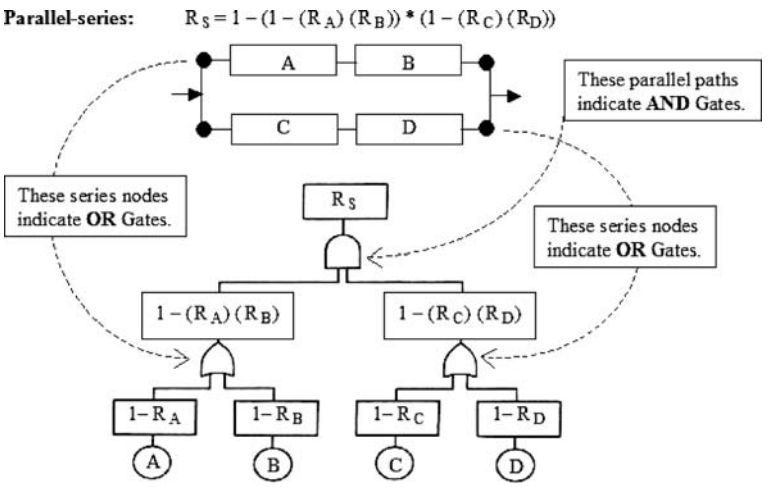


Fig. 5.29 RBD to fault tree transformation

b) RBD to Fault Tree Transformation

A fault tree represents system functions that, if they fail, produce a TOP event fault in place of a success event, which the reliability block path indicates. The series nodes of an RBD denote an OR gate beneath the TOP event of a fault tree. The parallel paths in an RBD denote the AND gate for redundant component functions in a fault tree. The reliability diagram can thus be relatively easily transformed into a fault tree, as shown in Fig. 5.29.

c) Fault Tree to RBD Transformation

An RBD represents system component functions that produce success in place of a TOP fault event, if these functions prevail. A fault tree can be transformed into a reliability diagram, as illustrated in Fig. 5.30.

d) RBD and Fault Tree to Event Tree Transformation

An event tree represents path sets in the success branches of the tree and all the cut sets in the failure branches of the tree. Therefore, if the path sets and cut sets of a system are known for the TOP event of a fault tree, then an event tree can be

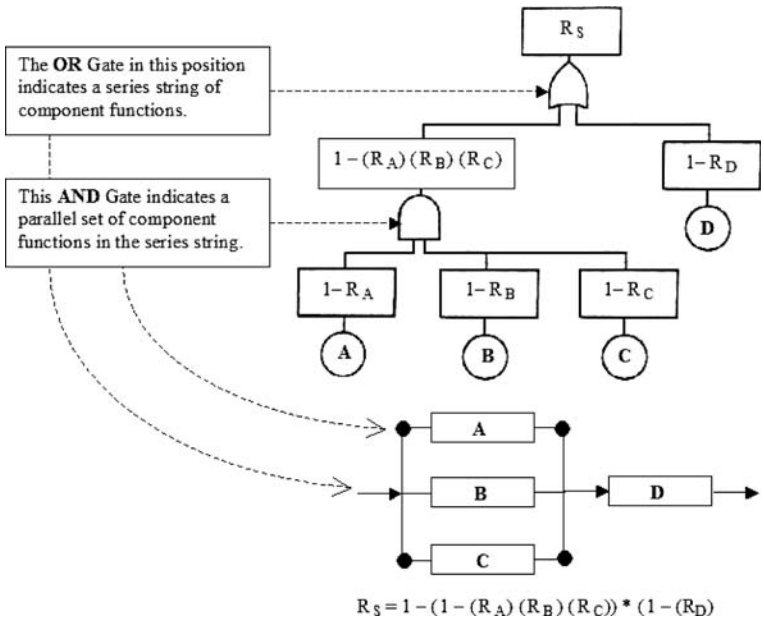


Fig. 5.30 Fault tree to RBD transformation

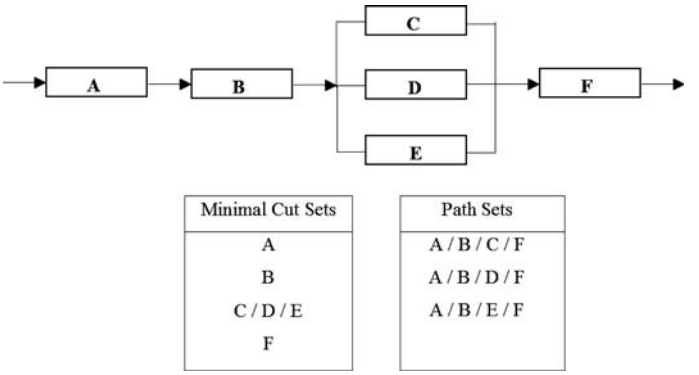


Fig. 5.31 Cut sets and path sets from a complex RBD

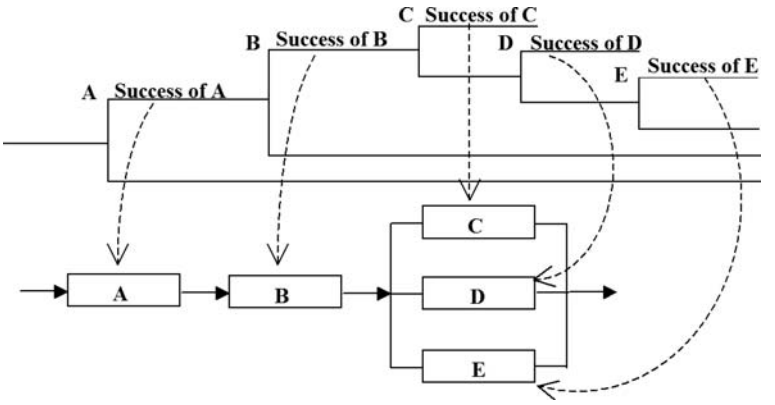


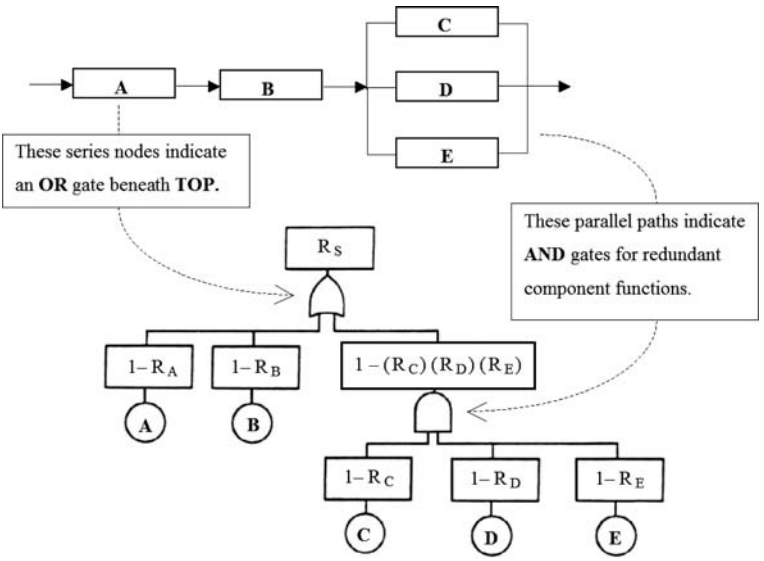
Fig. 5.32 Transform of an event tree into an RBD

constructed. Cut sets and path sets may be obtained from a reliability diagram as shown in Fig. 5.31 (cf. Fig. 5.32).

e) Event Tree to RBD and Fault Tree Transformation

An event tree represents path sets in the success branches of the tree and all the cut sets in the failure branches of the tree. To transform an event tree into an RBD, the process is reversed as illustrated in Fig. 5.32. Once the RBD is formed, a fault tree can be developed as illustrated in Fig. 5.33.

These techniques allow for weaknesses of any one of the analysis techniques to be overcome by transforming a system model into an equivalent logic model as another analysis technique. For example, a complex system that may be hard to model as a fault tree might be easily modelled with an RBD. Then, the RBD can



$$Rs = 1-(1-R_A)*(1-R_B)*(1-(R_C)(R_D)(R_E))$$

Fig. 5.33 Transform of an RBD to a fault tree

be transformed into a fault tree, and extensive quantitative or pseudo-quantitative (partially qualitative and quantitative) analysis can be performed.

However, these techniques do possess some limitations, such as the following:

- No new information concerning the system is obtained and the models are only as good as the models being transformed.
- The cut sets and path sets required to perform these transformations for large complex systems may require extensive computer resources to determine.

f) Structuring the Cause-Consequence Diagram

Previously in Sect. 5.2.1.4, a four-stage procedure to construct and analyse a cause-consequence diagram was given as:

- Step 1) Component failure event ordering.
- Step 2) Cause-consequence diagram construction.
- Step 3) Reduction.
- Step 4) System failure quantification.

If this procedure is to be considered as a generally applicable approach, it must be capable of dealing with the events that occur in more than one fault-tree structure attached to the decision boxes in any sequence path. It can be shown that the cause-consequence diagram method can deal with repeated events in a more efficient way

than that used for fault-tree analysis (FTA). Using the cause-consequence diagram method, there is no need to obtain the Boolean expression of the top event and then manipulate it to produce a minimal form prior to analysis.

The cause-consequence method deals with sequences of events that either occur (fail) or do not occur (work). The probability of a particular outcome is obtained by summation of the probability of all paths that lead to the outcome. Summation of the probabilities of the mutually exclusive paths results in the development of the reduced form that would be obtained from the fault tree following Boolean reduction. An algorithm has been developed that can trace through a cause-consequence diagram, and identify and extract any repeated basic events in more than one fault-tree structure on the same sequence path. Certain procedural steps are used in this extraction algorithm (Ridley et al. 1996).

Procedural steps used in an extraction algorithm to identify and extract any repeated basic events in more than one fault-tree structure on the same sequence path:

- Step 1) Identify the fault-tree structures in the path under inspection.
- Step 2) Each fault tree in a path is modularised and the independent sub-trees identified.
- Step 3) Each independent sub-tree for each fault-tree diagram is compared to the others and, following identification of common sub-trees or individual basic events, the cause-consequence diagram is modified.
- Step 4) The cause-consequence diagram is modified using the following rules:

  a. Following the identification of a common sub-tree or basic event, the common element is extracted and set as a new decision box at the highest point in the cause-consequence diagram that has all dependencies below it.
  b. The cause-consequence diagram is then duplicated on each branch.
  c. Having developed a decision box for the common sub-tree or basic event, the decision boxes that contained the common event prior to extraction require modification. The common event(s) are set to 1 (TRUE) in the fault trees following the NO outlet branch from the new decision box, as this indicates failure, and to 0 (FALSE) in the fault trees following the YES branch to signify that the common event(s) works.
  d. After extraction of the common sub-tree or basic event, each fault tree that has been modified requires reorganisation. Each fault tree containing the extracted Boolean variable is inspected and the fault trees modified by setting the Boolean variable to represent the path taken in the cause-consequence diagram.
  e. The cause-consequence diagram is then reduced to a minimal form by removing any redundant decision boxes that have been identified. This procedure is repeated until all sequence paths have been inspected and no repeated sub-trees or basic events discovered.

As an example, the technique is applied to the simple high-pressure protection system depicted in Fig. 5.34. The basic functions of the components of the system are shown in Table 5.15. The overall function of the protection system is to prevent

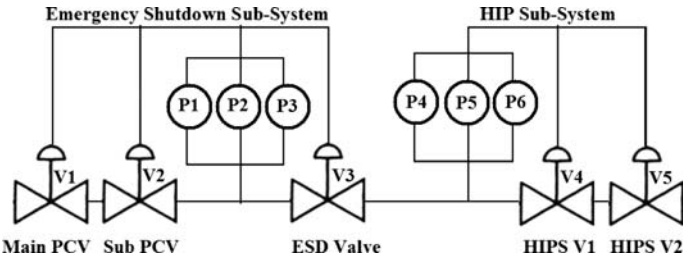


Fig. 5.34 High-integrity protection system (HIPS)

Table 5.15 Component functions for HIPS system

Component	Code	Function	Failure modes	λ and mean repair time	Maint. interval
Main PCV	V1	Stop high-pressure surge passing through system	Valve fails open: PCV-M	1.14×10^{-5}, 36.0	4,360
Sub-PCV	V2	Stop high-pressure surge passing through system	Valve fails open: PCV-S	1.14×10^{-5}, 36.0	4,360
ESD valve	V3	Stop high-pressure surge passing through system	Valve fails open: V-ESD	5.44×10^{-6}, 36.0	4,360
HIPS1	V4	Stop high-pressure surge passing through system	Valve fails open: VH1	5.44×10^{-6}, 36.0	4,360
HIPS2	V5	Stop high-pressure surge passing through system	Valve fails open: VH2	5.44×10^{-6}, 36.0	4,360
Solenoid	Sol	Supply power to valves	Fails energised: PCVs M, S, and ESD, and SH1, SH2	5.00×10^{-6}, 36.0	4,360
Relay contacts	RC	Supply power to solenoids (2 per solenoid)	Fails closed: R1–R10	0.23×10^{-6}, 36.0	4,360
Pressure sensors	Pr S	Indicate the level of pressure to the computer	Fails to record actual pressure: P1–P6	1.50×10^{-6}, 36.0	4,360
DCS	DCS	Reads information sent from pressure sensors and acts to close values	Fails to read or act on information	1.00×10^{-5}, 36.0	4,360

a high-pressure surge originating from process circulation pumps, to protect equipment located downstream of the process.

The first level of protection is the emergency shutdown (ESD) sub-system. This comprises three pressure sensors (P1, P2, P3), for which two out of three must indicate a high pressure to cause a trip. Two pressure control valves (PCVs), a main

PCV, a subsidiary PCV, and an emergency shutdown (ESD) valve (V1, V2, V3) activate to trip.

If a high-pressure surge is detected, the ESD sub-system acts to close the main PCV, the sub-PCV and the ESD valve. To provide an additional level of protection, a second sub-system is included, the high-integrity protection sub-system (HIPS).

This sub-system also comprises three pressure sensors (P3, P4, P5), for which two out of three cause a trip, and two isolation valves labelled HIPS1 and HIPS2 (V4, V5). The HIPS works in a manner identical to that of the ESD but has independent pressure sensors. These pressure sensors feed information for each sub-system into a common distributed control system (DCS).

The cause-consequence diagram is constructed following the rules given in Subsection f) above, including component failure event ordering, cause-consequence structure, reduction, and system failure quantification.

g) Event Ordering and Cause-Consequence Diagram Construction

The ordering is based on the action of components that could perform the task required by the system, i.e. main valve, subsidiary valve, ESD valve, HIPS1 valve and HIPS2 valve. The cause-consequence diagram is constructed by considering the

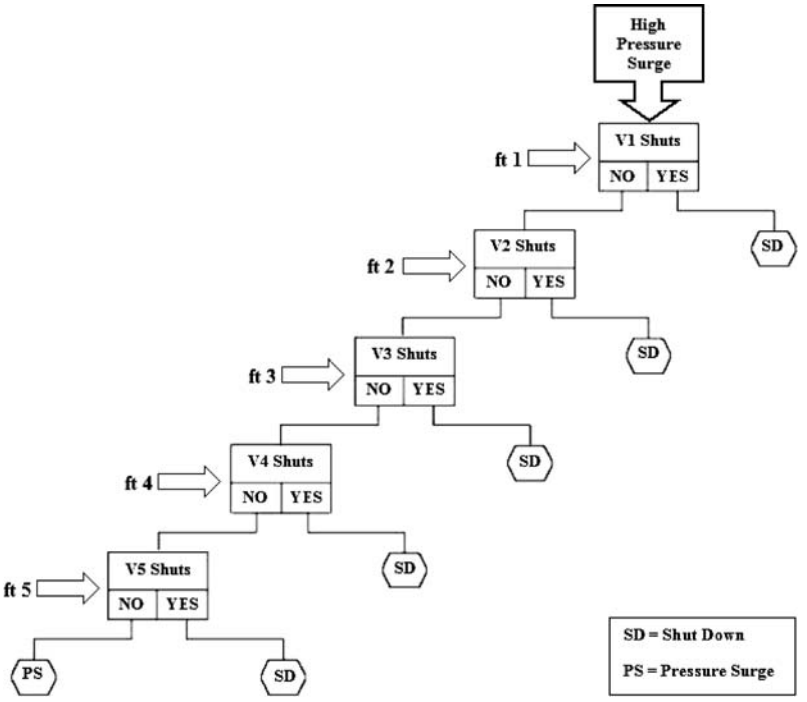


Fig. 5.35 Cause-consequence diagram for HIPS system (Ridley et al. 1996)

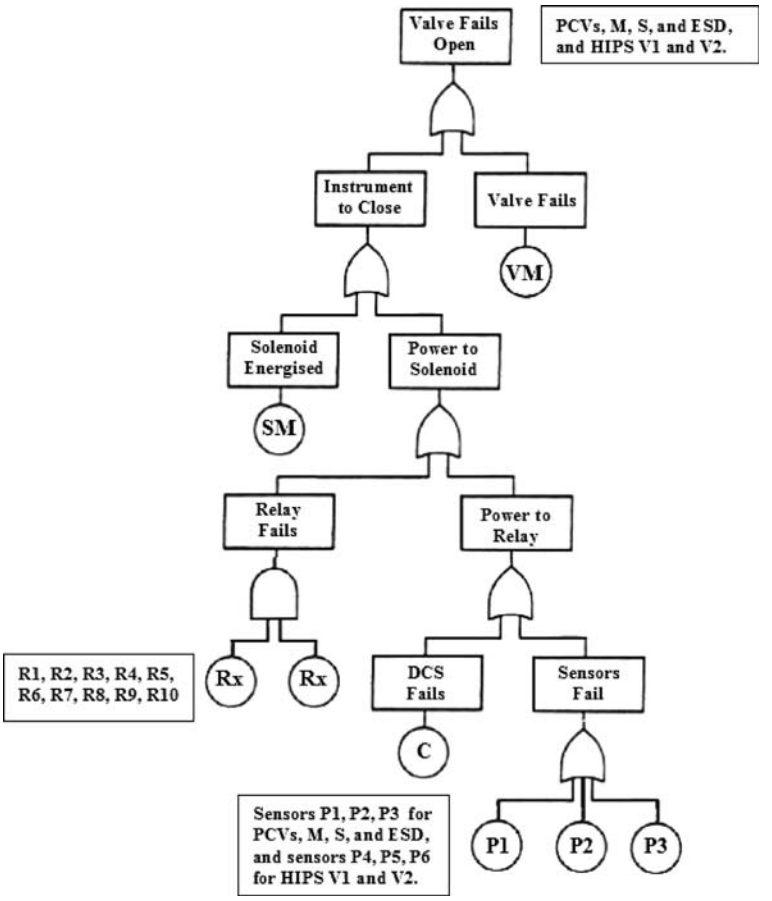


Fig. 5.36 Combination fault trees for cause-consequence diagram

functionality of each valve and their effect on the system. Following the removal of all redundant decision boxes, the minimal cause-consequence structure can be developed as indicated in Fig. 5.35. The combination fault trees developed for each decision box are illustrated in Fig. 5.36.

Following the construction of the cause-consequence diagram, each sequence path is inspected and any common independent sub-trees or basic events are identified. The first sequence path inspected in the HIPS system reveals that a common sub-module is present in ft1, ft2 and ft3, namely the failure of the pressure sensors P1, P2 and P3 respectively.

Extraction of this common sub-module, namely the failure of the pressure sensors P1, P2 and P3, results in a modified cause-consequence diagram depicted in Fig. 5.37. The cause-consequence diagram is reduced to a minimal form by removing any redundant decision boxes that have been identified. From the new version of the cause-consequence diagram, all sequence paths are investigated and modified

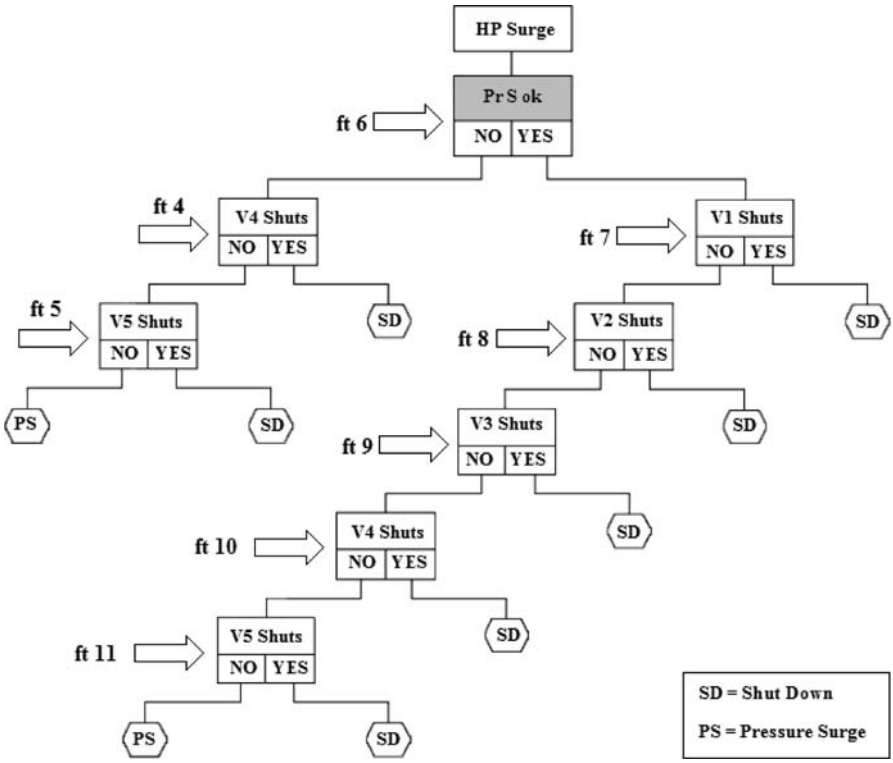


Fig. 5.37 Modified cause-consequence diagram for HIPS system (Ridley et al. 1996)

accordingly, using the rules outlined previously in Sub-section f). This procedure is repeated until all sequence paths have been inspected and no repeated sub-trees or basic events discovered.

The corresponding combination fault trees developed for the modified cause-consequence diagram for the HIPS system in Fig. 5.37—specifically, for 'valve fails open' (PCVs, M, S and ESD), as well as for 'sensors fail' (HIPS V1 and V2)—are given in Fig. 5.38.

The final cause-consequence diagram with corresponding combined fault trees can now be constructed as illustrated in Fig. 5.39.

The corresponding combined fault trees shown in Fig. 5.40 are now in a form where each path contains independent events in the decision boxes and can be easily quantified.

The probability of a high-pressure surge could now be obtained by summing the probabilities of ending in the consequence PS, which was reached via five mutually exclusive paths.

Therefore

$$\text{Probability (High Pressure)} = \sum_{i=1}^{n} P(\text{Path}\, i) \tag{5.83}$$

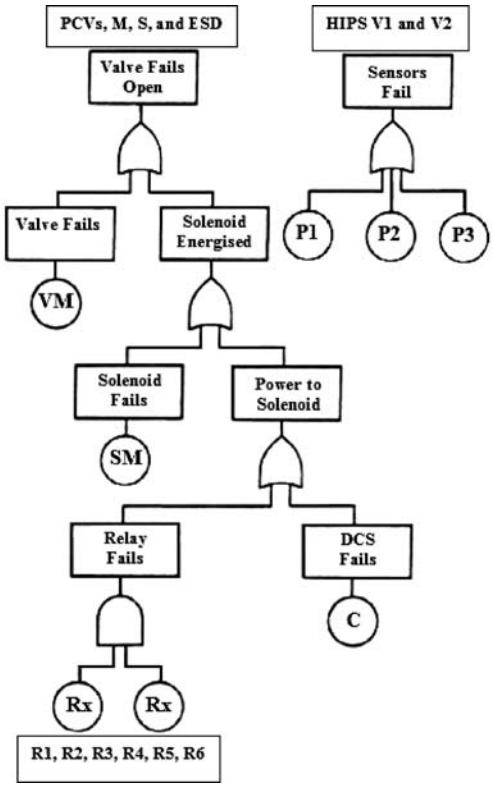


Fig. 5.38 Combination fault trees for modified cause-consequence diagram

Component failures on safety systems are not corrected during scheduled maintenance. Their failure probabilities are given by

$$Q_i = \lambda_i[\tau + \theta/2] \tag{5.84}$$

where:

Q_i = probability of the ith failure
λ_i = ith failure rate
τ = mean time to repair
θ = maintenance interval.

The calculated system unavailability is identical to that produced by the FTA method. This result does reflect well on the cause-consequence diagram method, in comparison to the FTA method, as it emphasises the fact that the example system can fail by a single component, namely the DCS. The remaining minimal cut sets are of order 4 or more and, therefore, have little effect on the overall system unavailability. For a system that contains a large number of small order minimal cut

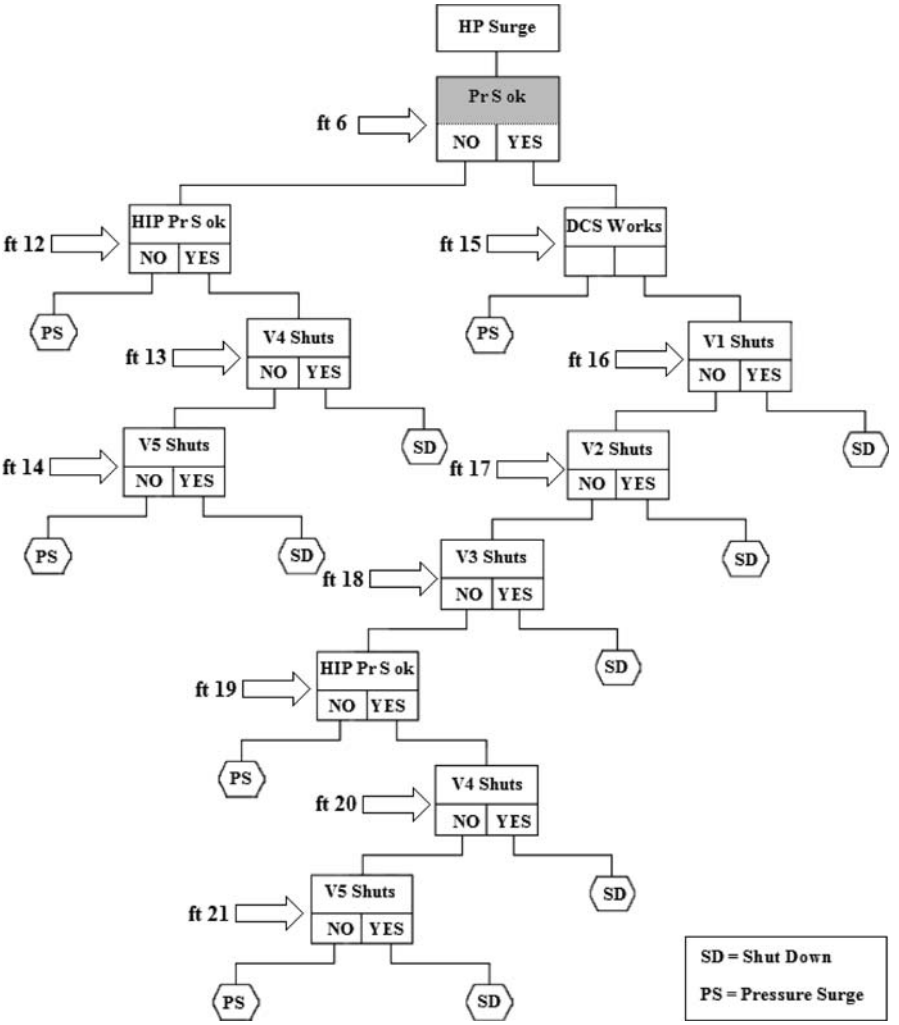


Fig. 5.39 Final cause-consequence diagram for HIPS system (Ridley et al. 1996)

sets, it can be seen that the cause-consequence diagram method would yield a more accurate result than that obtained from FTA.

The developed algorithm will produce the correct cause-consequence diagram and calculate the exact system failure probability for static systems with binary success or failure responses to the trigger event. This is achieved without having to construct the fault tree of the system, and retains the documented failure logic of the system (Ridley et al. 1996).

The cause-consequence diagram is reduced to a minimal form by, first, removing any redundant decision boxes and, second, manipulating any common failure events that exist on the same path. The common failure events can be extracted as common

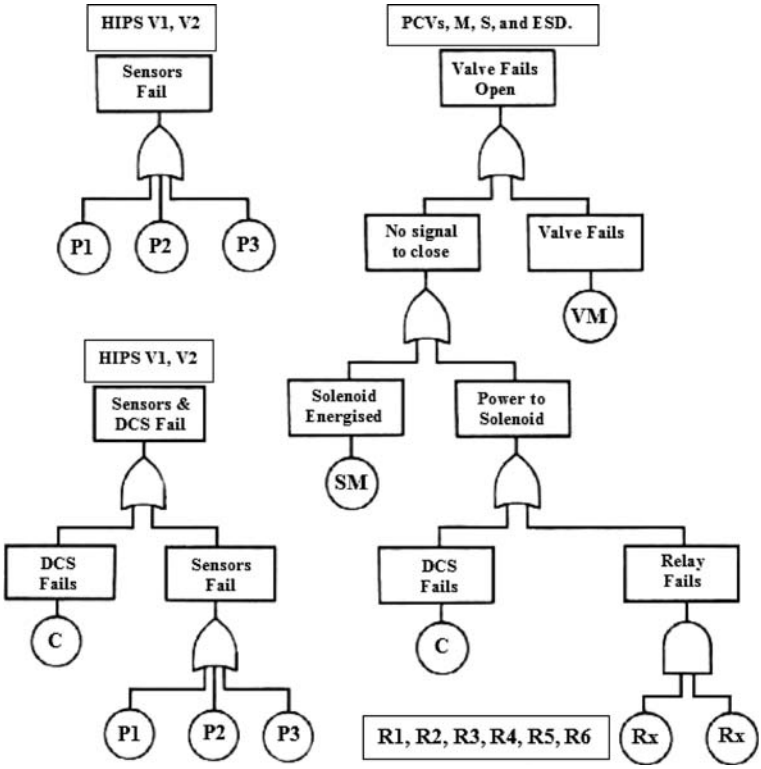


Fig. 5.40 Combination fault trees for the final cause-consequence diagram (Ridley et al. 1996)

sub-modules or individual events. This process is equivalent to constructing the fault tree, and identifying and extracting independent sub-modules. Thus, exact, rather than approximate calculations are performed.

5.2.4.3 Failure Modes and Safety Effects Evaluation

Failure modes and effects criticality analysis (FMECA) is a design discipline where an engineer examines and records the consequences of any (usually only single point) failure on the operation of a system. The purpose of the analysis is to highlight any significant problems with a design and, if possible, to change the design to avoid those problems (Price 1996). In contrast, *failure modes and safety effects (FMSE)* evaluation is a detail design discipline that examines and records the safety consequences of a system through *safety criticality analysis*.

a) Safety Criticality Analysis

In complex engineering designs, the determination of *safety criticality* is essentially an expansion of risk analysis in which focus is placed upon the importance of safety-critical equipment early in the engineering design stage. Any significant effect on the operational performance of critical equipment as a result of changes in designing for safety will inevitably have an impact on the performance of the total process. In effect, risk-based safety criticality analysis quantifies these impacts on the total process performance, whereby preventive maintenance tasks are scheduled according to required frequencies. Essential preventive maintenance intervals are set by equipment *age analysis* in which the rate of deterioration and resulting potential failure ages are determined through the statistical method of *residual life* evaluation. Safety criticality in process engineering is complex, and basically depends upon the reliability of equipment subject to a variety of failure risks. This complexity is due to the interaction between the various risks of failure. These risks are defined as the result of multiplying the consequence of failure by the probability of its occurrence.

Consequence of failure The main concern for equipment failure, particularly equipment functional failure, is its *consequence*. Consequences of functional failures may range from the *cost of replacement* of a failed component, to the *consequential damage* of equipment, and possibly to a *safety hazard* through loss of life or limb. The more complex equipment designs are, with regard to constituent components and their configuration, the more ways there are in which various functional failures can possibly occur.

Some typical process engineering consequences of functional failure are abnormal pressures, excessive vibration, overheating, cracking, rupturing, warping, etc. As many functional failures can be defined as there are different types of component functions. However, a point of interest that becomes evident after scrutinising these consequences of failure is that there are two *types* of consequences that can be defined, specifically *operational consequences of failure* and *physical consequences of failure*.

It is obvious that the consequences of functional failures such as abnormal temperature, abnormal pressure, excessive vibration, overheating, etc. are consequences affecting the *operational function* or *working performance* of the equipment or system. Similarly, the consequences of functional failures such as cracking, rupturing, warping, etc. are consequences affecting the *physical function* or *material design* of the equipment or system. Thus, at each level of a systems hierarchy, or systems breakdown structure (SBS), an item at a specific level may have functional failures of its operational or physical functions that may have consequences of functional failure affecting the operational or physical functions of a higher level of the systems hierarchy. These consequences of functional failure are then also recognised to be either operational consequences or physical consequences. Thus, the more complex equipment designs become, the more ways there are in which functional

failure can occur. As a result, equipment operational and physical consequences of functional failure can be grouped into five significant categories:

- *Safety* operational and physical consequences.
- *Economic* operational and physical consequences.
- *Environmental* operational and physical consequences.
- *Systems* operational and physical consequences.
- *Maintenance* operational and physical consequences.

Safety operational and physical consequences Safety operational and physical consequences of functional failure are alternately termed *critical functional failure consequences*. In general, if the *consequences of functional failure* are critical, then the functional failures resulting from the inability to carry out the operational or physical functions are defined as *critical failures*. Safety consequences of functional failure in certain operational or physical functions are always critical. In evaluating functional failure, the first consideration is safety.

Functional failures that fall into this category are classed as critical. These functional failures affect either the operational or physical functions of equipment that could have a *direct* effect on *safety*. The term *'direct'* implies certain limitations. The impact of the functional failure must be *immediate* if it is considered to be *direct*. Safety of equipment in this context implies certain specific definitions, where:

Safety is defined as "*not involving risk*".
Risk is defined as "*the chance of disaster or loss*".

It can be interpreted from these definitions that the concept of safety as not involving risk in the form of *disaster* has to do with *personal protection* against *injury* or the loss of '*life or limb*', and safety not involving risk in the form of *loss* of property has to do with *equipment protection* against '*consequential damage*'. Safety can thus be classified into two categories, one relating to *personal protection*, the other relating to *equipment protection*. Risk can be quantified as the product of the probability of occurrence (chance), with the level of severity of the risk (disaster or loss). Risk is an indication of the *degree of safety*. Thus:

$$\text{Risk} = \text{Severity} \times \text{Probability}$$

The *measure of probability* can be quantified in the form of statistical probability distributions or measures of statistical likelihood. Severity relates to the disaster or loss incurred. The *measure of severity* can thus be quantified based on two aspects—*accidents* and *incidents*, according to the two categories of safety (i.e. *personal protection* and *equipment protection*). In this regard, an *accident* is an undesired event that results in disastrous physical harm to a person. An *incident* is an undesired event that could result in a loss. In the context of safety, this loss is in the form of an *asset loss*, which implies *consequential damage* to equipment or property. Assessment of severity related to risk, or the *severity of risk*, would therefore be an *estimate* of the disaster or loss that can occur, whereas an *evaluation* of the severity related to risk would be an account of the *actual* disaster or loss that has occurred.

The *estimated severity* of risk is a vital tool in the evaluation of designing for safety, and is assessed on the basis of the *estimated measure of severity*, which is quantified in terms of two aspects, namely *accidents* and *incidents*, according to which an *estimation* of the possible occurrences of *accidents* or *incidents* needs to be made. This is known as the *estimated degree of safety (accidents or incidents)*.

The *estimated* degree of safety—accidents: This is assessed according to the contribution of the *estimated physical condition* of the equipment to its safety, the *estimated disabling injury frequency*, as well as the *estimated reportable accident frequency*, arising from functional failure predictions of the equipment resulting in *disastrous* safety consequence of failure.

However, not every critical functional failure results in an accident. Some such failures may have occurred with no disastrous safety consequences but, rather, with a loss in the form of an *asset loss*, which implies *consequential damage* to equipment or property. The *severity of risk* in this case is assessed on the basis of the measure of severity quantified in *incidents*, where an estimation of the *possible* occurrences of *incidents* is made. This is known as the *estimated degree of safety (incidents)*.

The *estimated* degree of safety—incidents: This is assessed according to the contribution of the *estimated physical condition* of the equipment to its safety, the *estimated downtime frequency*, as well as the *estimated reportable incident frequency*, arising from functional failure predictions of the equipment resulting in an *asset loss* consequence of failure. Aside from an assessment of severity related to risk, or the *severity of risk* being an assessment of the disaster or loss that can occur, the issue in designing for safety is not whether the estimated *degree of safety* is based on *accidents* or *incidents* being inevitable but, rather, whether they are *probable*—hence, the measure of *probability* in assessing risk.

Safety *operational* and *physical* consequences should always be assessed at the most conservative level and, in the absence of proof that a functional failure can affect safety, it is precautionary to nevertheless classify it by default as critical.

In contrast, the *actual severity* of risk is a vital tool in the *verification* of designing for safety, where the statistics of safety *operational* and *physical* consequences of functional failure, as well as of the *causes* of critical functional failures are essential for validating the safety criticality analysis applied during the detail design phase. The *actual severity* of risk is evaluated on the basis of the actual measure of severity that is quantified in the two aspects of *accidents* and *incidents*, according to which an analysis of the *actual* occurrences of *accidents* or *incidents* needs to be made. This is known as the *actual degree of safety (accidents or incidents)*.

The *actual* degree of safety—accidents: This is evaluated according to the contribution of the *actual physical condition* of the equipment to its safety, the *actual disabling injury frequency*, as well as the *actual reportable accident frequency*, arising from the functional failure history of the equipment resulting in *disastrous* safety consequence of failure. Similarly, *actual severity* is evaluated on the basis of the measure of severity quantified in *incidents*, where a determination of the *actual* occurrences of *incidents* needs to be made. This is known as the *actual degree of safety (incidents)*.

The *actual* degree of safety—incidents: This is evaluated according to the contribution of the *actual physical condition* of the equipment to its safety, the *actual downtime frequency*, as well as the *actual reportable incident frequency*, arising from the functional failure history of the equipment resulting in an *asset loss* consequence of failure. Besides safety operational and physical consequences of failure, the other consequences (economic, environmental, systems and maintenance) are typically measured as the *cost of losses* plus the *cost of repair* to the failed item and to any consequential damage (although, in reality, all safety consequences are eventually also measured as a cost risk). These cost risks of failure are also defined as the result of multiplying the consequence of failure (i.e. the *cost of losses* plus the *cost of repair*), by the probability of its occurrence.

Reliability analysis in engineering design tends, however, to simplify these risks to the point of impracticality where, for example, consideration is given only to single modes of failure, or only to random failure occurrences, or to maintenance that results in complete renewal and 'as new' conditions. In reality, the situation is much more complicated with interacting multiple failure modes, variable failure rates, as well as maintenance-induced failures that influence the rates of deterioration, and subsequent failure (Woodhouse 1999).

It is somewhat unrealistic to assume a specific failure rate of equipment within a complex integration of systems with complex failure processes. At best, the intrinsic failure characteristics of components of equipment are determined from quantitative probability distributions of failure data obtained in a somewhat clinical environment under certain operating conditions. The true failure process, however, is subject to many other factors, including premature or delayed preventive maintenance activities conducted during shutdowns of process plant.

It is generally accepted that shutdowns affect the failure characteristics of equipment as a whole, although it is debatable whether the end result is positive or negative from a *residual life* point of view, where residual life is defined as the remaining life expectancy of a component, given its survival to a specific age. This is a concept of obvious interest, and one of the most important notions in process reliability and equipment aging studies for safety criticality analysis.

Safety criticality analysis is thus always faced with combinations of interacting failure modes and variable failure rates, where the cumulative effects are much more important than estimates of specific probabilities of failure. Qualitative estimates of how long equipment might last in certain engineering processes, based on operating conditions and failure characteristics, are much more easily made than quantitative estimates of the chances of failure of individual equipment. These cumulative effects are represented in equipment survival curves where a best-fit curve is matched to specific survival data, and a pattern of risks calculated that would be necessary for these effects to be realised. In analysing survival data, there is often the need to determine not only the survival time distribution but also the residual survival time (or residual life) distribution. A typical equipment *survival curve* and *hazard curve* are illustrated in Fig. 5.41a and 5.41b (Smith et al. 2000).

Typical impact, risk exposure, lost performance, and direct cost patterns based on shutdown maintenance intervals for rotating equipment, as well as risk-based main-

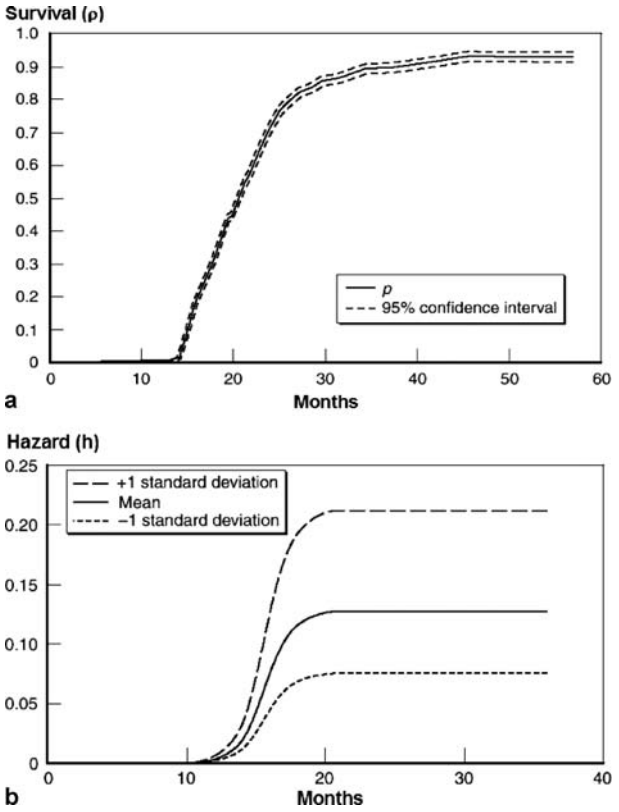


Fig. 5.41 **a** Kaplan–Meier survival curve for rotating equipment, **b** estimated hazard curve for rotating equipment

tenance patterns based on shutdown maintenance intervals for rotating equipment are illustrated in Fig. 5.42a and 5.42b (APT Maintenance 1999).

b) Risk-Based Maintenance

Risk-based maintenance is fundamentally an evaluation of maintenance tasks, particularly scheduled preventive maintenance activities in shutdown programs. It considers the impact of bringing forwards, or delaying, activities that are directed at preventing *cost risks* to coincide with essential activities that address *safety risks*. If the extent of these risks were known, and what they cost, the optimum amount of risk to take, and planned costs to incur, could be calculated. Similarly, better decisions could be made if the value of the benefits of improved performance, longer life and greater reliability was known. These risks and benefits are, however, difficult to quantify, and many of the factors are indeterminable. Cost/risk optimisation in this

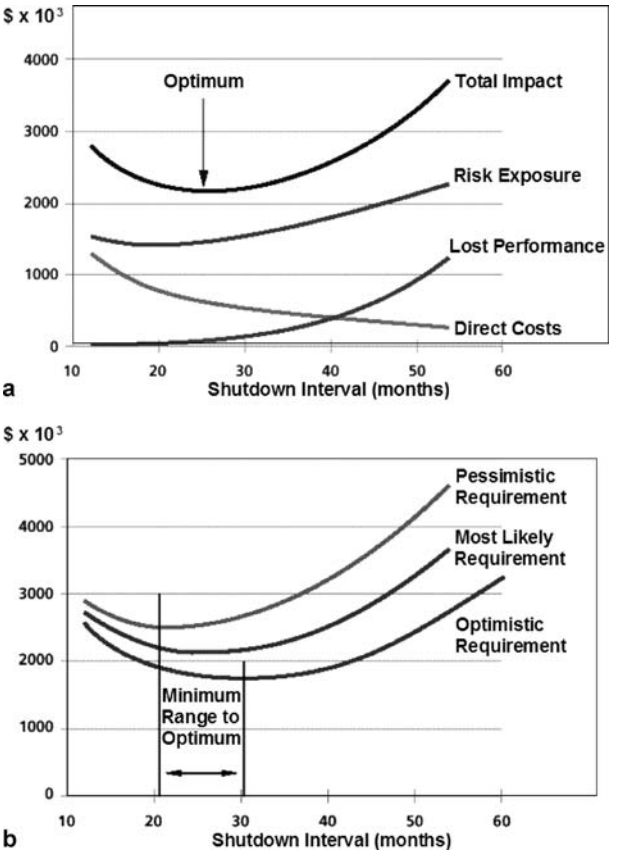


Fig. 5.42 **a** Risk exposure pattern for rotating equipment, **b** risk-based maintenance patterns for rotating equipment

context can thus be defined as the minimal total impact, and represents a trade-off between the conflicting interests of the need to reduce costs at the same time as the need to reduce the risks of failure. Both are measured in terms of cost, the former being the planned downtime cost plus the cost of preventive maintenance in an attempt to increase performance and reliability, and the latter being the cost of losses due to forced shutdowns plus the cost of repair and consequential damage.

The total impact is the sum of the planned costs and failure costs. When this sum is at a minimum, an optimal combination of the costs incurred and the failure risks is reached, as illustrated in Fig. 5.43.

Cost/risk trade-off decisions determine optimal preventive maintenance intervals for plant shutdown strategies that consider component renewal or replacement criteria, spares requirements planning, etc. Planned downtime costs plus the costs of preventive maintenance are traded-off against the risk consequences of premature or deferred component renewals or replacements, measured as the cost of losses plus

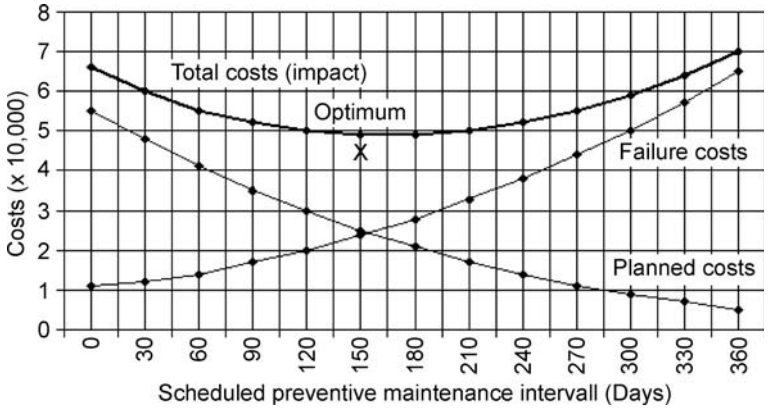


Fig. 5.43 Typical cost optimisation curve

the cost of repair. In each of these areas, cost/risk evaluation techniques are applied to assist in the application of a safety-critical maintenance approach.

Component renewal/replacement criteria are directly determined by failure modes and effects criticality analysis (FMECA), whereby appropriate maintenance tasks are matched to failure modes. In applying FMECA, the criticality analysis establishes a priority rating of components according to the consequences and measures of their various failure modes, which helps to prioritise the preventive maintenance activities for scheduled shutdowns. An example of an FMECA for process criticality of a control valve, based on failure consequences (downtime) and failure rate (1/MTBF), is given in Table 5.16.

Reliability, availability, maintainability and safety (RAMS) studies establish the most effective combination of the different types of maintenance (i.e. a *maintenance strategy*) for operational systems and equipment. The deliverable results are operations and maintenance procedures and work instructions in which the different types of maintenance are effectively combined for specific equipment.

Failure modes and effects criticality analysis (FMECA), as given in Table 5.16, is one of the most commonly used techniques for prioritising failures in equipment. The analysis at systems level involves identifying potential equipment failure modes and assessing the consequences of these for the system's performance.

Table 5.17 shows the designation of maintenance activities, the appropriate maintenance trade, and the recommended maintenance frequency for each failure mode, based on MTBF. It is evident that some activities need to be delayed to coincide with others.

Different types and levels of maintenance effort are applied, depending upon the process or functional criticality (Woodhouse 1999):

- Quantitative risk and performance analysis (such as RAM and FMECA) is warranted for about 5–10% of the most critical failure modes. This is where cost/risk optimisation is applicable for significant costs or risks that are sensitive to high-impact strategies.

Table 5.16 Typical FMECA for process criticality

Component	Failure description	Failure mode	Failure consequences	Failure causes	D/T (h) (plus damage)	MTTR (h) (repair time) and damage	MTBF (months)	Process criticality rating
Control valve	Fails to open	TLF	Production	Solenoid valve fails, failed cylinder actuator or air receiver failure	9	8	12	Medium critical
Control valve	Fails to open	TLF	Production	No PLC output due to modules electronic fault or cabling	4	2	6	Medium critical
Control valve	Fails to seal/close	TLF	Production	Valve disk damaged due to corrosion wear (same 'fails to open')	5	4	6	Medium critical
Control valve	Fails to seal/close	TLF	Production	Valve stem cylinders seized due to chemical deposition or corrosion	5	4	4	Medium critical
Instrument loop (press. 1)	Fails to provide accurate pressure indication	TLF	Maint.	Restricted sensing port due to blockage of chemical or physical accumulation	0	1	3	Low critical
Instrument loop (press. 2)	Fails to detect low pressure condition	TLF	Maint.	Low pressure switch fails due to corrosion or mechanical damage	0	2	3	Low critical
Instrument loop (press. 2)	Fails to detect low pressure condition	TLF	Maint.	Pressure switch relay or cabling failure	0	8	4	Low critical
Instrument loop (press. 2)	Fails to provide output signal for alarm	TLF	Maint.	PLC alarm function or indicator fails	0	8	4	Low critical

Table 5.17 FMECA with preventive maintenance activities

Component	Failure description	Failure causes	D/T (h) (plus damage)	MTTR (h) (repair time) and damage	MTBF (months)	Maintenance activity	Maintenance trade	Maintenance frequency
Control valve	Fails to open	Solenoid valve fails, failed cylinder actuator or air receiver failure	9	8	12	Service control valve. Replace components and test PLC interface	Instr. tech.	12 monthly
Control valve	Fails to open	No PLC output due to modules electronic fault or cabling	4	2	6	Covered by control valve service as above	Instr. tech.	12 monthly
Control valve	Fails to seal/close	Valve disk damaged due to corrosion wear (same causes as 'fails to open')	5	4	6	Remove control valve and check valve stem, seat and disk or diaphragm for deterioration or corrosion and replace with overhauled valve if required	Fitter	6 monthly
Control valve	Fails to seal/close	Valve stem cylinders seized due to chemical deposition or corrosion	5	4	4	Covered by control valve condition assessment and replace components	Instr. tech.	6 monthly

Table 5.17 (continued)

Component	Failure description	Failure causes	D/T (h) (plus damage)	MTTR (h) (repair time) and damage	MTBF (months)	Maintenance activity	Maintenance trade	Maintenance frequency
Instrument loop (press. 1)	Fails to provide accurate pressure indication	Restricted sensing port due to blockage of chemical or physical accumulation	0	1	3	Remove pressure gauge and check for blocked sensing lines and gauge deterioration. Replace with new gauge if required	Instr. tech.	3 monthly
Instrument loop (press. 2)	Fails to detect low pressure condition	Low pressure switch fails due to corrosion or mechanical damage	0	2	3	Verify correct operation of pressure switch and wiring. Test alarm's operation	Instr. tech.	3 monthly
Instrument loop (press. 2)	Fails to detect low pressure condition	Pressure switch relay or cabling failure	0	8	4	Covered by switch operation verification	Instr. tech.	3 monthly
Instrument loop (press. 2)	Fails to provide output signal for alarm	PLC alarm function or indicator fails	0	8	4	Covered by switch operation verification	Instr. tech.	3 monthly

- Rule-based analysis methods (such as RCM and RBI) are more appropriate for about 40–60% of the critical failure modes, particularly if supplemented with economic analysis of the resulting impact strategies. This is where cost/risk optimisation is applicable for the costs or risks for setting preventive maintenance intervals.
- Review of existing maintenance (excluding simple FMEA studies) provides a simple check at the lower levels of criticality to verify that there is a valid reason for the maintenance activity, and that the cost is reasonable compared to the consequences.

c) Safety Criticality Analysis and Risk-Based Maintenance

Safety criticality analysis was previously considered as the assessment of failure risks. In this context, safety criticality analysis is applied to determine the essential maintenance intervals, and the impact of premature or delayed preventive maintenance activities where failure risks are considered to be safety critical. A safety/risk scale is applied, based on a specific cost benchmark (usually computed as the cost of output per time interval) related to the cost of losses and the likelihood of failure.

A safety criticality model to determine the optimal maintenance interval, and the impact of premature or delayed preventive maintenance activities considers the following:

- A quantified description of the degradation process, using estimates wherever data are not available, as well as identification of failure modes and related causes.
- Cost calculations for material and maintenance labour costs for each failure mode, including possible consequential damage.
- Cost/risk calculations for alternative preventive maintenance intervals based on a specific cost benchmark related to the cost of losses and the likelihood of failure.
- Cost criticality rating of failure modes, and sensitivity testing to the limits of the likelihood of failure under uncertainty of unavailable or censored data.
- Identification of key decision drivers (which assumptions have the greatest effect upon the optimal decision), for review of the preventive maintenance program. In many cases, there are several interacting failure modes, causes and effects, all in the same evaluation.

The preventive maintenance program or, in the case of continuous processes, the shutdown strategy thus becomes a compromise of scheduled times and costs. Some activities will be performed ahead of their ideal timing, whilst others will be delayed to share the downtime opportunity determined by safety-critical shuts.

The risks and performance impact of delayed activities, and the additional costs of deliberate over-maintenance in others, both contribute to the costs for a particular shutdown program. The degree of advantage, on the other hand, is controlled

by the costs involved. The downtime impact (the cost of losses due to forced shutdowns as a result of failure, plus the cost of repair to the failed item and to any consequential damage) often dominates the direct cost advantage (planned shutdown lost opportunity costs, use of facilities, materials and labour costs, etc.) of shutting down and starting up again. Such a *cost criticality analysis* also reveals the scope for *de-bottlenecking* improperly evaluated reliability constraints by eliminating frequent interim shutdowns and extending operational run lengths. The analysis process is also able to calculate the net payback for such de-bottlenecking. The grouping and re-grouping of activities as well as re-programming the preventive maintenance program (i.e. combining activities in different bundles and moving the bundles to shorter or longer intervals) are fundamentally a scheduling problem, requiring the application of formalised risk analysis and decision criteria based on assessment scales, and the use of computer automated computation. Table 5.18 shows the application of cost criticality analysis to the FMECA for process criticality of the control valve given in Table 5.17. It indicates the cost criticality rating of each failure mode related to the cost of losses and the cost risk based on estimates of the likelihood of failure. Table 5.19 shows a comparison between the process criticality rating and the cost criticality rating of each failure mode of the control valve. In this case, the ratings correspond closely with one another.

The maintenance frequencies of the preventive maintenance activities that were typically based on the mean time between failures (MTBF) are, however, not relative to either the process criticality rating or the cost criticality rating. The maintenance frequencies thus require review to determine the optimal maintenance intervals whereby the impact of premature or delayed preventive maintenance activities is considered.

This example of a relatively important item of equipment, such as a process control valve, is typical of many such equipment in process plant where RAM, FMECA or RCM analysis do not provide sufficient information for decisive decision-making, as the equipment's failure modes are not significantly high risk but rather medium risk. Where the criticality ratings are not significant (i.e. evidence of high criticality), as in this case of the control valve, maintenance optimisation becomes difficult, necessitating a review of the risk analysis and decision criteria according to qualitative estimates.

d) Risk Analysis and Decision Criteria

In typical process plant shutdown programs, decisions concerning the extent and timing of component renewal/replacement activities are generally determined by the dominant failure modes that, in effect, relate to less than a third of the program's total preventive maintenance activities. Criticality ranking or prioritising of equipment according to the consequences of failure modes is essential for a risk-based maintenance approach, though comparative studies have shown that qualitative risk ranking is, in many cases, just as effective in identifying the key shutdown drivers, often at a fraction of the cost. Typically, these risks can be ranked by designating

Table 5.18 FMECA for cost criticality

Component	Failure description	Failure mode	Failure causes	Defect. MATL & LAB ($)/failure (incl. damage)	Econ. $/failure (prod. loss)	Total $/failure (prod. and repair)	Risk	Cost criticality rating
Control valve	Fails to open	TLF	Solenoid valve fails, failed cylinder actuator or air receiver failure	$5,000	$68,850	$73,850	6.00	Medium cost
Control valve	Fails to open	TLF	No PLC output due to modules electronic fault or cabling	$2,000	$30,600	$32,600	6.00	Medium cost
Control valve	Fails to seal/close	TLF	Valve disk damaged due to corrosion wear (same causes as 'fails to open')	$5,000	$38,250	$43,250	6.00	Medium cost
Control valve	Fails to seal/close	TLF	Valve stem cylinders seized due to chemical deposition or corrosion	$5,000	$38,250	$43,250	6.00	Medium cost
Instrument loop (press. 1)	Fails to provide accurate pressure indication	TLF	Restricted sensing port due to blockage of chemical or physical accumulation	$500	$0	$500	2.00	Low cost

Table 5.18 (continued)

Component	Failure description	Failure mode	Failure causes	Defect. MATL & LAB ($)/failure (incl. damage)	Econ. $/failure (prod. loss)	Total $/failure (prod. and repair)	Risk	Cost criticality rating
Instrument loop (press. 2)	Fails to detect low pressure condition	TLF	Low pressure switch fails due to corrosion or mechanical damage	$10,000	$0	$10,000	2.00	Low cost
Instrument loop (press. 2)	Fails to detect low pressure condition	TLF	Pressure switch relay or cabling failure	$10,000	$0	$10,000	2.00	Low cost
Instrument loop (press. 2)	Fails to provide output signal for alarm	TLF	PLC alarm function or indicator fails	$10,000	$0	$10,000	2.00	Low cost

Table 5.19 FMECA for process and cost criticality

Component	Failure description	Failure mode	Failure consequences	Total $/failure (prod. and repair)	Cost risk	MTBF (months)	Process criticality rating	Cost criticality rating	Maintenance frequency
Control valve	Fails to open	TLF	Production	$73,850	6.00	12	Medium criticality	Medium cost	12 monthly
Control valve	Fails to open	TLF	Production	$32,600	6.00	6	Medium criticality	Medium cost	12 monthly
Control valve	Fails to seal/close	TLF	Production	$43,250	6.00	6	Medium criticality	Medium cost	6 monthly
Control valve	Fails to seal/close	TLF	Production	$43,250	6.00	4	Medium criticality	Medium cost	6 monthly
Instrument loop (press. 1)	Fails to provide accurate pressure indication	TLF	Maint.	$500	2.00	3	Low criticality	Low cost	3 monthly
Instrument loop (press. 2)	Fails to detect low pressure condition	TLF	Maint.	$10,000	2.00	3	Low criticality	Low cost	3 monthly
Instrument loop (press. 2)	Fails to detect low pressure condition	TLF	Maint.	$10,000	2.00	4	Low criticality	Low cost	3 monthly
Instrument loop (press. 2)	Fails to provide output signal for alarm	TLF	Maint.	$10,000	2.00	4	Low criticality	Low cost	3 monthly

qualitative assessment values for the likelihood of occurrence and the impact that the risk may have on costs. Assessment values for risk may be designated as indicated previously, where risk has been defined as the result of multiplying the consequence of the failure mode (i.e. its severity) by the probability of failure (i.e. its likelihood):

$$\text{Risk } (R) = \text{Severity} \times \text{Probability (or Likelihood)}$$

Severity
The use of qualitative assessment scales for determining the severity of a failure consequence is common in risk analysis, where severity criteria are designated a value ranging from 10 to 1. The most severe consequence is valued at 10 (disabling injury—life risk), whereas no safety risk is valued at 1, or 0, as indicated in the risk assessment scale in Table 5.20.

Likelihood
Many different scales have been developed for determining the likelihood of failure occurrence. One commonly used scale is expressed in terms of 'probability qualifiers' given as:

$$\begin{aligned} \text{Actual occurrence} &= 0.95 \text{ to } 1.00\,, \\ \text{Probable occurrence} &= 0.50 \text{ to } 0.95\,, \quad \text{and} \\ \text{Possible occurrence} &= \text{less than } 0.50\,. \end{aligned}$$

Criticality
Once an overall total and an overall average value of risk has been assessed according to the risk assessment scale, a criticality rating can be defined for each failure mode, using the following expression:

$$\text{Criticality } (C) = \text{Risk} \times \text{Failure rate}$$

Failure Rate
If the failure rate for the item cannot be determined from available data, a representative estimation for failure rate in high-corrosive process applications can be used. This is done by the following qualifying values:

Qualification	Failure rate ($\times 10^{-4}$)
Very low	$<$100
Low	100 to 500
Medium	500 to 1,000
High	1,000 to 5,000
Very high	$>$5,000

Table 5.20 Risk assessment scale

Risk assessment scale Estimated degree of safety:	Risk assessment values: Degree of severity × Probability		
Severity criteria	Actual 0.95 to 1.00	Probable 0.50 to 0.95	Possible 0.01 to 0.05
(Disabling injury)	Deg. Prob. Risk	Deg. Prob. Risk	Deg. Prob. Risk
Life risk	10	10	10
Loss risk	9	9	9
Health risk	8	8	8
(Reported accident)			
People risk	7	7	7
Process risk	6	6	6
Product risk	5	5	5
(Physical condition)			
Damage risk	4	4	4
Defects risk	3	3	3
Loss risk	2	2	2
(No safety risk)	1	1	1
Overall risk	Total	Total	Total
Overall average	Average	Average	Average

e) Qualitative Criticality Analysis

Qualitative criticality analysis is structured in a *failure modes and safety effects (FMSE)* analysis, in contrast to the standard FMECA, which is based on failure rates, MTBF and MTTR. The outcome of the FMSE, given in Table 5.21, indicates that the dominant failure modes that are the key shutdown drivers in determining the optimum maintenance frequency are the two control valve failure modes of medium criticality and scheduled frequency of 6 months.

All other tasks relating to the control valve can be re-scheduled into this half-yearly shut. This implies that the annual scheduled service of the control valve can be premature with a low risk impact, and the quarterly scheduled checks or component replacements of the pressure instrument loops (pressure gauges and switches) can be delayed with low risk impact.

A cost criticality analysis can now be conducted on the basis of the shutdown frequency of 6 months being the estimated likelihood of failure for all the relevant failure modes. This approach is repeated for all those items of equipment initially found to be critical items according to a ranking of their consequences of failure. The task seems formidable but, following the Pareto principle (or 80–20 rule), in most cases 80% of cost risk consequences are due to only 20% of all components. Table 5.21 shows the application of qualitative risk assessment in the form of an FMSE for process criticality of the control valve given in Table 5.19.

Table 5.21 Qualitative risk-based FMSE for process criticality, where (1)=likelihood of occurrence (%), (2)=severity of the consequence (rating), (3)=risk (probability×severity), (4)=failure rate (1/MTBF), (5)=criticality (risk×failure rate)

Component	Failure description	Failure mode	Failure consequences	Failure causes	(1)	(2)	(3)	(4)	(5)	Criticality rating
Control valve	Fails to open	TLF	Production	Solenoid valve fails, failed cylinder actuator or air receiver failure	75%	6	4.50	0.083	0.37	Low criticality
Control valve	Fails to open	TLF	Production	No PLC output due to modules electronic fault or cabling	75%	6	4.50	0.167	0.75	Low criticality
Control valve	Fails to seal/close	TLF	Production	Valve disk damaged due to corrosion wear (same causes as 'fails to open')	100%	6	6.00	0.167	1.0	Medium criticality
Control valve	Fails to seal/close	TLF	Production	Valve stem cylinders seized due to chemical deposition or corrosion	100%	6	6.00	0.25	1.5	Medium criticality
Instrument loop (press. 1)	Fails to provide accurate pressure indication	TLF	Maint.	Restricted sensing port due to blockage of chemical or physical accumulation	100%	2	2.00	0.33	0.66	Low criticality

Table 5.21 (continued)

Component	Failure description	Failure mode	Failure consequences	Failure causes	(1)	(2)	(3)	(4)	(5)	Criticality rating
Instrument loop (press. 2)	Fails to detect low pressure condition	TLF	Maint.	Low pressure switch fails due to corrosion or mechanical damage	100%	2	2.00	0.33	0.66	Low criticality
Instrument loop (press. 2)	Fails to detect low pressure condition	TLF	Maint.	Pressure switch relay or cabling failure	75%	2	1.50	0.25	0.38	Low criticality
Instrument loop (press. 2)	Fails to provide output signal for alarm	TLF	Maint.	PLC alarm function or indicator fails	100%	2	2.00	0.25	0.5	Low criticality

f) Residual Life Evaluation

Component *residual life*, in the context of a renewal/replacement process that is typically carried out during scheduled preventive maintenance shutdowns in process plant, is in effect equivalent to the time elapsed between shutdowns. This is, however, not the true residual life of the component based on its reliability characteristics. The difference between the two provides a suitable means of comparison for maintenance optimisation of safety-critical components.

Optimum maintenance intervals are best determined through the method of *equipment age analysis*, which identifies the rate of component deterioration and potential failure ages. The risk-based maintenance technique of residual life assessment is ideally applied in equipment age analysis where the frequencies of preventive maintenance activities in shutdown programs can be optimised. However, residual life is widely used in modelling stochastic processes during detail engineering design, and is one of the random variables that determines the design requirements for component renewal/replacement; the other being the component age once the process design has progressed beyond the engineered installation stage, and has been in operation for some time.

In reliability theory, residual life appears as the time until the next failure, whereas for the renewal/replacement process it is normally expressed as a mathematical function of *conditional* reliability in which the residual life is determined from the component age. The mean residual life or remaining life expectancy function at a specific component age is defined to be the expected remaining life given survival to that age. It is a concept of obvious interest in maintenance optimisation, and most important in process reliability.

g) Failure Probability, Reliability and Residual Life

There are fundamentally two measures of reliability: the failure density function, which quantifies how many components would fail at different time points (i.e. a combination of how many components survive at each point, and the risk of failure in the interval up to the following time point), and the hazard rate, which is the conditional chance of failure, assuming the equipment has survived so far. It is the hazard rate that is essential for decisions about how long equipment can be left in service with a related risk of failure, or whether it should be renewed or replaced. Component failure density in a common series systems configuration (or in a complex system reduced to a simple series configuration) is defined by the following function

$$f_i(t) = \lim_{\Delta t \to 0} \frac{\alpha_S(t) - \alpha_S(t + \Delta t)}{\alpha_0 \Delta t} \tag{5.85}$$

where:

$f_i(t) =$ the ith component failure
$\Delta t \quad =$ the time interval

α_0 = the total number of components in operation at time $t = 0$
α_S = the number of components surviving at time t or $t + \Delta t$.

The ith component cumulative distribution function (failure probability) is defined by the following expression

$$F_i(t) = \int_0^t f_i(t)\,\mathrm{d}t \tag{5.86}$$

and the ith component reliability is defined by:

$$R_i(t) = \{1 - F_i(t)\}$$

Substituting the equation for $F_i(t)$ in the equation for $R_i(t)$ leads to

$$R_i(t) = 1 - \int_0^t f_i(t)\,\mathrm{d}t \tag{5.87}$$

However, a commonly used alternative expression for $R_i(t)$ is

$$R_i(t) = \mathrm{e}^{-\int_0^t \lambda_i(t)\,\mathrm{d}t} \tag{5.88}$$

where:

$\lambda_i(t)$ = the ith component hazard rate or instantaneous failure rate.

In this case, a component failure time can follow any statistical distribution function of which the hazard rate is known. The expression $R_i(t)$ is reduced to

$$R_i(t) = \mathrm{e}^{-\lambda_i t} \tag{5.89}$$

The mean time between failures (MTBF) is defined by the following expression

$$\mathrm{MTBF} = \int_0^\infty R(t)\,\mathrm{d}t \tag{5.90}$$

Substituting the expression for $R_i(t)$ and integrating in the series gives the model for MTBF—in effect, this is the sum of the inverse values of the component hazard rates, or instantaneous failure rates of all the components in the series

$$\mathrm{MTBF} = \left[\sum_{i=1}^{n} \lambda_i\right]^{-1} \tag{5.91}$$

where:

λ_i = the ith component hazard rate or instantaneous failure rate.

Residual life Let T denote the time to failure. The *survival function* can then be expressed as

$$R(t) = P(T > t) \tag{5.92}$$

The conditional survival function of a component that has survived without failure can now be formulated.

The *conditional survival function* of a component that has survived (without failure) up to time x is

$$\begin{aligned} R(t|x) &= P(T > t+x|T > x) \\ &= \frac{P(T > t+x)}{P(T > t)} \\ &= \frac{R(t+x)}{R(x)} \end{aligned} \tag{5.93}$$

$R(t|x)$ denotes the probability that a component (of age x) will survive an extra time t. The *mean residual life (MRL)* of a component of age x can thus be expressed as

$$\mathrm{MRL}(x) = \int_0^\infty R(t|x)\,\mathrm{d}t \tag{5.94}$$

If $x = 0$, then the initial age is zero, implying a new item, $\mathrm{MRL}(0) = \mathrm{MTTF}$, the mean time to fail. The difference between MTBF and MTTF is in their application. Although both are similarly calculated, MTBF is applied to components that are repaired, and MTTF to components that are replaced. The mean residual life (MRL) function or remaining life expectancy function at age x is defined to be the expected remaining life given survival to age x. Consider now the reliable life for the one-parameter exponential distribution, compared to the residual life

$$h(x) = \frac{\mathrm{MRL}(x)}{\mathrm{MTTF}} \tag{5.95}$$

Certain characteristics of the comparison between the *mean residual life MRL* and the *mean time to fail MTTF* are the following:

- When the time to failure for an item, T, has an exponential distribution (i.e. constant hazard rate), then the function $h(x) = 1$ for all x and $\mathrm{MRL} = \mathrm{MTTF}$.
- When T has a Weibull distribution with shape parameter $\beta < 1$ (i.e. a decreasing failure rate), then $h(x)$ is an *increasing* function.
- When T has a Weibull distribution with shape parameter $\beta > 1$ (i.e. an increasing failure rate), then $h(x)$ is a *decreasing* function.

Thus, in the case of scheduled preventive maintenance activities with frequencies less than their MTTF, the cost/risk of *premature* renewal or replacement is the loss of potential equipment life (accumulated over all components), equivalent to the sum of the differences between the residual life of each component and the scheduled

frequency. Similarly, for those scheduled preventive maintenance activities with frequencies greater than their MTTF, the cost/risk of *delayed* renewal or replacement is the cost of losses (accumulated over all components) due to forced shutdowns as a result of failure, plus the cost of repair to the failed component and to any consequential damage. The likelihood of failure is equivalent to the ratio of the differences between the MTTF of each component and the scheduled frequency, divided by the differences between the residual life of each component and the scheduled frequency. Table 5.22 shows the replacement of (1) = likelihood of occurrence and (4) = failure rate with the calculated residual life values, to the FMSE of Table 5.21.

h) Sensitivity Testing

Sensitivity testing in FMSE considers limits of the likelihood of failure. This is done by representing the likelihood as a statistical distribution (usually, the standard normal distribution), and determining the variance and standard deviation of the range of likelihood values. Sensitivity testing in this case is thus a statistical measure of how well a likelihood test correctly identifies a failure condition. This is illustrated in the concept tabulated below. The sensitivity is the proportion of 'true positives' or true likelihood of failure, and is a parameter of the test.

Specificity in the concept diagram is a statistical measure of how well a likelihood test correctly identifies the negative cases, or those cases that do not result in a failure condition. The significance level of the sensitivity test is a statistical hypothesis testing concept. It is defined as the probability of making a decision to reject the null hypothesis when the null hypothesis is actually true (a decision known as a type I error, or 'false positive determination'). The decision is made using the *P*-value of the hypothesis test. If the *P*-value is less than the significance level, then the null hypothesis is rejected. The smaller the *P*-value, the more significant the result is considered to be. Different α-levels of the hypothesis test indicate greater confidence in the determination of significance with smaller α-levels but run greater risks of failing to reject a false null hypothesis (a type II error, or 'false negative determination'). Selection of an α-level involves a compromise in tendency towards a type I error, or a type II error. A common misconception is that a statistically significant result is always of practical significance. One of the more common problems in significance testing of sensitivity is the tendency for multiple comparisons to yield spurious significant differences even where the null hypothesis is true. For example, in a comparison study of the likelihood of failure of several failure modes, using an α-level of 5%, one comparison will likely yield a significant result despite the null hypothesis being true.

During a sensitivity analysis, the values of the specified sensitivity variables are modified with changes to the expected value. For one-way sensitivity analyses, one variable is changed at a time. For two-way sensitivity analyses, two variables are changed simultaneously. For a more sophisticated sensitivity analyses, an FMSE what-if analysis is conducted. The differences between the outcomes of the qualitative risk-based FMSE and related cost risk for different expected values can then be

Table 5.22 FMSE for process criticality using residual life

Component	Failure description	Failure mode	Failure consequences	(1)	(2)	(3)	(4)	(5)	Criticality rating	Cost criticality rating	Maintenance frequency
Control valve	Fails to open	TLF	Production	75%	6	4.50	0.083	0.37	Low criticality	Medium cost	6 monthly
Control valve	Fails to open	TLF	Production	75%	6	4.50	0.167	0.75	Low criticality	Medium cost	6 monthly
Control valve	Fails to seal/close	TLF	Production	100%	6	6.00	0.167	3.0	Medium criticality	Medium cost	6 monthly
Control valve	Fails to seal/close	TLF	Production	100%	6	6.00	0.5	1.5	HIGH criticality	Medium cost	6 monthly
Instrument loop (press. 1)	Fails to provide accurate pressure indication	TLF	Maint.	100%	2	2.00	0.67	1.34	Medium criticality	Low cost	6 monthly
Instrument loop (press. 2)	Fails to detect low pressure condition	TLF	Maint.	100%	2	2.00	0.67	1.34	Medium criticality	Low cost	6 monthly
Instrument loop (press. 2)	Fails to detect low pressure condition	TLF	Maint.	100%	2	2.00	0.5	1.0	Medium criticality	Low cost	6 monthly
Instrument loop (press. 2)	Fails to provide output signal for alarm	TLF	Maint.	100%	2	2.00	0.5	1.0	Medium criticality	Low cost	6 monthly

	Condition (likelihood of failure)		
	True	False	
Positive	True positive	False positive (type I error, *P*-value)	Positive predicted value
Negative	False negative (type II error)	True negative	Negative predicted value
Sensitivity		Specificity	

determined. Using decision trees and influence diagrams details all the possible options for a decision model. Decision trees provide a more formal structure in which decisions and chance events are linked from left to right in the order they would occur. Probabilities of the likelihood of failure events are added to each node in the tree. A decision analysis generates a risk profile. The risk profile compares the sensitivity of different decision options. Such sensitivity analysis is best conducted with the aid of specialised application software such as @RISK©, in which the outcome is expressed as a probability distribution, as illustrated in the insert below (Fig. 5.44).

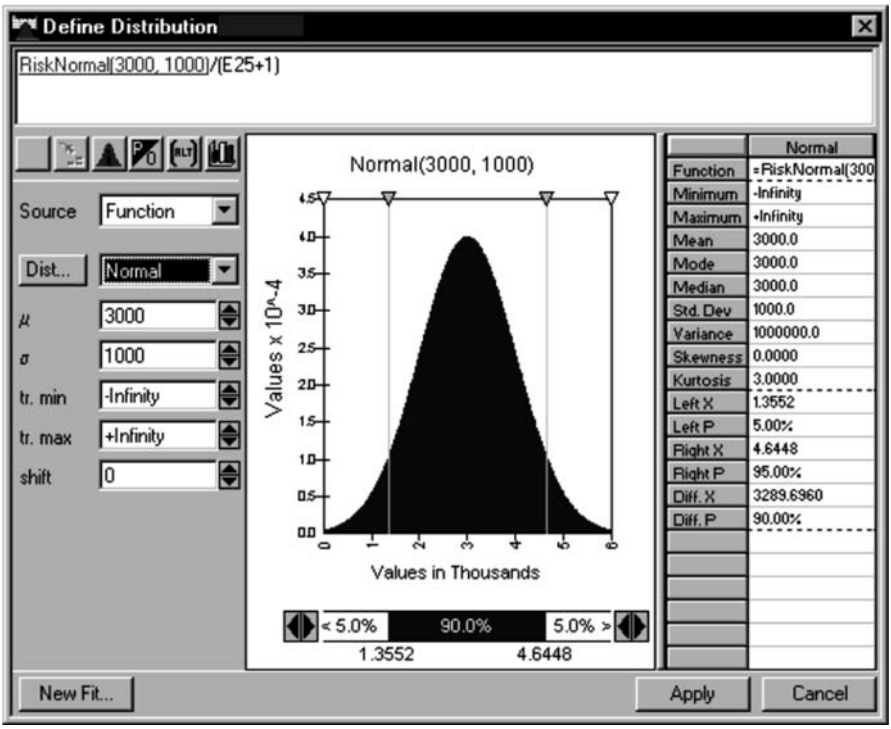


Fig. 5.44 Probability distribution definition with @RISK (Palisade Corp., Newfield, NY)

5.3 Analytic Development of Safety and Risk in Engineering Design

A significant factor in considering analytic development of safety and risk in engineering design is the extent to which *probabilistic analysis* and *deterministic analysis* can complement each other in safety and risk prediction, assessment and evaluation of engineered installations at each respective phase of the engineering design process. This requires an understanding of the advantages of each specific approach taken in the analysis of safety, and the basic concepts of *potential risk* and *residual risk* (de Gelder 1997).

Concepts of risk The prediction, assessment and evaluation of risk in the conceptual, preliminary/schematic or detail design stages respectively of engineered installations have to distinguish between:

- *potential risk*, which can lead to accidents or incidents if no protection measures are considered or taken,
- *residual risk*, which remains after having considered all measures taken to prevent accidents or incidents, and to mitigate their consequences.

The main contributions to residual risk stem from events that are not considered in the design, such as vessel rupture; an accident/incident progression worse than the assumptions considered in the design basis, such as multiple failures, common mode failures (resulting in complete failure of a safety system) and operator errors; cumulative occurrence of initiating events that are considered in the design but not accounted for, since cumulative occurrence is not considered to be a design basis event.

As considered previously, the assessment of risk requires two measures—specifically, the frequency of occurrence of potential accidents, and the severity of their consequences. During the analysis of safety, both these measures are considered with the objective that accidents with the most significant consequences should have the lowest frequencies of occurrence. The main objective of safety analysis is to verify that measures taken at the design stage, as well as during construction and operation of the engineered installation are adequate in achieving the prescribed safety requirements.

The probabilistic safety analysis approach The probabilistic approach enables the prediction or assessment of the major contributors to potential risk, and evaluation of the most significant contributors for further reduction of residual risk. The major steps in a probabilistic safety analysis are as follows:

- Identification of the initiating events and the plant operational states to be considered.
- Analysis of the possible accident scenarios, by means of event trees.
- Reliability analysis, by means of fault trees, of the systems considered in the event trees.

- Collection of probabilistic data (failure probability or unavailability for test and maintenance, initiating event frequencies).
- Use of analytic techniques such as sneak analysis, genetic algorithms and neural nets.
- Event sequence quantification, resulting in a frequency for each event.
- Interpretation of results (including sensitivity and importance analyses).

The deterministic safety analysis approach This approach has constituted a basis for the design of most high-risk engineered installations. The deterministic approach is based on regulations and guides established by the appropriate regulatory authority. The major steps in a deterministic safety analysis are the following:

- *Identification and categorisation of events considered in the design basis*:
  At the beginning of the design stage, a list of initiating events to be covered in the design is established and constitutes the so-called *design basis events*. These are then grouped into categories, based on their estimated frequency of occurrence. This categorisation of the initiating events is basically into classes, depending on the significance of the overall risk posed by the engineered installation. For example, the categorisation of initiating events into classes was established by the US Nuclear Regulatory Commission for high-risk engineered installations such as nuclear power plants (NUREG 75/014 1975; NUREG/CF-1401 1980). The following categorisation is of initiating events into classes:
  - Class 1: normal operation,
  - Class 2: incidents of moderate frequency,
  - Class 3: incidents/accidents of low frequency,
  - Class 4: hypothetical accidents.
- *Analysis of enveloping scenarios*:
  For each category, a number of enveloping scenarios are identified in such a way that their analysis covers all events to be considered in that category. Each enveloping scenario is then analysed by using conservative assumptions in the initial conditions of plant, such as:
  - power, flows, pressures, temperatures,
  - most unfavourable moment in the process cycle,
  - instrumentation uncertainties,
  - hypotheses concerning the accident/incident progression.
- *Evaluation of consequences*:
  The potential consequences of these enveloping scenarios are analysed using conservative assumptions, such as:
  - the initial activity of a primary circuit is supposed to be equal to the maximum activity allowed by the technical specifications,
  - unfavourable climatic conditions.

- *Verification with respect to acceptance criteria*:
  The results of the analysis of the enveloping scenarios are finally compared with predefined acceptance criteria. These acceptance criteria can be expressed in relation to parameters of the engineered installation, and to the protection of people and the environment. When all analyses show that acceptance criteria are met, the proposed design is accepted in the deterministic safety approach.

Below, various methodologies for the analytic development of safety and risk in the design of engineered installations are considered, incorporating *probabilistic analysis* in the respective prediction, assessment and evaluation of safety and risk problems at each phase of the engineering design process. Various AI analytic techniques presented, such as *evolutionary algorithms*, *genetic algorithms* and *neural networks*, are basically stochastic search and optimisation heuristics derived from classic evolution theory and implemented in intelligent computer automated methodology in the prediction, assessment and evaluation of engineering design safety and risk.

5.3.1 Analytic Development of Safety and Risk Prediction in Conceptual Design

In this section, the development of a *design space* is considered in which methods of design preferences and scenarios are integrated with analytic techniques such as *evolutionary algorithms*, *genetic algorithms* and/or *artificial neural networks* to perform multi-objective optimisation in designing for safety. In Sect. 5.4, computer automated methodology is presented in which optimisation algorithms have been developed for *knowledge-based expert systems* within a *blackboard model* that is applied in determining the integrity of engineering design. Certain approaches are therefore adopted for the prediction of risk in the conceptual design stage, specifically in:

i. Establishing an analytic basis for developing an intelligent computer automated system;
ii. Evolutionary computing and evolutionary design.

5.3.1.1 Establishing an Analytic Basis for Developing an Intelligent Computer Automated System

The goal is to establish an analytic basis for developing an intelligent computer automated system that will be able to work together with the designer during the different phases of the engineering design process—especially during the conceptual design phase when interaction and designer knowledge are sometimes more important than accuracy.

a) A Computer Automated Design Space

The core of a computer/human *design space* consists of four parts:

- The designer/design team.
- Fuzzy preference handling (for objective importance specification).
- Dynamic constraints handling (scenarios, etc.).
- Analytic module for multi-objective optimisation.

Furthermore, such a *design space* must be suited to applied *concurrent engineering design* in an integrated *collaborative design* environment in which automated continual design reviews may be conducted throughout the engineering design process by remotely located design groups. Therefore, interaction with the designer (or design team) is very important. The goal is to provide the designer with a multiple criteria decision aid for multiple criteria decision-making during the conceptual phase of the engineering design process.

The methodology is generic and could be easily integrated with other conceptual design problems. Such a computer/human *design space* is illustrated in Fig. 5.45.

b) Preferences and Fuzzy Rules

The problem of qualitative versus quantitative characterisation of the relative importance of objectives in a multi-objective optimisation framework is usually encountered during the conceptual design phase. At this initial stage of the engineering design process, it is much easier for the designer to give *qualitative* definition to the objectives (i.e. 'objective A is much more important than objective B') than to set a weighted value of objective A to, say, 0.1 or to 0.09. The method of *fuzzy preferences* and *induced preference order* is used for information transformation in which *predicates* are introduced (Fodor et al. 1994).

Table 5.23 shows the relation and intended meaning of some predicates.

These predicates, together with the complementary relations of $>$ and $\gg$, can help build the *relationship matrix R* necessary for 'words to numbers' transformation, and the induced order for the relation *R*. Integrated preferences in multi-objective optimisation techniques basically include two methods: one that uses

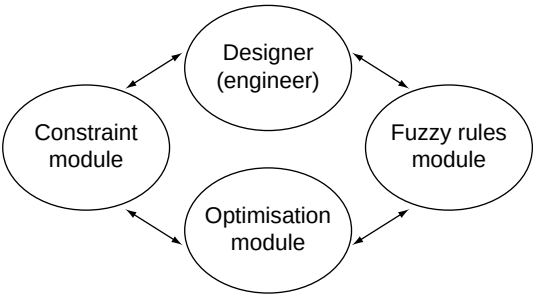


Fig. 5.45 Schema of a conceptual design space

Table 5.23 Fuzzy and induced preference predicates

Relation	Intended meaning
$\approx$	Is equally important
$<$	Is less important
$\ll$	Is much less important
#	Do not know
$\neg$	Is not important
!	Is important

weighted sums, and one that uses a *modified Pareto* method that computes the objective weights.

c) Dynamic Constraints and Scenarios

The other second tier module from Fig. 5.45 handles dynamic *constraints* and *scenarios*. Each scenario is a set of additional constraints or objectives that the designer can change, add and/or delete interactively. More formally, a scenario is represented as conjunctions of relations (constraints) in a fairly precise mathematical/modelling language. Each scenario is a function of variables, objectives and possible additional parameters. In an optimisation framework, these scenarios could return a value as a percentage of the relations satisfied for given input values. The concept behind the scenarios is that the designer can specify conditions that are not part of the mathematical model (such as 'set $y5 \in [0,4]$ or, if not possible, then set $y1 + y3 > 100$'). This allows the designer to focus on certain regions of the design space. An additional advantage is that scenarios are dynamic and are interpreted ad hoc without any change to the program or model, and can be added, modified or deleted 'online'.

Integrating scenarios in the design space provides the ability to assign a different level of importance to each scenario, and to calculate the value of a set of scenarios in different ways:

- Using weights or preferences for specifying scenario importance.
- Calculating multiple scenario values.
- Considering only one scenario at a time.

The third approach is adopted in the automated methodology presented in Sect. 5.4, as it enables the use of various imbedded software programs (analytic methods) that can analyse the various scenarios and signal any possibility or impossibility of satisfying the design constraints.

In the application of *optimisation algorithms* in artificial intelligence-based (AIB) modelling within a *blackboard model*, such as presented in Sect. 5.4, there is no need for specifying, quantitatively or qualitatively, the *importance* (as in the first method) or *order* (as in the second method) of the various scenarios.

d) The Optimisation Module

Optimisation in the early phases of engineering design represents a rather insignificant part of the overall design problem. The fuzzy nature of initial design concepts, and efficient exploration across the many different variants that the designer needs to assess are of greater interest. The methods of design preferences and scenarios are integrated with analytic techniques such as *evolutionary algorithms*, *genetic algorithms* and/or *artificial neural networks* to perform multi-objective optimisation in designing for safety.

Evolutionary computing (including evolutionary algorithms, genetic algorithms, and related models such as artificial neural networks) is based on a continuous and probabilistic representation of algorithmic optimisation (e.g. weight matrices) that would likely be able to provide the best scenario for design optimisation, in the sense that it achieves a better design with respect to performance, depending on the design problem (Cvetkovic et al. 1998).

5.3.1.2 Evolutionary Computing and Evolutionary Design

Design optimisation is a fairly common computational approach that attempts to utilise design requirements as an integral part of the design space. Design optimisation views requirements as a fixed set of criteria, and creates an evaluation function (referred to as the fitness function in artificial intelligence literature) against which the design solutions are weighed. However, design is seldom a static activity in time, especially during conceptual design. Requirements as well as design solutions change as the search for the best design progresses. This places a significant demand on the development of a suitable computational environment for interdisciplinary design collaboration in which various techniques for design concept generation as well as the evolution of design requirements and solutions are established, prompting a need for evolutionary techniques for design optimisation (Tang 1997).

The integration of evolutionary computing with artificial intelligence-based (AIB) design methodology allows for the development and integration of the basic building blocks of design (or examples of past or existing designs) that are represented in a *design knowledge base*. Several general-purpose *design knowledge sources* (or support systems) are similarly developed to support the design knowledge base. The design knowledge sources (or support systems) are developed to support the following design activities (Tang 1997):

- synthesis of conceptual design solutions from building blocks of design models and design requirements, using *inductive learning*,
- transferring conceptual design solutions into detailed design models containing spatial, geometric and structural *knowledge*,
- manipulation and partition of detailed design models into smaller design problem spaces containing suitably constrained design variables and constraints,
- searching for solutions in the partitioned design problem spaces using evolutionary computing techniques,

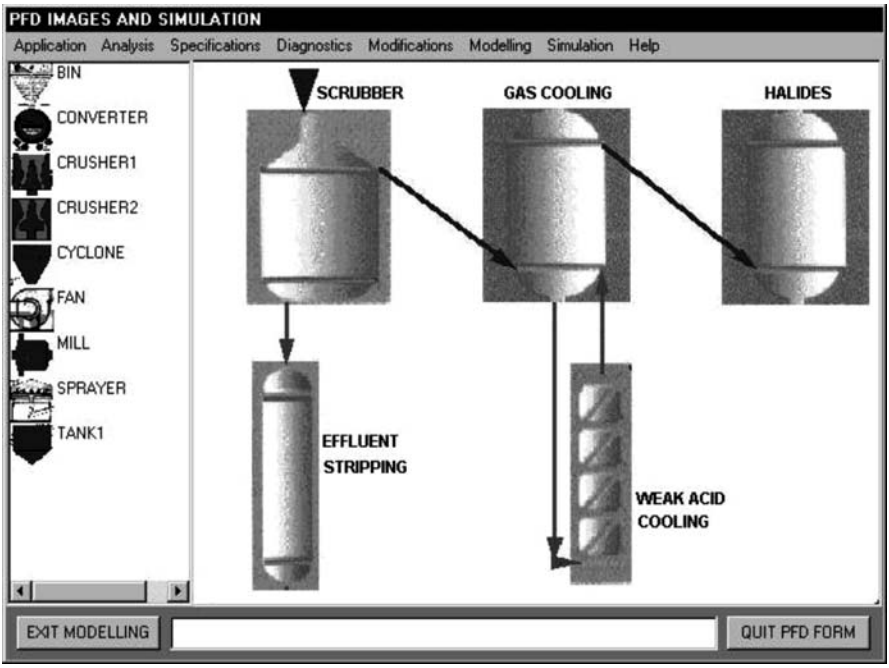


Fig. 5.46 Selecting design objects in the design knowledge base

- exploration of alternative design solutions when considering different design issues,
- documentation and explanation of design results.

The design knowledge base and design knowledge sources form the core of an integrated design support system. An artificial intelligence-based *blackboard system* is used to control the design knowledge sources and integrate the knowledge-based design applications. The design knowledge base contains design objects, constraints in terms of intended function and interfaces, as well as detailed information in terms of materials and geometry, etc.

The design knowledge base is developed by a *knowledge engineer* or by the various design teams. The design objects in the design knowledge base can be selected and synthesised to generate conceptual design solutions, as graphically indicated in Figs. 5.46 and 5.47. At an abstract level, a *conceptual design solution* identifies the basic components and their topological arrangement to the satisfaction of initial design requirements. At the early stages of the design process, many alternative conceptual design solutions must be analysed, evaluated and selected before confirming a design concept that can progressively evolve in detail for further investigation.

Once a conceptual design solution is selected, it is transformed into a *schematic design model* using the knowledge stored in advance in the design knowledge base. A schematic design model contains design variables and constraints describing the

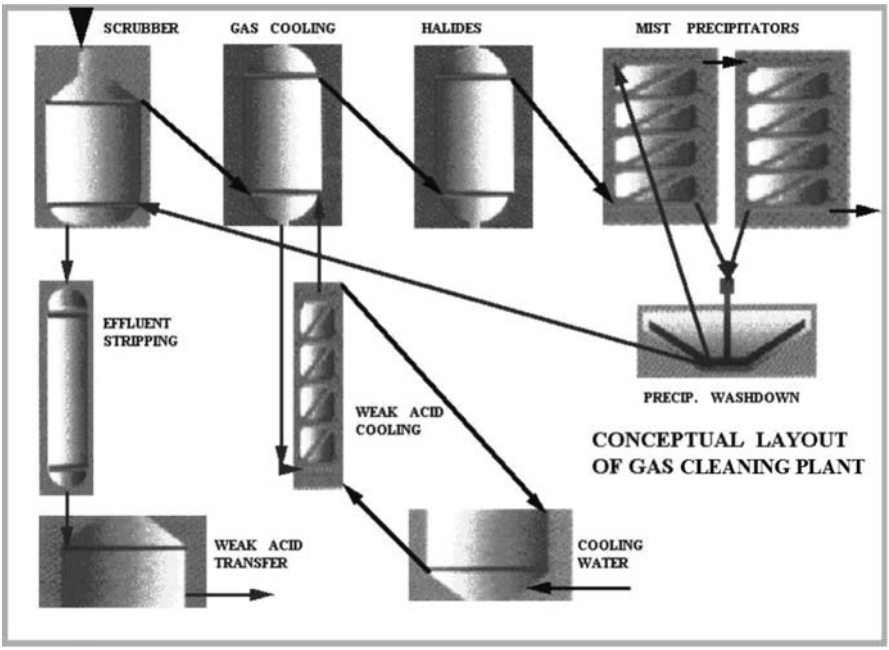


Fig. 5.47 Conceptual design solution of the layout of a gas cleaning plant

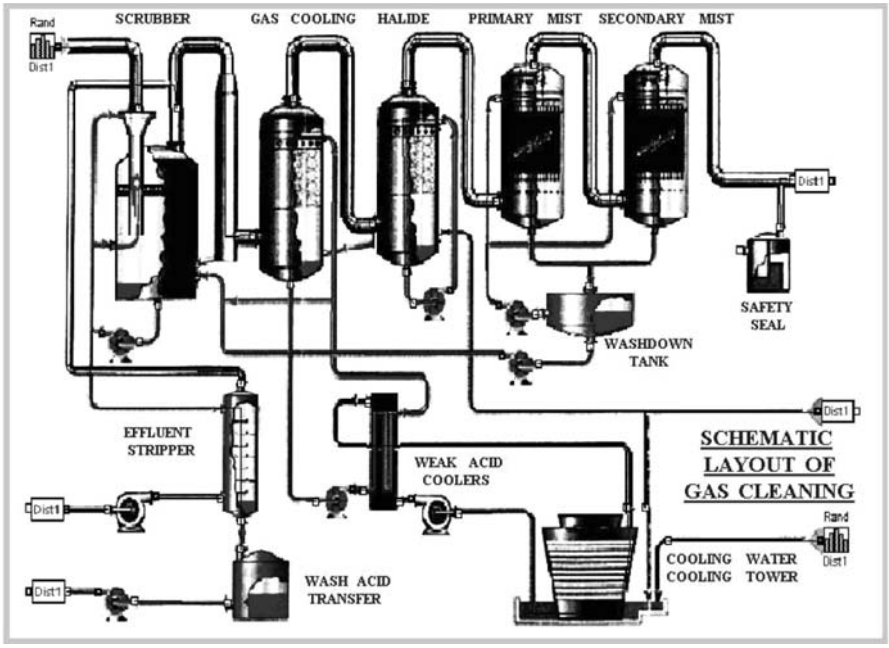


Fig. 5.48 Schematic design model of the layout of a gas cleaning plant

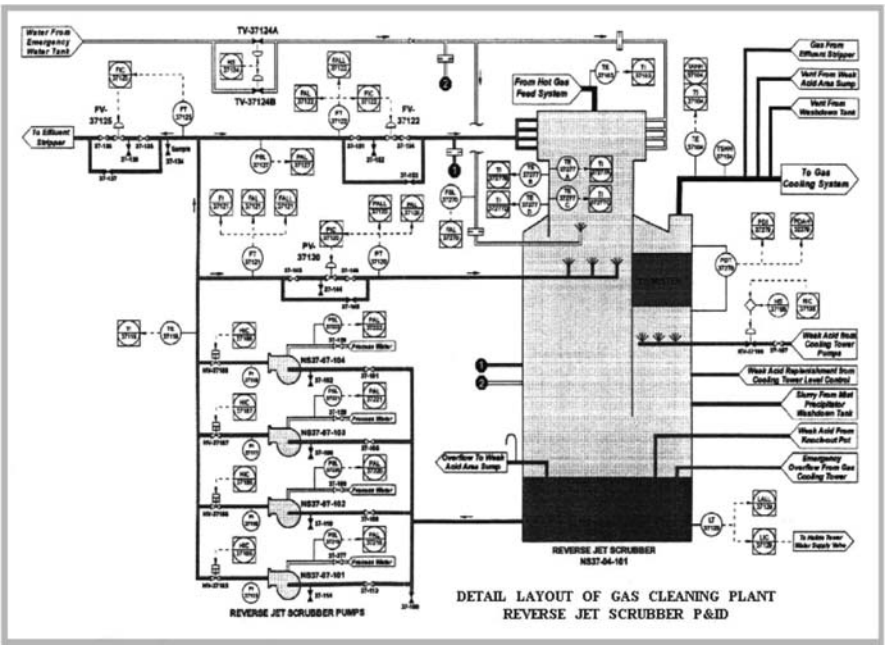


Fig. 5.49 Detail design model of the scrubber in the layout of a gas cleaning plant

structural and geometric feature of the design. A schematic design model of the gas cleaning plant is graphically illustrated in Fig. 5.48.

After evaluation of the design variables and constraints describing the structural and geometric feature of the design, a *detail design model* is prepared. In process engineering design, a detail design model typically has variables and constraints representing embodiment, structure and assembly, and dynamic flow and energy balance information of the process layout. A detail design model of the scrubber system of the gas cleaning plant is graphically illustrated in Fig. 5.49. A detail design model is computationally represented as a network of design variables and constraints that can be manipulated to identify critical equipment, for example, using *constraint-based techniques* (Smithers et al. 1990).

The network of design variables and constraints of a detail design model can be partitioned into smaller sets to identify relations. AI-based search methods, such as genetic algorithms and neural networks, can then be used to find a set of design variable values that best satisfy the constraints. This partition can be done based on the following:

- mathematical relations of the design variables,
- assembly of the detail process model,
- configuration of the systems layout,
- heuristics introduced by designers.

A partition of the constraint network identifies a small region of the design space in which, for example, design variables and constraints are evaluated to identify critical equipment in designing for safety, and explored using *evolutionary computing* techniques such as *evolutionary algorithms*.

a) Fundamentals of Evolutionary Algorithms (EA)

Evolutionary algorithms (EA) are stochastic search and optimisation heuristics derived from classic evolution theory. The basic idea is that if only those individuals of a population reproduce that meet a certain selection criteria, and the other individuals of the population die, then the population will converge to those individuals that best meet the selection criteria. If imperfect reproduction is included, the population begins to explore the search space and will move to individuals that have an increased selection probability, whereby this property is passed down to their descendants through inheritance. This population dynamics follows the basic rule of evolution theory, which can best be described as 'survival of the fittest'. To solve optimisation problems with an evolutionary heuristic, the individual items of a population group have to represent a possible solution of a given problem, and the selection probability is set proportional to the quality of the represented solution. The quality of the represented solution is termed the *fitness* (F) of the individual item.

For example, let A, B, C represent sets of items or population groups, and the current generation of the evolutionary process be indicated by s. Furthermore, a single individual item with the index i from the population A(s) is represented by $a_i(s)$. The quality of the solution represented by an individual item is termed the *fitness* F_i of the individual item $a_i(s)$. The selection probability of an individual item $a_i(s)$ is indicated by p_i. When a description of alternative solutions consists of n elements, the ith element forming a possible solution is termed *attribute* x_i. An individual item consists of several attributes x that could represent a possible solution that can then be optimised. An EA heuristic follows this scheme (Bäck 1994):

1. initialise a population group A($s = 0$)
2. evaluate fitness of all a_i from A(s)
3. select the fittest a_i as parents B(s) from A(s)
4. reproduce descendants C(s) from B(s).

Deductively, until a specific criterion is met, it thus follows that

$$\mathrm{A}(s+1) = \mathrm{C}(s) \tag{5.96}$$

In the first step, a population of random possible solutions is identified. The EA generational loop is then initialised whereby the fitness F_i for each individual item $a_i(s)$ within the current population $\mathrm{A}(s)$ is evaluated. The best individual items a_i are selected from the population $\mathrm{A}(s)$ as parents $\mathrm{B}(s)$ for the next generation. The selection probability p_i is set proportional to the fitness F_i of the individual item. From the selected parents $\mathrm{B}(s)$, descendants are reproduced to form the population $\mathrm{C}(s)$. In all EA heuristics, either the descendants are imperfect clones of the

parents with small variations, or the descendants are the product of multiple parents and inherit some attributes from the associated parents. The descendants $C(s)$ form the next generation denoted by the expression $A(s+1) = C(s)$.

A significant property of EA heuristics is that the search space is not explored by starting with only one possible solution but rather with a whole population of possible solutions, in which the individual items of a population group can interexchange solution attributes. Thus, compared with general optimisation techniques, an EA heuristic is more resistant to premature convergence towards a local optima in the search space.

Evolutionary computing techniques address design problems as a goal-directed search problem. This evolutionary approach is useful in engineering design applications. In such applications, the goal is to minimise the number of constraints that are violated in a particular design solution. The process of exploring a design solution involves symbolic computation in terms of constraint propagation and satisfaction. This exploration process is common to most engineering design domains. However, whilst the evolutionary computing approach relies more on automatic formation and evolution of design concepts, the EA heuristic approach emphasises the use of symbolic computation and heuristic-based evaluation and selection of a potentially large number of solutions, before any automatic searching methods are used (Tang et al. 1997).

The latter is particularly important in engineering design where the search space for a design solution needs to be confined to a small region. The EA heuristic approach can be usefully employed, applying specific techniques such as *genetic algorithms* that seamlessly scale between the exploration of the search space through genetic *crossover* and *mutation*, and the exploitation of known optima through the selection of fit individual items.

b) Fundamentals of Genetic Algorithms (GA)

Genetic algorithms (GA) originated from the work of John Holland and exhibit the most obvious mapping of natural evolutionary processes into a computer system, because they focus on the coding of *attributes* into a set of *genes*. The most common method of coding attributes is *binary coding* into a bit-string that represents these genes. Thus, some biological terms are used to illustrate the functionality of genetic algorithms (Holland 1992).

GA individual items GA individual items store the solution attributes in a coded representation. The most common representation is the binary coding of an attribute in a chain of bits. The bit-string consists of L number of bits, which are clustered into meaningful data representing information typically in the form of semantics, such as words, w_i. The decoded words w are the solution attributes x, which are to be optimised. Each attribute x_i is assigned to the word w_i. In the simplest case, a word codes a real number. In this case, the real number attributes are limited to a range of values, since the length l of a word is always limited.

If the range of an attribute and the length l of a word w_i are given, then the attribute is fixed to a real number. This coding is called standard binary coding. Similar coding styles can be found for nearly every data type that can be used as an attribute of a GA individual item, and can thus be optimised using the GA search heuristic. After the attributes x_i of an individual item have been determined—for instance, by decoding the word w_i—the fitness F_i can be calculated by using a target function $F(x_i)$ as the fitness function. After the fitness for every individual a_i of the population $\mathrm{A}(s)$ has been calculated, the best individual items of the population group $\mathrm{A}(s)$ are selected to be the parents of the next generation $\mathrm{A}(s+1)$. This is called *Holland's fixed-length coding* (Holland 1992).

The main advantages of genetic algorithms are that they are very easy to implement and they can be applied to nearly every kind of optimisation problem. Because of the general binary coding style, almost any data type can be stored in an individual item and then be optimised by the GA heuristic. However, there are also some drawbacks using binary coding. For example, if real numbers are used as attributes, they become discretised (i.e. distinctively separate) and, because of the non-linear behaviour of standard binary coding, the search space for a design solution that is confined to a small region can get disrupted.

5.3.2 Analytic Development of Safety and Risk Assessment in Preliminary Design

For safety systems of which the failure could result in loss of life, it is imperative that the best use is made of systems that are optimal and not just adequate, and that a design optimisation scheme is applied for systems that require a high likelihood of functional reliability on demand. Considering a more advanced analytic development of safety and risk assessment in preliminary design, a *genetic algorithm (GA)* is used to perform design optimisation, resulting in a design specification for later evaluation during the detail design phase. Analyses of system designs are carried out using the latest advances in *fault-tree analysis (FTA)*, utilising the *binary decision diagram (BDD)* approach whereby the method can be applied to high-integrity protection systems (HIPS). Varying parameters, which inevitably affect the action of the GA, are thus considered to determine areas where the application of genetic algorithms for safety and risk assessment in preliminary design could be improved.

5.3.2.1 Genetic Algorithms in Optimal Safety System Design

Failure of a safety system for a potentially hazardous industrial process may have severe consequences, possibly resulting in personal injuries or loss of life. It is therefore imperative that such systems have a high likelihood of functioning on demand. One measure of system performance is the probability that the system will fail to operate when required. Typically, the preliminary design of a safety system follows

the traditional design process of analysis, assessment, appraisal and redesign. If, following analysis, the preliminary design does not meet some predetermined acceptability target for system reliability, then deficiencies in the design are removed, and the assessment and appraisal stages are repeated.

Once the predicted system reliability of a design reaches the acceptable criteria, the design process stops and the system is adopted. For a system of which the failure could result in fatality, it would inevitably be considered that a merely adequate level for system reliability is not sufficient. It is highly unlikely, however, that the design parameters can be manually selected such that optimal system performance can be achieved within the set design criteria and constraints.

An approach by which optimal performance can be obtained, using the fault-tree analysis (FTA) method to determine the availability of each system design, was previously described in Sect. 5.2.3.2 dealing with design optimisation in designing for safety. The method is in the form of an iterative scheme that produces a sequence of system designs gradually improving the safety system performance. When the design can no longer be improved due to restrictions of the design criteria constraints, the optimisation procedure terminates (Andrews 1994).

An alternative methodology is presented (Andrews et al. 1997), which incorporates the latest advances in the fault-tree analysis technique, based on binary decision diagrams and utilising a genetic algorithm (GA) to perform the optimisation (Painton et al. 1995).

Further research into utilising a genetic algorithm to perform design safety optimisation considers the effects of modifying the GA process and the parameter values used, in order to make the GA process more accurate and effective (Pattison et al. 1999).

a) Safety Design Considerations

Safety systems are designed to operate when certain conditions occur, and to prevent their development into a hazardous situation. Where possible, safety systems should not be designed so that single component failures can prevent the system from functioning. To ensure this, several options are available (Pattison et al. 1999):

- Redundancy or diversity can be incorporated into the system. Redundancy duplicates elements within a system, while diversity involves the addition of a totally different means of achieving the same function.
- Component selection is another design option. Each component selected for the design is chosen from a group of possible alternatives. The design engineer must decide how to trade off the specific characteristics of each component to give the most effective overall system performance.
- The time interval between preventive maintenance activities is a further consideration. This is generally assigned on an ad hoc basis after the design has been fixed. Significant gains are to be made by considering the maintenance frequency at the design stage.

The choice of design is not unrestricted, in that practical considerations place limits on resources both during the design stage as well as in the later stages of the engineered installation, preventing a completely free choice of system design and rendering some design variations infeasible.

b) The Design Optimisation Problem

The objective of the design optimisation problem is to maximise design integrity by minimising system unreliability and unavailability through manipulation of the design variables such that constraint propagated limitations are not violated. Different optimisation approaches to determine optimal design solutions have included *dynamic programming*, *integer programming*, *mixed integer programming*, as well as *non-linear programming* and *heuristics*. Dynamic programming in this context is applicable to maximise reliability for a system given a single cost constraint in which the problem is to identify the optimal levels of redundancy (Bellman et al. 1962).

The dynamic programming approach can also be applied to more difficult design problems in which a system has multiple sub-systems and components, each with constraints on cost and weight. For each sub-system, several component choices are made with different reliability, cost and weight. However, to accommodate such multiple constraints, the use of a Lagrangian multiplier within the objective function is essential (Fyffe et al. 1968). While such a formulation provides a selection of different components, the search space is restricted to consider only solutions where identical components are in parallel. The use of a Lagrangian multiplier with dynamic programming is, however, often inefficient, necessitating the use of a surrogate constraints approach (Nakagawa et al. 1981).

An alternate approach to the design optimisation problem has been to use integer programming. In applying integer programming, it is necessary to restrict the search space and prohibit mixing of different components within a sub-system. To maximise reliability given non-linear but separable constraints, many variations of the problem can be transformed into an equivalent integer programming problem, using a branch-and-bound approach (Ghare et al. 1969). The design optimisation problem can also be formulated as a multi-objective decision-making problem with distinct goals for reliability, cost and weight (Misra et al. 1991). There have been several effective uses of mixed integer and non-linear programming to solve the redundancy allocation problem in optimising a specific design. In these problems, component reliability is treated as a continuous variable, and component cost is expressed as a function of reliability and several other parameters (Tillman et al. 1977).

While the redundancy allocation problem in design optimisation has been studied in great detail and, in practice, many system designs use multiple different (yet functionally similar) components in parallel, two areas that have not been sufficiently analysed are the implications of mixing functionally similar components within a parallel sub-system, and the use of k-out-of-n: G redundancy ($k > 1$). A typical example is the determination of solutions to the redundancy allocation problem

for a system design comprising series-parallel components in a high-integrity protection system (HIPS). In such cases, use of genetic algorithms (GAs) in design optimisation is most appropriate. The power of genetic algorithms is that they can easily be adapted to diverse design scenarios including those with functionally similar components, k-out-of-n: G redundancy, and more complex forms of redundancy.

c) Genetic Algorithms (GAs)

The use of genetic algorithms (GAs) in designing for safety in process engineering systems is a new approach to determining solutions to the redundancy allocation problem for a series-parallel system design comprising multiple components in a high-integrity protection system (HIPS). In such design problem formulations, there are specified numbers of sub-systems and, for each sub-system, there are multiple component choices that can be selected and used in parallel. For designed systems using off-the-shelf component types, with known cost, reliability and weight, system design and component selection become a combinatorial optimisation problem where new system designs are composed largely of standard component types (pressure sensors, pressure control valves, etc.) with known characteristics. The problem is then to select the optimal combination of components with specific levels of redundancy, to collectively meet reliability and weight constraints at a minimum cost or, alternatively, to maximise reliability given cost and weight constraints.

The GA optimisation approach is one of a family of heuristic optimisation techniques that has been demonstrated to converge to the optimal solution for many diverse, difficult problems, although optimality cannot always be guaranteed. The ability of the GA to efficiently find good solutions often depends on properly customising the encoding, operators and fitness measures to the specific engineering design problem. Genetic algorithms have been used to solve many difficult combinatorial optimisation problems with large and complex search spaces.

For a fixed design configuration and known incremental decreases in component failure rates and their associated costs, a GA can be used to find maximum reliability solutions to satisfy specific cost constraints. The algorithm can be formulated to optimise reliability, mean time between failure (MTBF), and availability (Painton et al. 1995).

Genetic algorithms have also been used in the analysis of series-parallel systems with multiple sub-systems and unique component choices for each sub-system (Coit et al. 1994), and to find solutions to the redundancy allocation problem where there are several failure modes (Ida et al. 1994). An interesting feature of this work, which will be considered in greater detail in a later section, is the use of neural network approximations to sub-system reliability, instead of exact solutions.

The GA methodology A genetic algorithm (GA) is a stochastic optimisation technique patterned after natural selection in biological evolution (Goldberg 1989). The main advantage of the GA is that there are very few restrictions on the form of the solutions. The GA thoroughly examines the search space, and readily identifies

design configurations that will improve the final solution but would not be identified using prior dynamic programming, integer programming or non-linear programming formulations of the same design optimisation problem. The GA methodology is characterised by:

- Encoding of solutions.
- Generation of an initial population.
- Selection of parent solutions for breeding.
- Crossover breeding operator.
- Mutation operator.
- Culling of inferior solutions.
- Iteration, i.e. repetition of steps 3–6 until termination criteria is met.

An effective GA depends on complementary *crossover* and *mutation* operators. The effectiveness of the crossover operator dictates the rate of convergence, while the mutation operator prevents the algorithm from prematurely converging to local optima. The number of children and mutants produced with each generation are variable parameters that are held constant during a specific trial (Smith et al. 1996).

Solution encoding Traditionally, solution encoding has been a binary string, as considered later in the example. For combinatorial optimisation, however, an encoding using integer values can be more efficient. Each possible solution to the redundancy allocation problem can be viewed as a collection of components in parallel for each sub-system. The *selected components* can be chosen in any combination from among the available components. These *selected components* are indexed in descending order of reliability (1 being the most reliable, etc.). The solution encoding is a vector representation in which each of the sub-systems is represented by the *selected components*, which form a particular solution and are listed according to their reliability index. The sub-system representations are then placed adjacent to each other to complete the vector representation.

As an example, consider a system with sub-systems $s = 3$, with available components for each sub-system equating to $m_1 = 5$, $m_2 = 4$, $m_3 = 5$, and the maximum number of components predetermined to be $n_{\max} = 5$. The solution string $v_q = (11\,666|22\,355|46\,666)$ represents a prospective solution with two of the most reliable components used in parallel for the first sub-system, two of the second most reliable, and one of the third most reliable components used in parallel for the second sub-system, and one of the fourth most reliable components used for the third sub-system. Certain assumptions are inevitably made, typically:

- The component reliabilities are known and deterministic.
- Failures of individual components are independent.
- All redundancy is active redundancy without repair.

Initial population In general, the minimum effective population size would grow with problem size. For example, for a given population size P, the initial population is determined by randomly selecting p solution vectors. For each solution, s integers are randomly selected to represent the number of components in parallel n_i for

a specific sub-system. Then, n_i components are randomly and uniformly selected from among the m_i components that are available. The *selected components* are sequenced in accordance with their reliability.

Objective function A typical objective function of the redundancy allocation problem in design optimisation is the sum of the reliability or cost, and a dynamic penalty function determined by the relative degree of infeasibility of the solution set. Thus, in the specific case of a redundancy allocation problem for a series-parallel system, the problem formulation is to *maximise* reliability (problem P1) or to *minimise* cost (problem P2), given that these constraints are specified for each sub-system.

This is given in the algorithmic expressions of

Problem P1:

$$\max \prod_{i=1}^{x} R_i(x_i|k_i) \tag{5.97}$$

Problem P2:

$$\min \sum_{i=1}^{s} C_i(x_i) \leq C \tag{5.98}$$

$$\sum_{i=1}^{s} W_i(x_i) \leq W$$

where:

$R_i(x_i|k_i)$ = reliability of sub-system i, given k
$C_i(x_i)$ = total cost of sub-system i
$W_i(x_i)$ = total weight of sub-system i
k_i = minimum number of components in parallel required for sub-system i to operate.

Within the two problem formulations, system weight and cost are often defined as linear functions because this is a reasonable representation of the cumulative effect of component cost and weight. Using probability principles, it can be shown that system reliability can be expressed as a function of the decision variable x_i, as indicated in Eq. (5.99) below. However, with such a general form of system reliability, it is not possible to determine a linearly equivalent objective function or constraint, as is done in integer programming formulations.

$$R_s(x_1, x_2, \ldots, x_s|k) = \prod_{i=1}^{x} R_i(x_i|k_i) \tag{5.99}$$

Dynamic penalty function It is important to search through the *infeasible region* of the solution set, particularly for highly constrained problems, because in most cases the optimal solution can efficiently be reached via the infeasible region and, often, good feasible solutions are a product of breeding between a feasible and an infeasible solution.

To provide an efficient search through the infeasible region, but to ensure that the final best solution is feasible, a *dynamic penalty function* based on a significant criterion is defined to incrementally increase the penalty for infeasible solutions.

For cost minimisation (problem P2), the objective and penalty functions are defined as follows

$$f(\lambda, v_q) = \sum_{i=1}^{s} C_i(x_i) + P(\lambda, v_q) \tag{5.100}$$

where:

s = total number of sub-systems
λ = Lagrangian multiplier vector
v_q = vector encoding of solution q
$f(\lambda, v_q)$ = fitness for the q member of the population
$C_i(x_i)$ = objective function for total cost of sub-system i
$P(\lambda, v_q)$ = penalty function for q member of the population.

Crossover breeding operator The crossover breeding operator provides a thorough search of areas of the sample space that demonstrate to produce good solutions. For the developed GA, parents are selected based on the ordinal ranking of their objective function. A uniform random number U, between 1 and p, is selected, and the solution with the ranking closest to U is selected as a parent, following an appropriate selection procedure (Smith et al. 1993). The crossover operator retains all identical genetic information from both parents, and is then randomly selected with equal probability from either of the two parents for components that differ. Because the solution encoding is ranked, matches will inevitably be found.

Mutation operator The mutation operator performs random perturbations to selected solutions. A predetermined number of mutations within a generation is set for each GA trial. Each value within the solution vector, which is randomly selected to be mutated, is changed with probability equal to the mutation rate. A mutated component is changed with 50% probability, and a randomly chosen component with 50% probability (Smith et al. 1996).

Evolution A survival of the fittest strategy is employed. After crossover breeding, the p best solutions from among the previous generation and the new child vectors are retained to form the next generation. The fitness measure is the objective function value.

Mutation is then performed after culling inferior solutions from the population. The best solution within the population is never chosen for mutation, to ensure that the optimal solution is never altered via mutation. The GA is terminated after a preselected number of generations, although the optimal solution is usually reached much earlier.

d) Systems Analysis with GAs and Fault Trees

Commonly with mathematical optimisation problems, such as linear programming, dynamic programming and various other optimisation techniques, an explicit objective function is derived that defines how the characteristic to be minimised is related to the variables. However, in many design optimisation problems, an explicit objective function cannot be formulated, and system performance is assessed by *fault-tree analysis (FTA)*. This is often the case in safety systems design. The nature of the design variables also adds difficulty to the problem. Design variables that represent the levels of duplication for fully or partially redundant systems, as well as the period between preventive maintenance, are all integer. Furthermore, selecting component types is governed by Boolean variables, i.e. selection or non-selection.

A numerical scheme is therefore required that produces integer values for these variables, since it will not be appropriate to utilise a method where real numbers are rounded to the nearest whole number. Constraints involved in this problem fall into the category of either explicit or implicit constraints. Expected maintenance downtime, for example, can be represented by an explicit function of the design parameters; however, the number of spurious process trips can be assessed only via a full analysis of the system, which will again require employment of the fault-tree analysis methodology. As no explicit objective function exists for most preliminary designs of safety systems, particularly in redundancy allocation problems for design optimisation, fault trees are used to quantify system unreliability and/or unavailability of each potential design. It is, however, a time-consuming and impractical task to construct a fault tree for each design variation, especially at the lower systems hierarchy levels. To resolve this difficulty, *select events* can be used to enable the construction of a single fault tree capable of representing causes of the system failure mode for each possible system design. Select events in the fault tree, which are either TRUE or FALSE, are utilised to switch on or off different branches to model the changes in the *causes* of failure for each design alternative.

As an example, consider the choice of a valve type, from the possible alternative valves V_1, V_2 or V_3 in a safety system (Pattison et al. 1999). The fault tree is shown in Fig. 5.50.

If valve type V_1 is selected, the *select event* H_1 corresponding to the selection of this valve is set to TRUE. Select events H_2 and H_3, corresponding to the selection of V_2 and V_3, are conversely set to FALSE. A contribution to the top event arises from the left-most branch only. The two right-most branches are, in effect, switched off. Levels of redundancy are handled similarly. Furthermore, the spurious trip frequency for each design is an implicit constraint that requires the use of fault-tree analysis to assess its value. Select events are again used to construct a fault tree capable of representing each potential design for this failure mode.

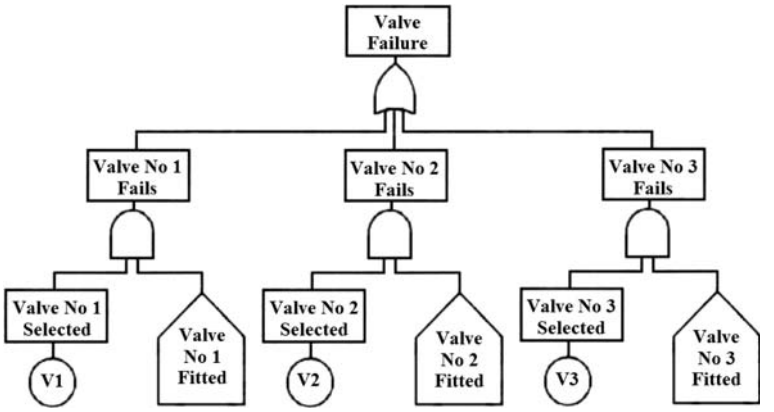


Fig. 5.50 Fault-tree structure for safety valve selection (Pattison et al. 1999)

e) Algorithm Description Using Binary Decision Diagrams

A *binary decision diagram (BDD)* is a type of oriented graph used notably for the description of algorithms. It basically consists of two types of nodes: the decision or test node, and the output node. The decision node is equivalent to an if-then-else instruction that realises a test on a binary variable and, according to this value, indicates the following node. The output node produces a value.

There are two rules of BDD assemblage, namely that there is one and only one initial node (the entry point of the algorithm), and that the output point of a node can be connected to only one entry point of another node. More precisely, a BDD is a rooted, directed, acyclic graph with an unconstrained number of in-edges and two out-edges, one for each of the 1 and 0 decision paths of any given variable. As a result, the BDD has only two terminal nodes, representing the final value of the expression, 1 or 0—although occasionally the zero (false) node and edges leading to it are omitted (Akers 1978; Bryant 1986).

To improve efficiency of analysis, the *binary decision diagram (BDD)* method is used to solve the resulting fault tree. The BDD is composed of terminal and non-terminal vertices that are connected by branches in the diagram. Terminal vertices have the value of either 0 or 1, whereas the non-terminal vertices correspond to the basic events of the fault tree. Each vertex has a 0 branch that represents the basic event of non-occurrence (i.e. it works), and a 1 branch that represents the basic event of occurrence (i.e. it fails). Thus, all paths through the BDD terminate in one of two states—either a 1 state, which corresponds to system failure, or a 0 state, which corresponds to system success. The BDD represents the same logical function as the fault tree from which it is developed; however, the BDD produces more accurate results. As an example, consider the BDD illustrated in Fig. 5.51.

The fault-tree structures for each system failure mode are converted to their equivalent BDD. Analysis of a BDD has proven to be more efficient than the quantification of the fault-tree structure because evaluation of the minimal cut sets for

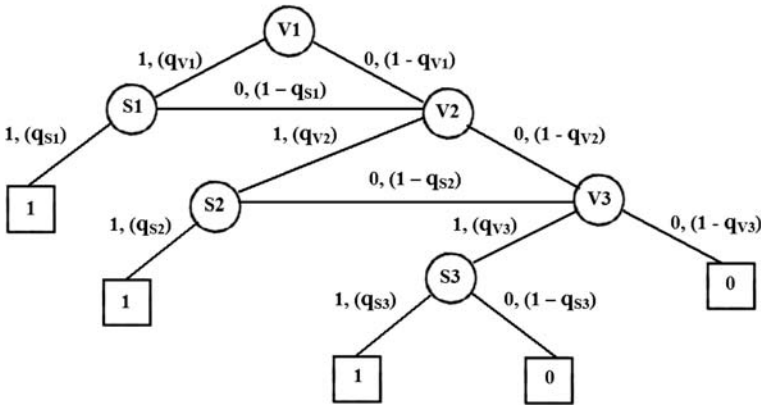


Fig. 5.51 Binary decision diagram (BDD) for safety valve selection

quantification is not required. For the purpose of BDD construction, select events in the fault tree are treated as basic events. Using this process, the fault tree for the component design variables that is shown in Fig. 5.50 is represented by the BDD in Fig. 5.51.

The quantity q appearing on the 1 and 0 branches developed from each node in Fig. 5.51 represents the probability of each path. The select events are turned on or off by setting their probability to 1 or 0 respectively. Consider, for example, the design where valve 1 has been selected for the fault tree shown in Fig. 5.50. This is then presented by $S_1 = 1$, $S_2 = 0$, $S_3 = 0$ for the *select events* and, hence, the corresponding probabilities $q_{S_1} = 1$, $q_{S_2} = 0$ and $q_{S_3} = 0$ are set on the equivalent BDD. The only path to a terminal 1 node leaves V_1 and S_1 on their 1 branches, which have probability q_{V_1}. The probability values assigned to each *select event*, which are determined by a particular design, are automatically assigned to the BDD. Thus, the major advantage of the BDD is its practicality.

f) Example of Genetic Algorithm Application

As an example, the BDD methodology is applied to a high-pressure protection system. The example is taken from Sect. 5.2.4.2 dealing with the structuring of the cause-consequence diagram, in which the CCD diagramming technique was applied to the simple high-pressure protection system as depicted in Fig. 5.34. The features of this high-integrity protection system (HIPS) are shown in Fig. 5.52.

The function of the protection system is to prevent a high-pressure surge originating from process circulation pumps, to protect equipment located downstream of the process. Returning to the previous example, the basic functions of the components of the system are shown in Table 5.1. The first level of protection is the emergency shutdown (ESD) sub-system with its specific pressure control valves (PCVs). Pressure in the pipeline is monitored using pressure transmitters (P_1, P_2 and P_3). When

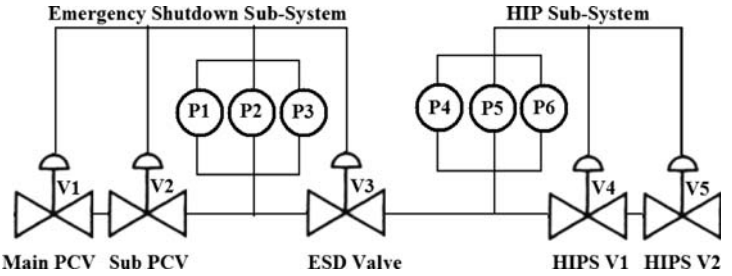


Fig. 5.52 High-integrity protection system (HIPS): example of BDD application

Table 5.24 Required design criteria and variables

Design criteria	Design variable
How many ESD valves are required? (0, 1, 2)	E
How many HIPS valves are required? (0, 1, 2)	H
How many pressure transmitters for each sub-system? (1, 2, 3, 4)	N_1, N_2
How many transmitters are required to trip?	K_1, K_2
Which ESD/HIPS valves should be selected?	V
Which pressure transmitters should be selected?	P
What should the maintenance interval be for each sub-system?	θ_1, θ_2

the pipeline pressure exceeds the permitted value, the ESD system acts to close the main PCV (V_1) and sub-PCV (V_2), together with the ESD valve (V_3). To provide an additional level of protection for high integrity, a second level of redundancy is incorporated by inclusion of a high-integrity protection system (HIP sub-system). This works in a similar manner to that of the ESD system but is completely independent in operation with its specific pressure control valves, HIPS V_1 (V_4) and HIPS V_2 (V_5). Even with a relatively simple system such as this, there are a vast number of options for the engineering designer to consider. In this example, it is required to determine values for the design variables given in Table 5.24.

Several constraints have been placed on the design criteria, as follows:

- The total system cost must be minimised.
- Spurious system shutdowns would be unacceptable if this was more than once per year.
- The average downtime per year owing to preventive maintenance must be minimised.

Genetic Algorithm Implementation As previously indicated, genetic algorithms (GAs) belong to a class of robust optimisation techniques that use principles mimicking those of *natural selection* in genetics. Each individual design at assembly level, and at component level where such components have been identified in the preliminary design phase, is coded as a string of parameter values where each string is analogous to a *chromosome* in nature. The GA method is then applied with a population of strings, each string being assigned a measure of its *fitness*. Selection (or

reproduction, as it is termed in genetics) then exploits this fitness measure. The greater the fitness value, the higher is the string's chance of being selected for the next generation.

The entire process is influenced by the action of the *genetic operators*—typically, *crossover* and *mutation*. Crossover involves crossing information between two solution strings that are already selected to enter the next generation. Mutation is the alteration of a parameter value on the solution string. Both operators enable exploration of different system designs.

To specify a safety system design, a value is assigned to each of the ten design variables given in Table 5.24. These values are then expressed in binary form, such as a string of binary digits. Each variable is given a particular length, in order to accommodate the largest possible value in binary form. In total, each string representing the design variables can be interpreted as a set of concatenated integers coded in binary form. However, the restricted range of values assigned to each parameter does not in each case correspond to the representative binary range on the solution string. For this reason, a procedure is used to code and, in subsequent generations, to check the feasibility of each string.

Evaluating String Fitness Constraints are incorporated into the optimisation by *penalising* the fitness when they are violated by the design.

The fitness of each string consists of four parts (Pattison et al. 1999):

1. Probability of system failure unreliability.
2. Penalty for exceeding the total cost constraint.
3. Penalty for exceeding the total maintenance downtime constraint.
4. Penalty for exceeding the spurious trip constraint.

The result is a *fitness value* for each design, which can be referred to as the penalised system unreliability of design. Calculating this system unreliability involves derivation of the penalty formula for excess cost, spurious trip occurrences, and maintenance downtime. If a particular design exceeds any of the stated limits, the respective penalty is added to the system unreliability of design. The formula used for the penalty function is described later. The penalised probability of system unreliability is thus calculated using the following expression

$$Q'_{\mathrm{s}} = Q_{\mathrm{s}} + C_{\mathrm{P}} + T_{\mathrm{P}} + D_{\mathrm{P}} \tag{5.101}$$

where:

Q'_{s} = penalised probability of system unreliability
Q_{s} = un-penalised prob. of system unreliability
C_{P} = penalty due to excess cost
T_{P} = penalty due to excess spurious trips
D_{P} = penalty due to excess maintenance downtime (DT).

Derivation of the Penalty Formula If the performance of a design is significantly improved owing to comparatively small excesses in one or more of the constraints, the specific design deserves further consideration. Conversely, excessive abuse of

the limits with only a small degree of performance improvement implies that the design be discarded.

It is essential that an appropriate penalty be applied to the system unreliability when constraints are violated. For example, a spurious trip can affect the reliability of the system *and* result in a loss of production. For this reason, a spurious trip is expressed in terms of unreliability and cost. This is achieved using a multiplying factor that, rather than being fixed, varies according to the system unreliability of the design, as indicated in (Eq. 5.102) below.

A *penalty function* based on *sub-system* unreliability and cost is defined to incrementally increase the penalty. This requires careful consideration of unreliability and cost minimisation of the design being penalised, where the objective and penalty functions are defined as follows

$$f_{\text{System}} = \sum_{i=1}^{s} [1 - R_i(x_i)] - C_i(x_i) \tag{5.102}$$

where:

s = total number of sub-systems
x_i = decision variable relating to system reliability
f_{System} = fitness function for system unreliability and cost
$R_i(x_i)$ = objective function for total reliability of sub-system i
$C_i(x_i)$ = objective function for total cost of sub-system i.

In this expression of the fitness function, the relationship between unreliability and excess cost is assumed to be linear. However, although small excesses in cost may be tolerated, as the extra cost becomes much larger its feasibility should significantly decrease. For this reason, an exponential relationship is preferred for the objective function for the total cost of sub-system i, as given in (Eq. 5.102).

To illustrate this, consider a particular design in which a base level in system performance is assumed and an unreliability value of 0.02 (i.e. 0.98 or 98% reliability) for the system is considered reasonable. Should the cost of a design exceed a certain base level (say, 1,000 units), the excess cost percentage should be reflected in the system unreliability as a corresponding percentage improvement about that base level. If the relationship between unreliability and excess cost is assumed to be linear, a design that costs 1,100 units should show an improvement of at least 0.002 in unreliability (i.e. 10%). However, the multiplying factor of 0.002, or 10% of the base level performance, is the area of concern if the value is a fixed percentage of system unreliability. With such a fixed multiplying factor, the penalty formula does not properly take into account system unreliability of comparative designs that are being penalised.

To further illustrate this, consider the following example: design A costs 1,100 units and has an un-penalised system unreliability of 0.02 (reliability of 0.98 or 98%). The objective function for total system cost is given as the exponential relationship of the ratio of total system costs to a base cost of 1,000 units, which is modified by the power $5/4$ (Pattison et al. 1999).

This is expressed as

$$\sum_{i=1}^{s} C_i(x_i) = \frac{[C_s]}{C_b}^{5/4} \tag{5.103}$$

Applying the penalty function formula of (Eq. 5.102) then gives the following:

$$f_{\text{System}} = 0.02 \times \frac{[1{,}100]}{1{,}000}^{5/4} = 0.0225$$

The cost penalised fitness value is 0.0225, a fitness decrement of approximately 25% compared to the un-penalised unreliability of 0.02.

Design B costs 1,150 units but has an un-penalised system unreliability of 0.015 (i.e. reliability of 0.985 or 98.5%). Applying the penalty formula gives a cost penalised fitness value of 0.018, a fitness decrement of approximately 20% compared to the un-penalised unreliability of 0.015. The comparative cost penalty for the fitter string (design A) is thus greater by 5% (25–20%). The difference in un-penalised system reliability between design A and design B is only 0.5%. Thus, the penalty should take the fitness value of the system to be penalised into consideration. Consider, therefore, a particular design with cost C. The percentage excess of the system's cost is calculated as X_c. The multiplying factor is then derived by calculating X_c percent of the system unreliability of the engineering design under consideration.

Reproduction probabilities The fitness value, or penalised system unreliability, is evaluated for each string. For the purpose of selection in the GA, each string is assigned a *reproduction probability* that is directly related to its fitness value. In the safety system optimisation problem, the smaller the fitness value, the fitter is the string and, hence, the greater should be its chance of reproduction. For cases such as these, a possible approach is to let the reproduction probability be one minus the fitness value. However, the penalised system unreliability fitness values may result in all reproduction probabilities of a string having similar values, thereby detracting from the specific fitness information available to the GA. A more objective method is required that retains the accuracy of each string's fitness value during conversion to its corresponding reproduction probability.

Converting the fitness value Each design receives a measure of its fitness. This is the design string's penalised system unreliability. However, this value is not in an appropriate form to be used directly in the selection process of the GA, since the smaller the fitness value, the better is the design. A specialised conversion method is required that gives rise to weighted percentages in accordance with the fitness value of each string. A system with a performance twice as good as that of another should have twice the percentage allocation.

One conversion method is to allocate each string to one of three categories according to its fitness value. Strings in category 1 are automatically given 0%, as this category consists of poor system designs and these are eliminated from the succeeding generation. Strings in category 2 contain relatively unfit designs, and are allocated a relative portion up to a total of 5%. The strings that fall into category 3 are of

ultimate interest. The remaining 95% is then allocated to each string, depending on how much their fitness value exceeds a base limit of 0.1. The percentage allocated to each category is fixed and, therefore, independent of the number of strings that it contains. Problems occur, however, when a very high or a very low proportion of strings fall into a particular category, and an improved method is required that is able to cope with very diverse populations and simultaneously to show sensitivity to a highly fit set of strings. This is done by proportioning the percentage allocation for a category by a weighted distribution of the fitness value of each string in the category and the number of strings it contains.

GA parameters The GA requires the following selection parameters to be set:

- population size,
- crossover rate,
- mutation rate and
- number of generations.

The values entered for these parameters have a marked effect on the action of the GA and on the penalised system unreliability of the best overall string for each parameter set. To obtain an indication of the effect of setting each parameter to a particular value, the penalised system unreliability obtained is summed and averaged against results obtained for the mutation rate, crossover rate and population size for the example GA.

g) Results of the GA Methodology

The simple example GA is a very effective design tool in its application to the high-pressure protection system shown in Fig. 5.47. The modified cost penalty and the modified conversion method established the preferred GA methodology. This modified GA demonstrates the ability to find and explore the fittest areas of the search space and it is able to differentiate between highly fit strings as the algorithm progresses, whereby retention of the best design over later generations is achieved. Using the modified GA, the characteristics of the best design obtained for the design variables given in Table 5.24 are represented in Table 5.25.

Table 5.25 GA design criteria and variables results

Design criteria	Design variable	Sub-system	
		ESD	HIPS
How many ESD valves are required? (0, 1, 2)	E	0	–
How many HIPS valves are required? (0, 1, 2)	H	–	2
How many transmitters per sub-system? (0, 1, 2, 3, 4)	N_1, N_2	4	4
How many transmitters are required to trip?	K_1, K_2	1	2

5.3.3 Analytic Development of Safety and Risk Evaluation in Detail Design

The engineering design process presents two fundamental problems: first, most engineering systems have complex, non-linear integrative functions; second, the design process is fraught with uncertainty, typically when based on iterative evolutionary computational design. This trial and error feedback loop in detail design evaluation needs to be tightened by improving design analysis before the onset of system manufacturing or construction (Suri et al. 1989).

Artificial neural networks (ANN) offer feasible solutions to many design problems because of their capability to simultaneously relate multiple quantitative *and* qualitative variables, as well as their ability to form models based solely on minimal data, rather than assumptions of linearity or other static analytic relations. Basically, an artificial neural network is a *behaviour model* built through a process of *learning* from forecast example data of the system's behaviour. The ANN is progressively modified using the example data, to become a usable model that can predict the system's behaviour, expressed as relationships between the model's variables. During the learning process, the network evaluates relationships between the descriptive or explicative data (i.e. network inputs) and the outcome or explained data (i.e. network outputs). The result of the learning process is a *statistical model* that is able to provide estimates of the likely outcome. The predictive power of the ANN is assessed on test data excluded from the learning process.

Because ANNs need training data, experimental results or similar systems data must be available. These, however, are usually limited, as large amounts of data cannot easily be generated in the detail design phase of the engineering design process. To obtain the best possible ANN models, and to validate results, strategies that maximise learning with sparse data must be adopted. One such method is the 'leave-k-out' procedure for training (Lawrence 1991). A small number, *k*, of vectors out of the training vectors are held back each time for testing, and networks are trained, changing the *k* holdback vectors each time. Since the size of each network is usually modest for product design applications, and the number of training vectors small, training progresses rapidly, and creating multiple networks is not a burden. Another method for sparse training data is to insert 'noise' into the training set, creating multiple variant versions of each training vector.

5.3.3.1 Artificial Neural Network Modelling

Predictive *artificial neural network (ANN)* modelling can relate multiple quantitative and qualitative design parameters to system performance. These models enable engineering designers to iteratively and interactively test parameter changes and evaluate corresponding changes in system performance before a prototype is actually built and tested. This 'what-if' modelling ability both expedites and economises on the design process, and eventually results in improved design integrity. ANN models

can also supplement controlled experiments during systems testing to help ascertain optimum design specifications and tolerances. Artificial neural networks have been successfully applied to develop predictive networks for *system performance sensitivity* studies of the effects of alterations in design parameters. After translating as many parameters as possible into continuously valued numeric measures, so that alternate designs can be better compared, a 'leave-k-out' training procedure is used to develop predictive networks for performance on each of the quality tests, based on the design parameter specifications. A sensitivity model for each neural network is built by changing each design parameter in small increments across its range. Design engineers can thus use the models interactively to test the effects of design changes on system performance. In this way, designs can be optimised for performance, given manufacturing and cost constraints, before prototype models are built (Ben Brahim et al. 1992).

A further use of ANN models in engineering design is for the models to act as an *expert system*, where rules are *learned* directly through system instances, rather than defined through *knowledge engineering*. Artificial neural networks have also been successfully applied in engineering design by training a *multi-layered network* to act as an expert system in designing system mechanical components. The method uses documented design policies, heuristics, and design computation to construct a rule base (or decision table). The network is then trained on representative examples adhering to this rule base. This approach, which uses neural networks in lieu of expert systems, is advantageous in that rules are learned directly through design examples, rather than through tedious and often problematic knowledge acquisition (Zarefar et al. 1992).

A disadvantage of using neural networks in lieu of expert systems, though, is that explanation and tracing through the reasoning process are impossible; the neural networks then act essentially as a *black box*. The application of expert systems is considered later in greater detail.

The disadvantage, however, of using expert systems on their own is the time required for analysis and formatting, which are increased and not decreased. Experts systems are slow to develop and relatively expensive to update, as well as having fundamental, epistemological problems such as the appropriateness in representing knowledge in the form of decision rules or decision trees. The need to manually update expert systems each time expertise changes is cumbersome, while with artificial neural networks, all that is required is to launch a new learning process. The immense advantage of ANN models in lieu of expert systems is that analysis proceeds directly from factual data to the model without any manipulation of example data.

Artificial neural networks (ANN) can also be mathematically termed as *universal approximations* (according to Kolmogorov's theorem; Kolmogorov 1957), in that they have the ability to represent any function that is either linear or non-linear, simple or complicated. Furthermore, they have the ability to learn from representative examples, by error back propagation. However, artificial neural networks supply answers but not explanations. The ANN model embodies *intuitive associations* or correlations, not *causal relations* or explanations. ANN models are predictive (i.e.

close to reality) and associative (i.e. include typical profiles) but *not* descriptive. Examining the artificial neural network itself only shows meaningless numeric values. The ANN model is fundamentally a *black box*. On the other hand, being continuous and derivable, one can explore ANN models beyond simple statistical interrogation to determine typical profiles, explicative variables (network inputs), and apply example data to determine their associated probabilities. Artificial neural networks have the ability to account for any functional dependency by discovering (i.e. learning and then modelling) the nature of the dependency without needing to be prompted. The process goes straight from the data to the model without intermediary interpretation or problem simplification. There are no inherent conditions placed on the predicted variable, which can be a yes/no output, a continuous value, or one or more classes among *n*, etc. However, *artificial neural networks are insensitive to unreliability in the data*.

Artificial neural networks have been applied in engineering design in predictive modelling of system behaviour using *simulation* augmented with ANN model interpolation (Chryssolouris et al. 1989), as well as in interpolation of *Taguchi robust design* points so that a full factorial design can be simulated to search for optimal design parameter settings (Schmerr et al. 1991).

An artificial neural network is a set of elements (i.e. *neurodes* or, more commonly, *neurons*) linked to one another, and that transmit information to each other through connected links. Example data (*a* to *i*) are given as the inputs to the ANN model. Various values of the data are then transmitted through the connections, being modified during the process until, on arrival at the bottom of the network, they have become the predicted values—for example, the pair of risk probabilities P1 and P2 indicated in Fig. 5.53.

a) The Building Blocks of Artificial Neural Networks

Artificial neural networks are highly distributed interconnections of adaptive nonlinear *processing elements (PEs)*, as illustrated below in Fig. 5.54.

The connection strengths, also called the network *weights*, can be adapted so that the network's output matches a desired response. A more detailed view of a PE is shown in Fig. 5.55.

An artificial neural network is no more than an interconnection of PEs. The form of the interconnection provides one of the key variables for dividing neural networks into families. The most general case is the fully connected neural network. By definition, any PE can feed or receive activations of any other, including itself. Therefore, when the weights are represented in matrix form (the weight matrix), it will be fully populated. A (6×6) PE fully connected network is presented in Fig. 5.56.

This network is called a *recurrent network*. In recurrent networks, some of the connections may be absent but there are still feedback connections. An input presented to a recurrent network at time t will affect the networks output for future time steps greater than t. Therefore, recurrent networks need to be operated over time. If the interconnection matrix is restricted to feed-forward activations (no feedback

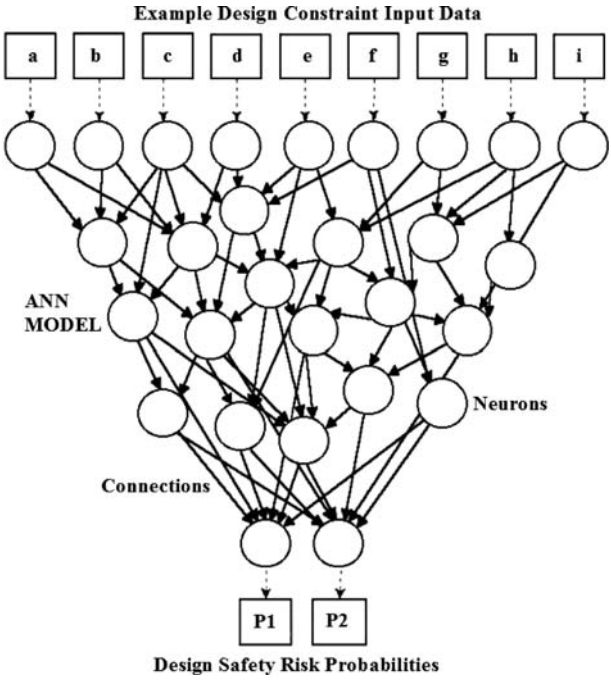


Fig. 5.53 Schematic layout of a complex artificial neural network (Valluru 1995)

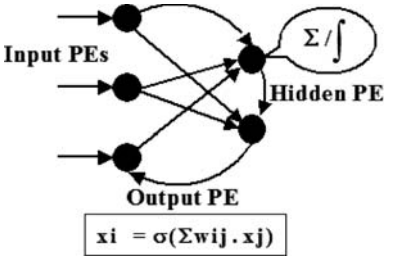


Fig. 5.54 The building blocks of artificial neural networks, where σ is the non-linearity, x_i the output of unit i, x_j the input to unit j, and w_{ij} are the weights that connect unit i to unit j

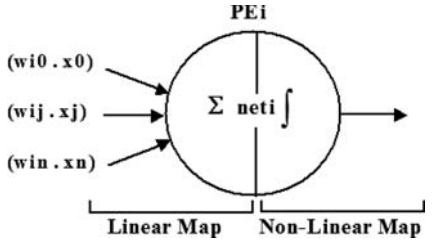


Fig. 5.55 Detailed view of a processing element (PE)

nor self connections), the ANN is defined as a *feed-forward network*. Feed-forward networks are *instantaneous mappers*, i.e. the output is valid immediately after the presentation of an input. A special class of feed-forward networks is the *layered*

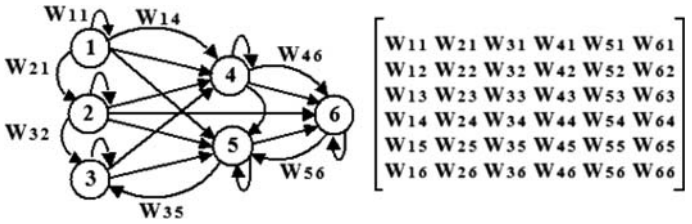
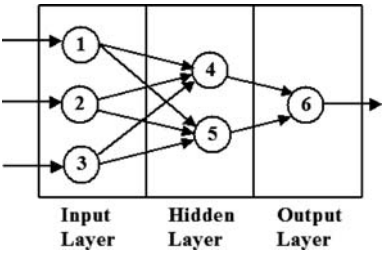


Fig. 5.56 A fully connected ANN, and its weight matrix

Fig. 5.57 Multi-layer perceptron structure



class, also termed a *multi-layer perceptron (MLP)*. This describes a network that consists of a single layer of non-linear PEs without feedback connections. Multi-layer perceptrons have PEs arranged in layers whereby the layers that receive input are called the input layers, layers in contact with the *outside world* are called output layers, and layers without direct access to the *outside world*, i.e. connected to the input or output, are called hidden layers (Valluru 1995).

The *weight matrix* of a multi-layer perceptron can be developed as follows (Figs. 5.57 and 5.58): from the example MLP in Fig. 5.57, the input layer contains PEs 1, 2 and 3, the hidden layer contains PEs 4 and 5, and the output layer contains PE 6.

Figure 5.58 shows the MLP's weight matrix. Most entries in the weight matrix of an MLP are zero. In particular, any feed-forward network has at least the main diagonal, and the elements below it populated with zeros. Feed-forward neural networks are therefore a special case of recurrent networks. Implementing partially connected topologies with the fully connected system and then zeroing weights is inefficient but is sometimes done, depending on the requirements for the artificial neural network. A case in point would be the weight matrix of the MLP below:

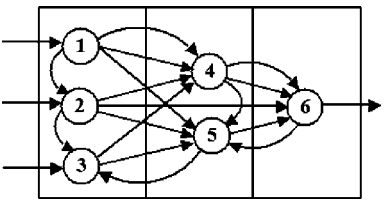


Fig. 5.58 Weight matrix structure for the multi-layer perception

$$\begin{bmatrix} 0 & 0 & 0 & W41 & W51 & 0 \\ 0 & 0 & 0 & W42 & W52 & 0 \\ 0 & 0 & 0 & W43 & W53 & 0 \\ 0 & 0 & 0 & 0 & 0 & W64 \\ 0 & 0 & 0 & 0 & 0 & W65 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

b) Structure of Artificial Neural Networks

A basic artificial neural network (ANN) structure thus consists of three layers: the input layer, the hidden layer, and the output layer, as indicated in Fig. 5.59 (Haykin 1999).

This MLP works in the following manner: for a given *input vector*

$$[(x_0) \setminus \text{vec}] = \{a_0, \ldots a_i\} \tag{5.104}$$

the following *output vector* is computed

$$[(o_0) \setminus \text{vec}] = \{c_0, \ldots c_i\} \tag{5.105}$$

The ANN implements the function f where

$$f([(x_0) \setminus \text{vec}]) = [(o_0) \setminus \text{vec}] \tag{5.106}$$

The basic processing element (PE) group of the MLP is termed the *artificial perceptron (AP)*. The AP has a set of input connections from PEs of another layer, as indicated in Fig. 5.60 (Haykin 1999).

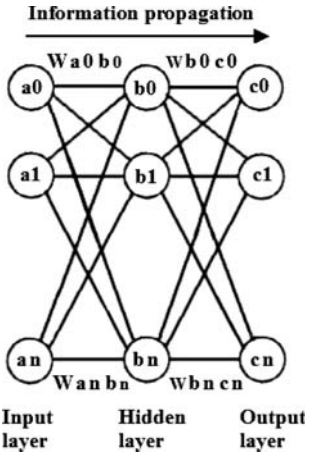


Fig. 5.59 Basic structure of an artificial neural network

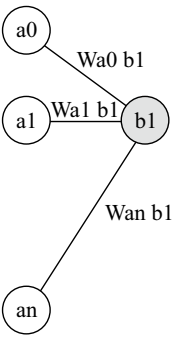


Fig. 5.60 Input connections of the artificial perceptron (a_n, b_1)

An AP computes its output in the following fashion: the output is usually a real number, and is the function of the activation, z_i, where

$$b_i = \sigma_i(Z_i) \tag{5.107}$$

The activation is computed as follows

$$z_i = \sigma_j w_j a_j \tag{5.108}$$
$$\delta = \text{the activation function}$$

There are many different activation functions (σ) in use. ANNs that work with binary vectors usually use the step-function:

$$\sigma(z) = 1$$
$$z \in [\theta, \infty) \quad \text{else } 0 \quad (\text{usually } \theta = 0)$$

These activation functions (σ) are called *threshold logic units (TLUs)*, as indicated in the binary step-function illustrated in Fig. 5.61.

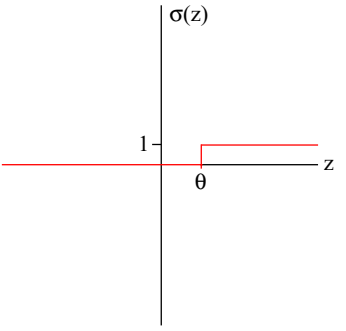


Fig. 5.61 The binary step-function threshold logic unit (TLU)

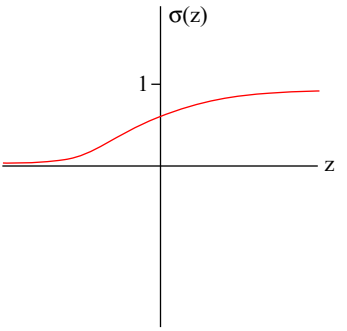


Fig. 5.62 The non-binary sigmoid-function threshold logic unit (TLU)

Graphic examples of threshold logic units (TLU) (Fausett 1994):

Non-binary ANNs often use the *sigmoid function* as activation function where the parameter ρ determines the shape of the sigmoid, as indicated in Fig. 5.62 and in Eq. 5.109

$$\sigma(z) = [1/(1 + e^{z/\rho})] \tag{5.109}$$

The most significant advantage of an MLP is that the artificial neural network is highly parallel. The MLP is also robust in the presence of noise (i.e. deviations in input) where a small amount of noise will not drastically affect the output. Furthermore, it can deal with unseen output, through generalisation from the learned input-output combinations. The threshold function ensures that the activation value will not go beyond certain values (generally, between 0 and 1) and prevents against catastrophic evolutions (loop effect where values become higher and higher).

c) Learning in Artificial Neural Networks

The basic operation of each AP is to multiply its input values by a *weight* (one per input), add these together, place the result into a threshold function, and then send the result to the *neurodes* downstream in the following layer. The learning mechanism of artificial neural networks is as follows: each set of example data is input to the ANN, then these values are propagated towards the output through the basic operation of each AP.

The prediction obtained at the ANN's output(s) is most probably erroneous, especially at the beginning. The error value is then computed as the difference between the expected value, and the actual output value. This error value is back-propagated by going upwards in the network and modifying the *weights* proportionally to each AP's contribution to the total error value. This mechanism is repeated for each set of example data in the learning set, while performance on the test set improves.

This learning mechanism is called *error back propagation*. The method is not unique to artificial neural nets, and is a general method (i.e. gradient method) applicable to other evolutionary computation objects.

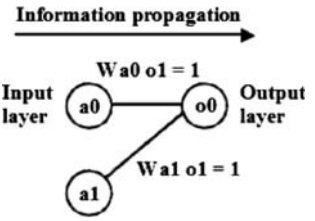


Fig. 5.63 Boolean-function input connections of the artificial perceptron (a_n, o_0)

Table 5.26 Boolean-function input values of the artificial perceptron (a_n, o_0)

a_0	a_1	z	o_0
0	0	0	0
0	1	1	0
1	0	1	0
1	1	2	1

For example, consider the input connections of the AP of an artificial neural network implementing the Boolean AND function ($\theta = 2$), as illustrated in Fig. 5.63 (Haykin 1999).

Consider all the possible values of the ANN implementing the Boolean AND function ($\theta = 2$) for a_0, a_1, z, and o_0.

The two-dimensional pattern space of the *AP* can now be developed according to the values given in Table 5.26. This is illustrated in Fig. 5.64. The TLU groups its input vectors into two classes, one for which the output is 0, the other for which the output is 1. The pattern space for an n input unit will be n-dimensional (Fausett 1994).

If the TLU uses threshold θ, then for the $[(x_0)\backslash\text{vec}]$ input vector, the output for the decision plane $\sum_{\forall i} w_i\, a_i \geq \theta$ will be 1, else 0. The equation for the decision plane is $\sum_{\forall i} w_i\, a_i = \theta$, which is a diagonal line, as illustrated in Fig. 5.64. Thus, in the case of the previous example:

$$w_0 a_0 + w_1 a_1 = \theta \Leftrightarrow a_1 = -(w_0/w_1) \cdot a_0 + (\theta/w_1)$$

Learning rules Several learning rules are used to train threshold logic units (TLUs), such as the *gradient descent technique* and the *delta learning rule*.

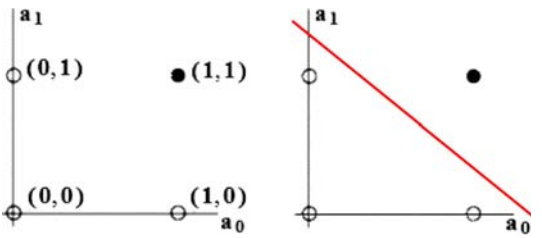


Fig. 5.64 Boolean-function pattern space and TLU of the artificial perceptron (a_n, o_0)

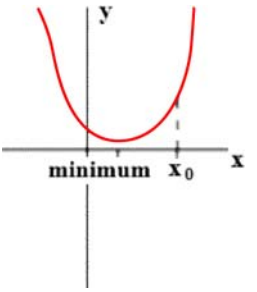


Fig. 5.65 The gradient descent technique

Suppose y is a function of x ($y = f(x)$), $f(x)$ is continuous, and the derivative $\mathrm{d}y/\mathrm{d}x$ can be found at any point. However, if no information is available on the shape of $f(x)$, local or global minimums cannot be found using classical methods of calculus. The slope of the tangent to the curve at x_0 is $[\mathrm{d}y/\mathrm{d}x]_{x_0}$.

For small values of Δx, Δy can be approximated using the expression

$$y_1 - y_0 = [\mathrm{d}y/\mathrm{d}x]_{x_0}(x_1 - x_0) \tag{5.110}$$

where:

$$\Delta y = y_1 - y_0$$
$$\Delta x = x_1 - x_0 \,.$$

Let:

$$\Delta x = \mathrm{d}y/\mathrm{d}x \cdot \alpha \Rightarrow \Delta y$$
$$= \alpha(\mathrm{d}y/\mathrm{d}x)^2$$

where:

α is a small parameter not to overshoot any minimums or maximums.

Starting from a given point (x_0) in Fig. 5.65, the local minima of the function $f(x)$ can be found by moving down the curve ($\Delta x = \mathrm{d}y/\mathrm{d}x \cdot \alpha$), until Δy becomes negative (at that point, the curve has already started moving away from the local minima). This technique is termed the *gradient descent*. The gradient descent technique is used to train TLUs.

d) Back Propagation in Artificial Neural Networks

Consider the ANN of Fig. 5.66. Assume the neurodes are the TLUs $\alpha(x) = 0$, $\forall x$ (Haykin 1999).

The *back-propagation (BP)* algorithm accounts for errors in the *output layer* using all the weights of the ANN. Thus, if a TLU in the output layer is off, it will change weights not only between the hidden and output layer but also between the input and hidden layer. The BP algorithm uses the *delta learning rule* expressed as $\Delta w_i = \alpha(t_j - z_j) \cdot a_{ji}(\Delta x = \mathrm{d}y/\mathrm{d}x\,\alpha)$.

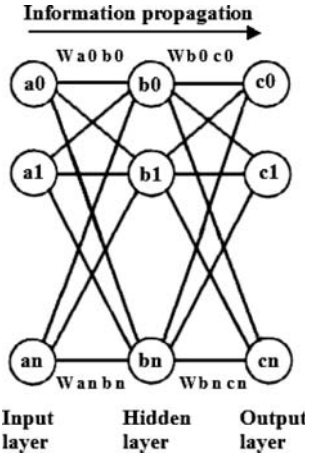


Fig. 5.66 Basic structure of an artificial neural network: back propagation

If the training set consists of the following pairs for the TLU:

$$\langle [(x_j)\backslash \text{vec}], t_j \rangle, \quad j = 0, \ldots n \quad \text{and} \quad [(x_j)\backslash \text{vec}] = \langle a_{j0}, \ldots a_{jm} \rangle$$

then the error for each pair is defined as

$$E_j = \tfrac{1}{2}(t_j - o_j)^2 \cdot j \tag{5.111}$$

The total error for the training set is

$$E = \sum\nolimits_{\forall j} E_j \tag{5.112}$$

where $E_j \; \forall j$ is a function of the weights connected to the TLU.

Thus, for all possible weight vectors, there exists an error measure (E) for a given training set. However, since the activation function is a step function, the error measure would not be a continuous function. The value o_j must be changed to z_j in the definition of the error E_p, which means that the activation level is used, rather than the produced output to compute the error. This yields a continuous function

$$E_j = \tfrac{1}{2}(t_j - z_j)^2 j \tag{5.113}$$

It can be shown that the slope of E_j with respect to the ith weight is: $-(t_j - z_j)\, a_{ji}$; the delta learning rule is thus expressed as

$$\Delta w_i = \alpha (t_j - z_j) a_{ji} (\Delta x = \mathrm{d}y/\mathrm{d}x\, \alpha) \tag{5.114}$$

when working with the jth training pair. Thus, for a training set defined as:

$$\langle [(x_j)], t_j \rangle, \; j = 0, \ldots, m, \; x_j = \langle x_{j0}, \ldots, x_{jn} \rangle, \text{ and } t_j = \langle t_{j0}, \ldots, t_{jn} \rangle$$

i) Compute the output of the hidden layer using x_j.
ii) Compute the output of the output layer using the output of the hidden layer $(b_0 \ldots b_n)$.
iii) Calculate the error for each output node. For the kth output node: $\delta_k = (t_{jk} - z_k)$, where z_k is the activation of the kth output node.
iv) Train the output nodes using the delta rule, assuming mth hidden node, kth output node is: $\Delta w_{bmck} = \alpha \delta_k b_m$.
v) Calculate the error for each hidden node. For the mth hidden node: $\delta_m = \sum_{k=1\ldots n} \delta_k w_{bmck}$ where δ_k is the computed error for the kth output node.
vi) Train hidden nodes using the delta rule (assuming the hth input node, lth hidden node): $\Delta w_{ahbm} = \alpha_m x_{jh}$.

These steps are repeated for each training vector, until the ANN produces acceptable outputs for the input vectors.

e) Fuzzy Neural Rule-Based Systems

The basic advantage of neural networks is that the designer does not have to program the system. Take, for example, a complex ANN of which the input is an $n \times n$ bitmap, which it recognises as the process equipment model (PEM) on the AIB blackboard (assuming the ANN is capable of distinguishing between a vessel, tank and container, then the input layer has n^2 nodes, and the output layer has three nodes, one for each PEM). In the ideal case, the designer does not have to write any code specific, and simply chooses an appropriate ANN model and trains it. The logic of each PEM is encoded in the weights and the activation functions.

However, artificial neural networks also have their drawbacks. They are fundamentally black boxes, whereby the designer does not know what part of a large designed network is responsible for a particular part of the computed output. Thus, the network cannot be modified to improve it.

ANN models are good at reaching decisions based on incomplete information (i.e. if the input vector does not match any of the training vectors, the network still computes a reasonable output in the sense that the output will probably be close to the output vector of a training vector that, in turn, is close to the input). Fuzzy rule-based systems are good at dealing with imprecise information. However, determining their membership functions is usually difficult. The fuzzy rule-based neural network basically makes up a membership function based on training vectors.

Consider for example, the fuzzy rules (Valluru 1995):

R_1: IF x is F_1 THEN z is H_1
R_2: IF x is F_2 THEN z is H_2
... and
R_n: IF x is F_n THEN z is H_n .

To teach this rule-base to an ANN, the training pairs are: $((F_1, H_1) \ldots (F_n, H_n))$.

The problem is that the fuzzy sets F_i and H_i are both defined by their membership functions μ, with domain R, the set of real numbers, the input vectors of the training set having infinite elements.

Obviously, it is impossible to have infinitely large neural networks, so the membership functions are transformed so that they are discrete (by taking samples at equal intervals). Furthermore, the range of the membership functions are contained to the interval [0, 1]. If the range is $[-\infty,+\infty]$, the transform T is then $D_{[-\infty,+\infty]} \rightarrow D_{[0,1]}$.

This is termed a *loss-less transformation*. To graphically present this transformation, as illustrated in Fig. 5.67, draw a semicircle in the region defined by $0 < x < 1, 0 < y < 0.5$, with the centre (0.5, 0.5), and draw lines to all points on the x-axis. $T(x_0)$ is the x coordinate of the intersection of the line crossing the x-axis at x_0 with the semicircle.

With k samples of the membership function at x_i and $i = 0 \ldots k, x_i = i/k$, i, the training set of the fuzzy neural network is:

$$\{(\mu_{F_i}(x_0), \mu_{F_i}(x_1) \ldots \mu_{F_i}(x_k)), (\mu_{H_i}(x_0), \mu_{H_i}(x_i) \ldots \mu_{H_i}(x_k)) | i = 0 \ldots n\}$$

The training set consists of pairs of sampled membership functions. The pairs correspond to the rules of the fuzzy rule-based neural network considered. As indicated previously, the advantage of fuzzy rule-based neural networks is the fact that the designer does not have to program the system, and the fuzzy neural network makes the membership functions. With the example above, the membership functions were already known. In actual use of fuzzy ANN models, the membership functions would be extracted from the training pairs using the ANN.

Fuzzy artificial perceptrons (FAP) Fuzzy *T-norm functions* have the following properties:

$$\begin{aligned} &T : [0,1]x[0,1] \rightarrow [0,1], T(x,y) = T(y,x), T(0,x) = 0, \\ &T(1,x) = x, T(T(x,y),z) = T(x,T(y,z)), \\ &x \leq a \cap y \leq b \rightarrow T(x,y) \leq T(a,b) \end{aligned}$$

From the definition of *intersection* of fuzzy sets, the notation

$$\mu_{F \cap G}(x,y) = \min(\mu_F(x), \mu_G(y)) \text{ is a } T\text{-norm, where } x = y\,.$$

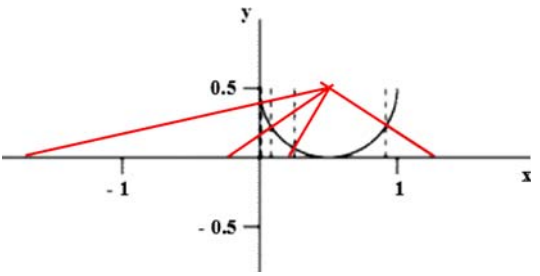


Fig. 5.67 Graph of membership function transformation of a fuzzy ANN

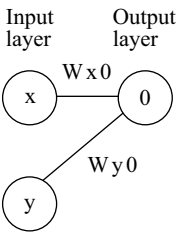


Fig. 5.68 A fuzzy artificial perceptron (AP)

Fuzzy *T-conorm functions* have the following properties:

$$T : [0,1]x[0,1] \rightarrow [0,1], T(x,y) = T(y,x), T(0,x) = x,$$
$$T(1,x) = 1, T(T(x,y),z) = T(x,T(y,z)),$$
$$x \leq a \cap y \leq b \rightarrow T(x,y) \leq T(a,b)$$

From the definition of *union* of fuzzy sets, the notation

$$\mu_{F \cup G}(x,y) = \max(\mu_F(x), \mu_G(y)) \text{ is a } \textit{T-conorm}, \text{ where } x = y\ .$$

A fuzzy *artificial perceptron (AP)* can now be defined; these are really ANNs with two input neurodes (x and y), no hidden layer, and an output neurode o (Fig. 5.68). The weights are w_{xo} and w_{yo}.

Fuzzy AND AP: x, y, o, w_{xo}, $w_{yo} \in [0,1]$. Where t is a T-norm function, s is a T-conorm function: $o = t(s(x,w_{xo}), s(y,w_{yo}))$.

Fuzzy OR AP: x, y, o, w_{xo}, $w_{yo} \in [0,1]$. Where t is a T-norm function, s is a T-conorm function: $o = s(t(x,w_{xo}), t(y,w_{yo}))$.

f) Artificial Neural Networks in Engineering Design

As indicated previously, an ANN is a computer model consisting of many simple processing elements (PEs) in layered structures. The PEs interact through weighted connections that, when manipulated, enable an ANN to recognise patterns from sample data of system (or assembly/component) performance based on specific input variables. Neural networks can also be used to *predict* input variables for conditions that have not been determined experimentally.

Figure 5.69 is an example of an ANN-generated, three-dimensional plot of predicted wear rate for a mechanical device, as a function of piston sliding distance and sliding speed. The figure depicts wear rate values obtained for different distances and speeds (Fusaro 1998).

Critical parameters such as load, speed, sliding distance, friction coefficient, wear, and material properties are used to produce models for each set of sample data.

The study shows that artificial neural networks are able to model such simple systems, illustrating the feasibility of using ANN models to perform *accelerated life testing* on more complicated prototype mechanical systems. The following graph

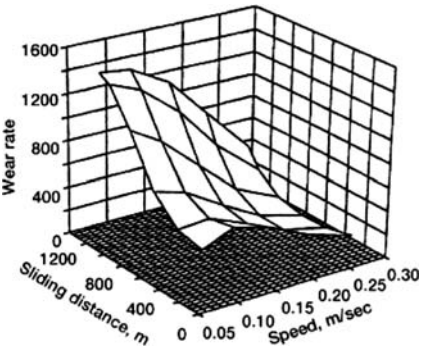


Fig. 5.69 Three-dimensional plots generated from a neural network model illustrating the relationship between speed, load, and wear rate (Fusaro 1998)

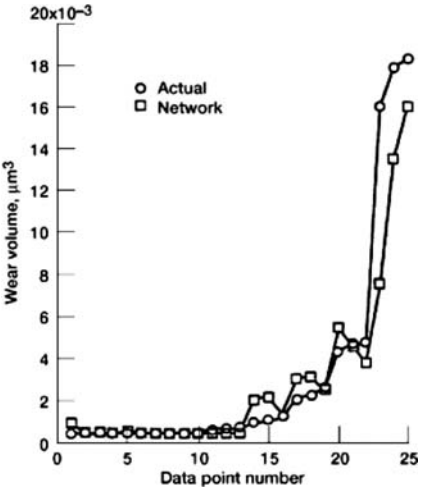


Fig. 5.70 Comparison of actual data to those of an ANN model approximation (Fusaro 1998)

(Fig. 5.70) compares actual wear data to those generated from an ANN model. As the graph illustrates, the correlation is very good (Fusaro 1998).

ANNs are normally classified by learning procedure, the most common being unsupervised and supervised learning. In *unsupervised learning*, the network is trained by internal characterisation of data patterns, with no other information or teaching requirement. This type of ANN is appropriate to *preliminary* engineering design applications, as it can analyse the possible *occurrence* of a process failure condition but not necessarily the type of failure characteristics or extent of the fault.

In *supervised learning*, individual values of the weighted connections between neurodes are adjusted during training iterations to produce a desired output for a given input. Knowledge is thus represented by the structure of the network and the values of the weights. This type of ANN is appropriate to *detail* design applications supported by sample data. This procedure offers several advantages in the field of pattern recognition and analysis of sample failure data, including an ability to learn from examples, and the ability to generalise. The generalisation property results in a network trained only on *representative* input sample data, to be able to

provide relatively accurate results without being trained on all possible input data. Thus, the primary advantage of ANN models over operational modelling and expert system approaches is that representative sample data can be used to train the ANN with no prior knowledge of system operation (Farell et al. 1994).

ANN models typically exhibit the rule-based expert system characteristics of *knowledge-based expert systems* without the need for prior representation of the rules. However, it is the ability to generalise and form accurate evaluations from design specification data not present in the sample data training set that is the key requirement of the ANN.

Example of ANN in engineering design—preparation of training data The majority of designs based on process engineering analysis rely on operational models or simulated processes. While providing guidelines for design implementation, they do not highlight inherent problems regarding information quality and availability. For this reason, engineering design data depend on practical process information, such as sensitivity of parameters to fault conditions and, of course, expert process design knowledge. As an example of the application of ANN models in engineering design, a feed-forward ANN topology, using the back-propagation learning algorithm for training, is investigated for pump fault prediction (Lippmann 1987).

This ANN topology incorporates a *supervised training* technique and, thus, it is necessary to define training data prior to the ANN analysis. Process measurements relating to potential fault conditions and normal operation, including information on types of failure, are necessary for ANN learning. This information can, however, be difficult to obtain in practical situations. Knowledge for ANN training is established from models or experience.

Engineering processes and systems are often complex and difficult to incorporate precise descriptions of normal and faulty operating conditions into models. Data founded on experience can be based on quantitative measurements or even qualitative information derived from previous measurements. The quantitative approach, involving data corresponding to historically experienced failures in similar systems and equipment, produces a more accurate evaluation of the design specifications but is dependent on data quality. In real-world situations, the quality of historical condition data and records relating to failure conditions of complex systems is more often questionable. Furthermore, it is unlikely that every potential failure would be experienced historically; consequently, qualitative data are often incorporated to expand quantitative data in the design knowledge base, or even used on their own if no quantitative data are available. However, in situations such as critical pump failure analysis, where problems can be manifested in various forms depending on the design type and size, qualitative data are not considered precise.

A database of historical pump problems and typical failure data of similar pumps enabled an initial approach to pump failure prediction based on quantitative data. The *cumulative sum charting method* is applied to assign specific parameter measurements to pump operating conditions for ANN training purposes. The *cusum chart* is constructed from an initial choice of target values. The difference between each measurement and the target is added to a cumulative sum. This value is plotted

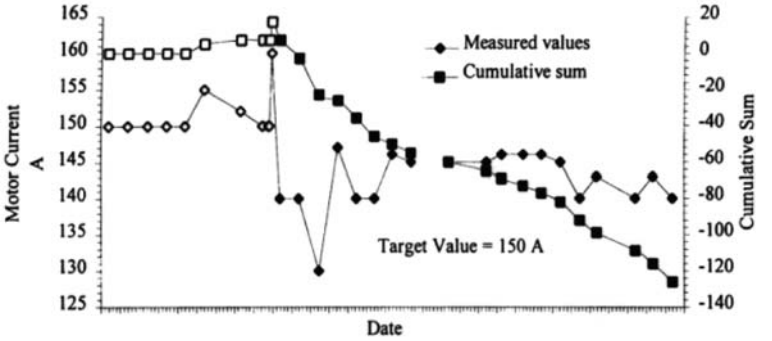


Fig. 5.71 Example failure data using cusum analysis (Ilott et al. 1997)

to provide a simple yet effective way to determine minor deviations in parameter levels. A knowledge base is established from parameters commonly available for typical fault conditions of similar pumps, as the ANN requires consistent parameter input to distinguish between different operating conditions. The parameters used in the example are motor current and delivery pressure (Ilott et al. 1995).

From motor current data prior to failure, a target value is chosen for calculation of the cumulative sum, such as 150 A. Initial observation of the sample data highlights the difficulty in identifying fault data. For example, the motor current data relating to a specific fault may be consistently higher during the initial stages of operation, due to a primary bearing problem. On further examination of the sample data, there is evidence of a marked deviation in motor current values that coincide with a decrease in delivery pressure. The cusum chart clearly indicates a deviation in motor current operating level from positive to negative during the sample data period, indicating the motor current to be consistently below target value.

This procedure is repeated for all historical pump failures to establish a usable knowledge base of pump failure data. Figure 5.71 shows the motor current data prior to failure, including both sample data and cumulative sum values.

ANN model experimental procedure A feed-forward ANN is trained using the back-propagation learning algorithm to predict pump operating conditions from features provided by the knowledge base of motor current and delivery pressure values. The knowledge base established from the cusum analysis is split into training data and test datasets for ANN implementation. These datasets typically include a series of data patterns, each incorporating one motor current and one delivery pressure parameter value, relating to specific fault conditions as well as normal pump operation. The data patterns are input to the ANN every training iteration. Once trained to a preset number of iterations or error levels, the ANN is tested with data not presented in the training dataset to verify generalisation capability. The quantity and quality of data available for ANN training purposes is an important issue and dictates the confidence in results from the ANN model. Sufficient data would provide good representation of the decision space relating to specific fault conditions and

normal pump operation. The exact quantity of data required cannot be specified but insufficient data cause poor generalisation ability.

In designing non-complex pumping systems where adequate models can be developed, the knowledge base can simply be manufactured. The ANN model is trained using the back-propagation learning algorithm where the sum squared error (SSE) between the desired and actual ANN output is used to amend weighted connections between the PEs to reduce the SSE during further training. For complex system designs, many amendments are required due to re-investigation and system alterations. Representation capability of an ANN is determined by the size of the input space.

The example ANN structure consists of three layers, and its topology consists of two sets of input neurodes (values of delivery pressure and motor current scaled between 0 and 1), several hidden neurodes, and five output neurodes (for fault conditions and normal operation). The ANN topology is illustrated in Fig. 5.72 (Ilott et al. 1997).

The example involves training the ANN model to a predefined error level, to investigate the effect on generalisation ability. The learning rule performs weight adjustment in order to minimise the SSE. Furthermore, a learning coefficient is used to improve ANN learning. The learning coefficient governs the size of the weight change with every iteration and subsequently the rate of decrease of the SSE value, and is adjusted dynamically so as to speed up network training. Convergence speed refers to the number of iterations necessary for suitable training of the ANN.

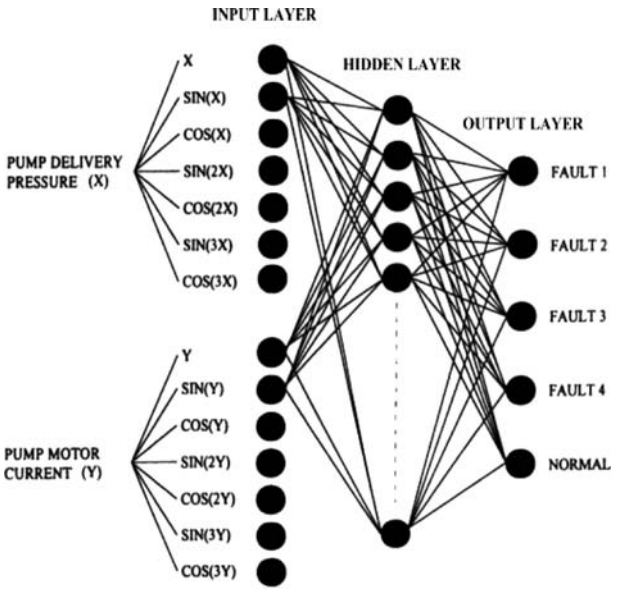


Fig. 5.72 Topology of the example ANN (Ilott et al. 1997)

Fuzzy ANN modelling Fuzzy ANN modelling is based on fuzzy pre-processing of input data. The purpose of such fuzzy pre-processing is to observe the effect of data representation on ANN performance with respect to the sensitivity of the pump parameters to identification of pump failure conditions. This methodology considers the definition of qualitative membership functions for each input parameter, and is considered an alternative method to increase ANN representation capability through compression of training data. Using the pump example, a motor current of 140 A would have membership of 0.5 to membership function 2 (MF2), a lower degree of membership to MF3 (0.06) and no membership to MF1. This procedure is repeated for delivery pressure and a value of each parameter MF is input to the ANN. An example of the fuzzy membership functions for motor current and delivery pressure parameters is given in Fig. 5.73a,b.

Example results The example results focus on the importance of data quality and, consequently, pre-processing with respect to ANN convergence speed and generalisation ability. The ANN topology is trained to investigate the effect of data quality

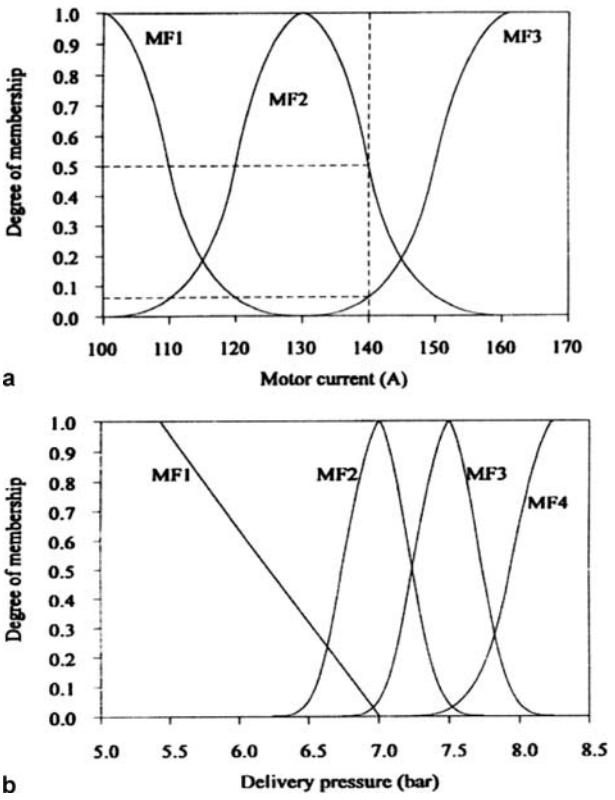


Fig. 5.73 a) An example fuzzy membership functions for pump motor current (Ilott et al. 1995), **b)** example fuzzy membership functions for pump pressure (Ilott et al. 1995)

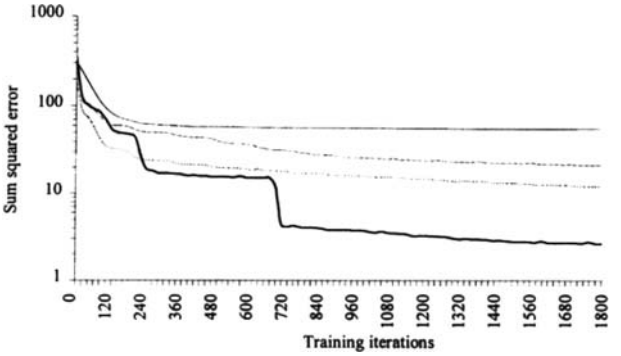


Fig. 5.74 Convergence rate of ANN iterations

on ANN performance. The SSE value is used to gauge the accuracy of training. The ANN converges faster with each iteration of refined test training data, as indicated in Fig. 5.74. After ANN training, generalisation ability is investigated using the original test set patterns. The quality of training data has a considerable effect on generalisation ability, which varies with the type of failure, and is lower for failure classes defined by fewer measurements in the training dataset. The example focused on maximising a design knowledge base despite the inherent limitations of real sample data.

The *cusum charting procedure* is a valuable tool in the development of the ANN knowledge base, through identification of parameter deviations in the sample data. The quality of training data as well as pre-processing both influence ANN convergence rate and ANN generalisation ability. Generalisation is one of the primary goals in training neural networks, to ensure that the ANN performs well on data that it has not been trained on. The standard method of ensuring good generalisation is to divide the training data into multiple datasets.

The most common datasets are the training, cross validation, and testing datasets. Refinement of the original training data improves ANN generalisation ability. The fuzzy pre-processing methodology results in a better improvement to ANN generalisation ability but is slow to converge during learning. The fuzzy pre-processing technique converges much faster during the learning phase, and produces generalisation ability comparative to that of the fuzzy approach.

Conclusion Accurate ANN analysis of pump failure conditions, based on a limited supply of historical data, is feasible for engineering design application during the detail design phase. However, the use of ANN models for engineering design, particularly in designing for safety, is dependent upon the availability of historical data and the sensitivity of parameter values in distinguishing between failure conditions. ANN analysis capability is also very much dependent upon methods of knowledge base generation, and the availability of design knowledge expertise (Ilott et al. 1995).

g) ANN Computational Architectures

Neural networks can be very powerful learning systems. However, it is very important to match the neural architecture to the problem. Several learning architectures are available with neural network software packages. These architectures are categorised into two groups: supervised and unsupervised. Supervised architectures are used to classify patterns or make predictions. Unsupervised neural networks are used to classify training patterns into a specified number of categories.

Supervised learning paradigms (back-propagation, probabilistic, and general regression) are composed of at least three layers: input, hidden and output. In each graphical representation, the input layer is on the left and the output layer on the right. Hidden layers are represented between the input and output layer. The input layer contains variables used by the network to make predictions and classifications. Analysis of data patterns or learning takes place in the hidden layer. The output layer contains the values the neural network is predicting or classifying. Information in the input layer is weighted as it passed to the hidden layer.

The hidden layer weight values are received from the input layer and produces outputs. Historical information is continuously analysed by the system through back propagation of error, where error is passed backwards until it is reduced to acceptable levels. Learning takes place when the neural network compares itself to correct answers and makes adjustments to the weights in the direction of the correct answers. Variations of supervised learning paradigms include differences in the number of hidden neurodes and/or weight connections.

The unsupervised network is composed of only two layers: input and output. The input layer is represented on the left and the output layer on the right. Information fed into the input layer is weighted and passed to the output layer. Learning takes place when adjustments are made to the weights in the direction of a succeeding neurode. In the illustrations below, each artificial neural network architecture is represented by a graphic containing rectangles and lines. Rectangles depict layers and lines depict weights.

Several types of supervised neural networks and one unsupervised neural network are illustrated collectively in Figs. 5.75 through to 5.81 (Schocken 1994).

ANN model architecture: supervised neural networks (I=input layer, H=hidden layer, O=output layer)

Standard back propagation —each layer is connected to the immediately previous layer (with either one, two or three hidden layers). Standard back-propagation neural networks are known to generalise well on a wide variety of problems (Fig. 5.75).
Jump connection back propagation—each layer is connected to every previous layer (with either one, two or three hidden layers). Jump connection back-propagation networks are known to work with very complex patterns, such as patterns not easily noticeable (Fig. 5.76).
Recurrent back-propagation networks with dampened feedback—each architecture contains two input layers, one hidden layer, and one output layer (Fig. 5.77).

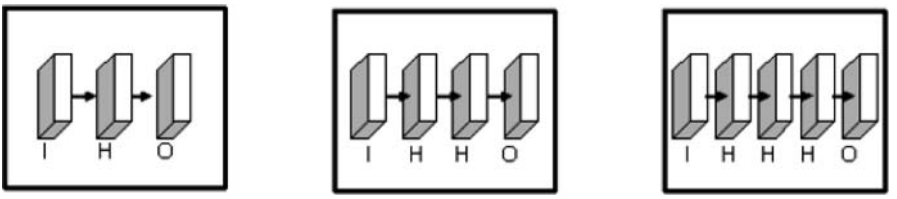


Fig. 5.75 Standard back-propagation ANN architecture (Schocken 1994)

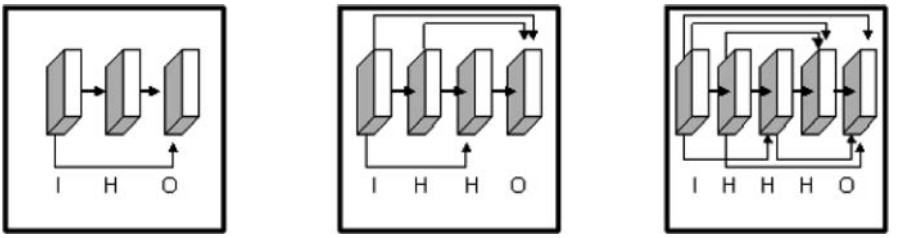


Fig. 5.76 Jump connection back-propagation ANN architecture (Schocken 1994)

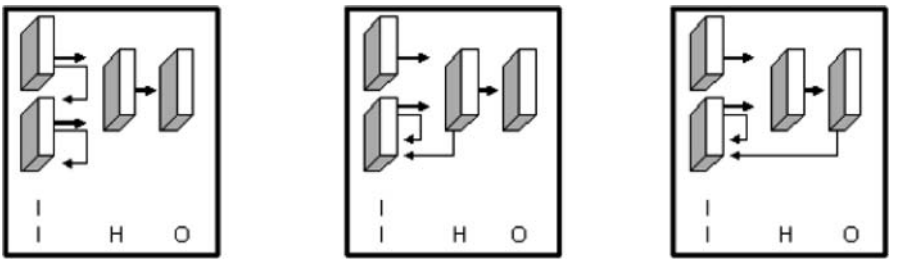


Fig. 5.77 Recurrent back-propagation with dampened feedback ANN architecture (Schocken 1994)

The extra input layer retains previous training experiences, much like memory. Weight connections are modified from the input, hidden or output layers, back into the network for inclusion with the next pattern. Recurrent back-propagation networks with dampened feedback networks are known to learn sequences and time series data.

Ward back propagation—each architecture contains an input layer, two or three hidden layers, and an output layer. Different activation functions (method of output) can be applied. Ward networks are known to detect different features in the low, middle and high dataset ranges (Fig. 5.78).

Probabilistic (PNN)—each layer is connected together. The hidden layer contains one neurode per data array. The output layer contains one neurode for each possible category. PNNs separate data into a specified number of output categories and train quickly on sparse data (Fig. 5.79).

General regression (GRNN)—each layer is connected together. Hidden and output layers are the same as PNN. Rather than categorising data like PNN, however,

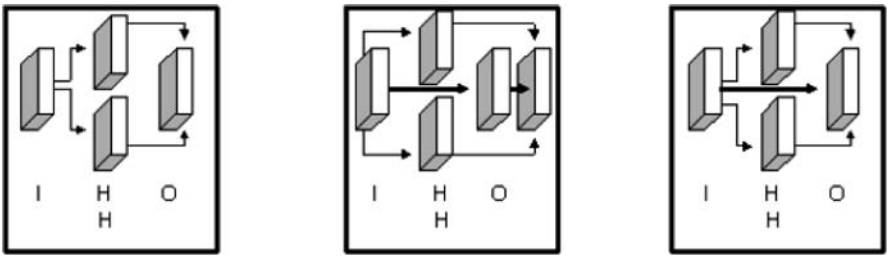


Fig. 5.78 Ward back propagation ANN architecture (Schocken 1994)

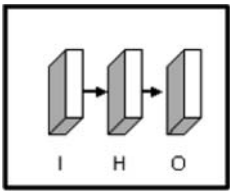


Fig. 5.79 Probabilistic (PNN) ANN architecture (Schocken 1994)

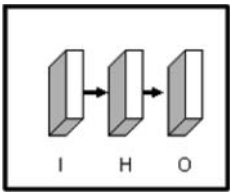


Fig. 5.80 General regression (GRNN) ANN architecture (Schocken 1994)

GRNN applications are able to produce continuous valued outputs and respond better than back propagation in many cases (Fig. 5.80).

Unsupervised neural network *Kohonen self-organising map*—contains an input and an output layer. One neurode is present in the output layer for each category specified by the user. Kohonen networks are known to separate data into a specified number of categories (Fig. 5.81).

In Sect. 5.4, an artificial intelligence-based *blackboard model* is used to hold shared information in a general and simple model that allows for the representation of a variety of modelled system behaviours. The AIB blackboard system is prescribed for problem-solving in knowledge-intensive domains that require large

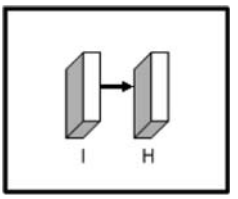


Fig. 5.81 Kohonen self-organising map ANN architecture (Schocken 1994)

amounts of diverse and incomplete knowledge, therefore necessitating multiple co-operation of various knowledge sources.

One knowledge source, a neural expert program (Lefebvre et al. 2003), is embedded in the AIB blackboard for processing of time-varying information, such as non-linear dynamic modelling, time series prediction, and adaptive control of various engineering design problems.

5.4 Application Modelling of Safety and Risk in Engineering Design

Returning to Sect. 1.1, the five main objectives that need to be accomplished in pursuit of the goal of the research in this handbook are:

- the development of appropriate theory on the integrity of engineering design for use in mathematical and computer models;
- determination of the validity of the developed theory by evaluating several case studies of engineering designs that have been recently constructed, that are in the process of being constructed or that have yet to be constructed;
- application of mathematical and computer modelling in engineering design verification;
- determination of the feasibility of a practical application of intelligent computer automated methodology in engineering design reviews through the development of the appropriate industrial, simulation and mathematical models.

The following models have been developed, each for a specific purpose and with specific expected results, in partly achieving these objectives:

- *RAMS analysis model*, to validate the developed theory on the determination of the integrity of engineering design.
- *Process equipment models (PEMs)*, for application in dynamic systems simulation modelling to initially determine mass-flow balances for preliminary engineering designs of large integrated process systems, and to evaluate and verify process design integrity of complex integrations of systems.
- *Artificial intelligence-based (AIB) model*, in which relatively new *artificial intelligence (AI)* modelling techniques, such as inclusion of *knowledge-based expert systems* within a *blackboard model*, have been applied in the development of intelligent computer automated methodology for determining the integrity of engineering design.

The third model, the *artificial intelligence-based (AIB) model*, will now be considered in detail in this section.

5.4.1 *Artificial Intelligence-Based (AIB) Blackboard Model*

Artificial intelligence (AI) has been applied to a number of fields of engineering design. Although there are some features that the various design areas share, such as the need to integrate heuristics with algorithmic numerical procedures, there are also some important differences. Each field of engineering seems to recognise the importance of representing declarative concepts, although specific needs vary. In process engineering, for example, the hierarchical representation of components with their functional relationships seems to be vital. In mechanical engineering, the representation of solid geometric shapes has been thoroughly studied and is viewed as being crucial to the successful evolution of computer aided design or manufacturing CAD/CAM systems. Artificial intelligence in engineering design can be described as a discipline that provides a multi-level methodology for knowledge-based problem-solving systems, in which a *knowledge-level specification* of the system (and the class of problems it must solve) is mapped into an *algorithm-level description* of an efficient search algorithm for efficiently solving that class of problems.

The algorithm description is then mapped into program code at the *program level*, using one or more programming paradigms (e.g. procedural programming, rule-based programming or object-oriented programming, OOP), or *shells* (e.g. RAM-ESP), or commercially available sub-systems (e.g. CLIPS, JESS or EXSYS).

The application of AI to engineering design thus represents a specialisation of software engineering methodology to:

- Design tasks
  (specified at the '*knowledge level*').
- Design process models
  (described at the '*algorithm level*').
- Design programs built from *shells*
  (implemented at the '*program level*').

Integration of the design process with blackboard models The quality of engineering design using traditional CAD techniques is adversely affected by two features of the design process.

Features of the design process affecting the quality of engineering design are:

- Limited scope in addressing problems that arise in the many stages of the development of an engineered installation.
- A lack of understanding of the essential processes involved in engineering design.

Both of these are related to systems integration issues. The life cycle of an engineered installation can be described by a collection of *projects*, where each project involves a coherent set of attributes, such as the design, manufacturing or assembling of a system. Traditional CAD tools typically address some narrow aspect of the design project, and fail to provide integrated support for the development of an

engineered installation, particularly evaluation of design integrity. Essentially, modern engineering design of complex systems requires an approach that allows multiple, diverse program modules, termed knowledge sources, to cooperate in solving complex design problems.

The (AIB) blackboard model The *artificial intelligence-based (AIB) blackboard model* that has been developed enables the integration of multiple, diverse program modules into a single problem-solving environment for determining the integrity of engineering design. This AIB blackboard model is a database that is used to hold shared information in a centralised model that allows for the representation of a variety of modelled system behaviours. Given the nature of programming for blackboard systems, it is prescribed for problem-solving in knowledge intensive domains that require large amounts of diverse and incomplete knowledge, therefore requiring multiple cooperation of various knowledge sources in the search of a large problem space.

The AIB blackboard model consists of a data structure (the blackboard) containing information (the context) that permits a set of modules (knowledge sources) to interact. The blackboard can be seen as a global database or working memory in which distinct representations of knowledge and intermediate results are integrated uniformly. It can also be seen as a means of communication among knowledge sources, mediating all of their interactions in a common display, review and performance evaluation area. The engineering design methodology for the AIB blackboard model, presented in the following graphical presentation (Fig. 5.82), applies the concept of object-oriented programming.

Object-oriented programming (OOP) has two fundamental properties, *encapsulation* and *inheritance*. Encapsulation means that the user (the engineering designer) can request an action from an object, and the object chooses the correct operator, as opposed to traditional programming where the user applies operators to operands and must assure that the two are type compatible. The second property, namely inheritance, greatly improves the re-usability of code, as opposed to traditional programming where new functionality often means extensive re-coding.

In this way, the AIB blackboard model may be structured so as to represent different levels of abstraction and also distinct and possibly overlapping solutions in the design space of complex engineering design problems. In terms of the type of problems that it can solve, there is only one major assumption—that the problem-solving activity generates a set of intermediate results.

The AIB blackboard model for engineering design integrity consists of four sections, each section containing six design modules, culminating in a summary design analysis module particular to each specific section (Fig. 5.83). The first section of the AIB blackboard model contains modules or knowledge sources for assessing preliminary design (inclusive of conceptual design basics), such as process definition, performance assessment, RAM assessment, design assessment, HazOp analysis, and critical process specifications, including a summary process analysis module. The second section contains modules for evaluating detail design, such as systems definition, functions analysis, FMEA, risk evaluation, criticality analysis,

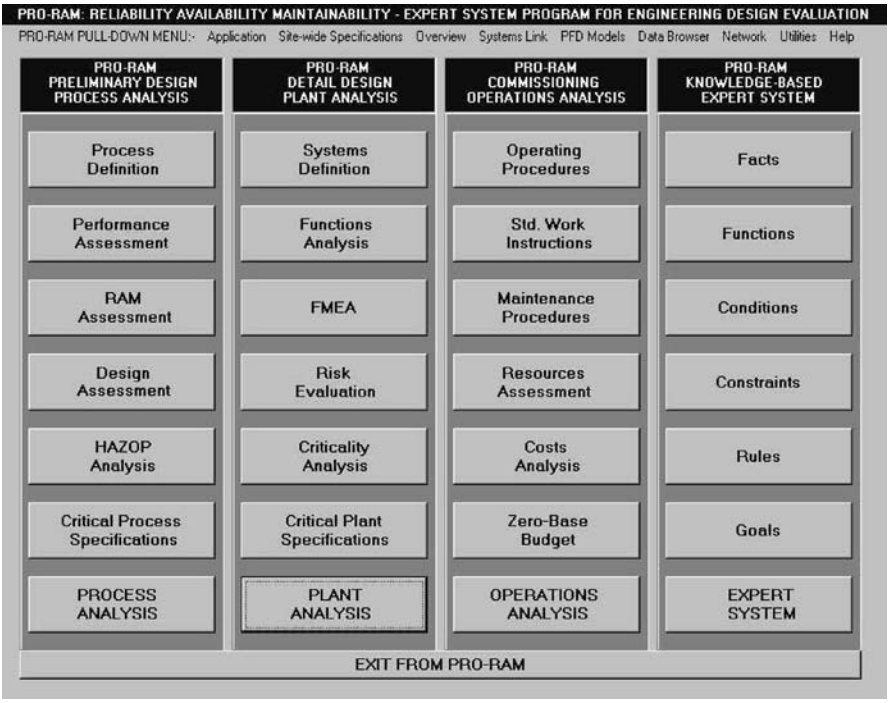


Fig. 5.82 AIB blackboard model for engineering design integrity (ICS 2003)

and critical plant specifications, including a summary plant analysis module. The third section contains modules related to operations analysis, and the fourth section contains modules of knowledge-based expert systems relating to the modules of the three former sections. Thus, the expert system module called 'facts' relates to process definition, systems definition and operating procedures, etc.

Most engineering designs are still carried out manually with input variables based on *expert judgement*, prompting considerable incentive to develop model-based techniques. Investigation of safety-related issues in engineering designs can effectively be done with discrete event models. A process plant's physical behaviour can be modelled by state transition systems, where the degree of abstraction is adapted both to the amount of information that is available at a certain design phase, and to the objective of the analysis. A qualitative plant description for designing for safety is sufficient in the early design phases, as indicated in Figs. 5.83 to 5.87. However, the verification of supervisory controllers in later design phases requires finer modelling such as the development of timed discrete models. The procedure of model refinement and verification is later illustrated by the application of *expert systems*.

A systematic hierarchical representation of equipment, logically grouped into systems, sub-systems, assemblies, sub-assemblies and components in a *systems breakdown structure (SBS)*, is illustrated in Fig. 5.84.

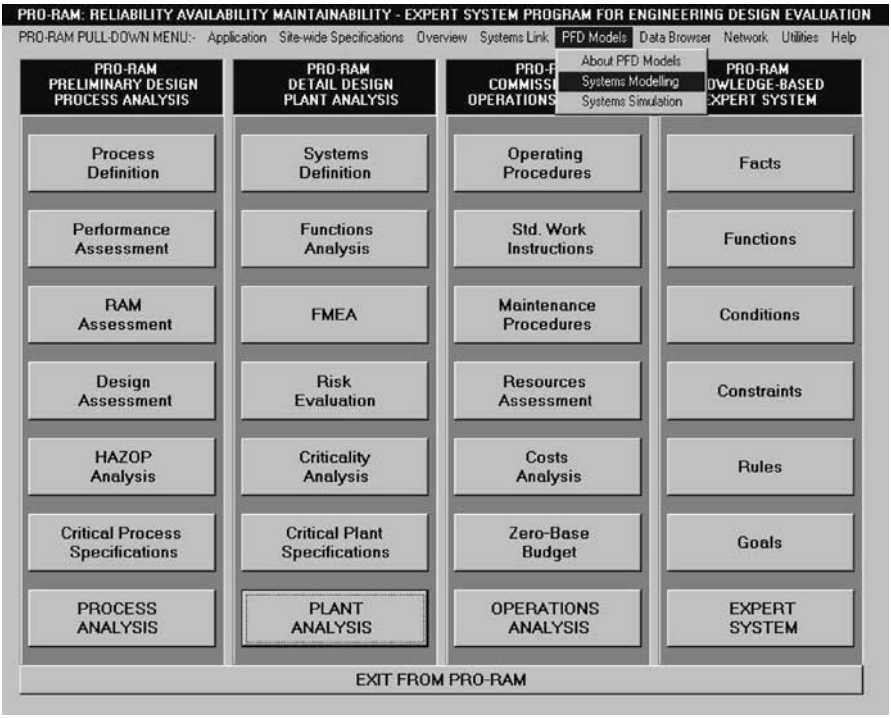


Fig. 5.83 AIB blackboard model with systems modelling option

The *systems breakdown structure (SBS)* provides visibility of process systems and their constituent assemblies and components, and allows for safety and risk analysis to be summarised from system level to sub-system, assembly, sub-assembly and component levels. The various levels of the *SBS* are normally determined by a framework of criteria established to logically group similar components into sub-assemblies or assemblies, which are then logically grouped into sub-systems or systems. This logical grouping of items at each level of an *SBS* is done by identifying the actual physical design configuration of the various items at one level of the *SBS* into items of a higher level of the systems hierarchy, and by defining common operational and physical functions of the items at each level. When designing or analysing a system for safety, a method is needed to determine how the variables are interrelated. System hierarchical models based on a structured *SBS*, as illustrated in Fig. 5.85, provide formulations of the core concept of a system in order to match the particular modelling perspective—for example, establishing FMEA and criticality analysis in designing for safety.

The particular model formalisms that are used depend on the objectives of the modelling requirements and the modelling techniques applied. In the case of *schematic design modelling*, the formalisms commonly used are *functional* (what a system can do), *behavioural* (describes or predicts the system's dynamic response)

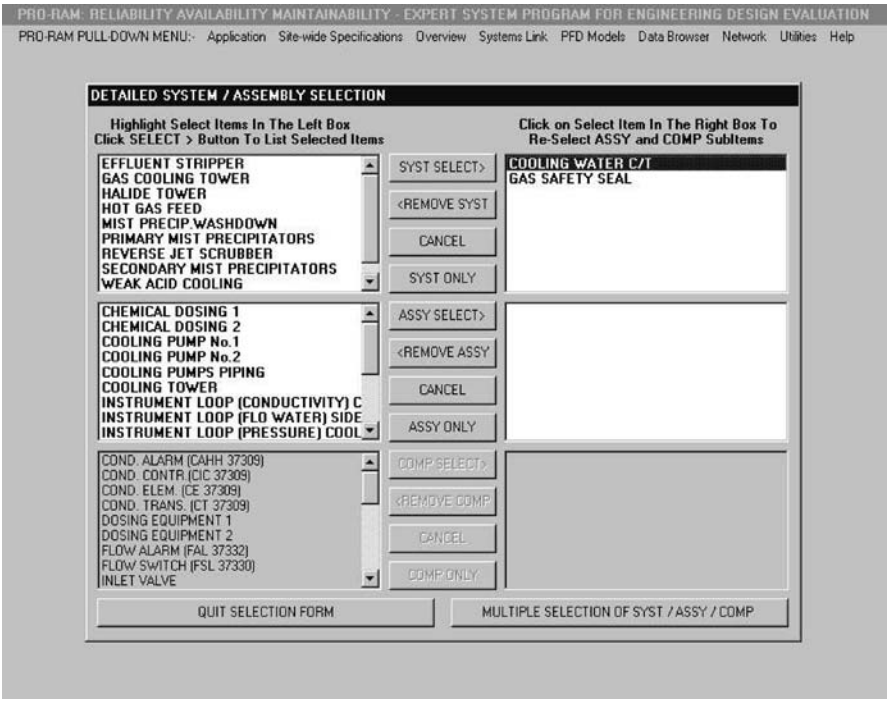


Fig. 5.84 Designing for safety using systems modelling: system and assembly selection

and *schematic* (an iconic model of the system's structure and connectivity). Thus, a schematic design model contains design variables and constraints describing the structural and geometric feature of the design. A detail design model typically has variables and constraints representing embodiment, structure and assembly, and dynamic flow and energy balance information of the process layout. Designing for safety begins with a schematic design model, as graphically illustrated in Fig. 5.85, and development of a systems hierarchical structure as graphically illustrated in Fig. 5.86.

The *treeview* illustrated in the left column of Fig. 5.86 enables designers to view selected equipment (assemblies, sub-assemblies and components) in their cascaded systems hierarchical structure.

The equipment and their codes are related according to the following systems breakdown structure (SBS):

- components,
- assemblies,
- systems,
- sections,
- operations,
- plant.

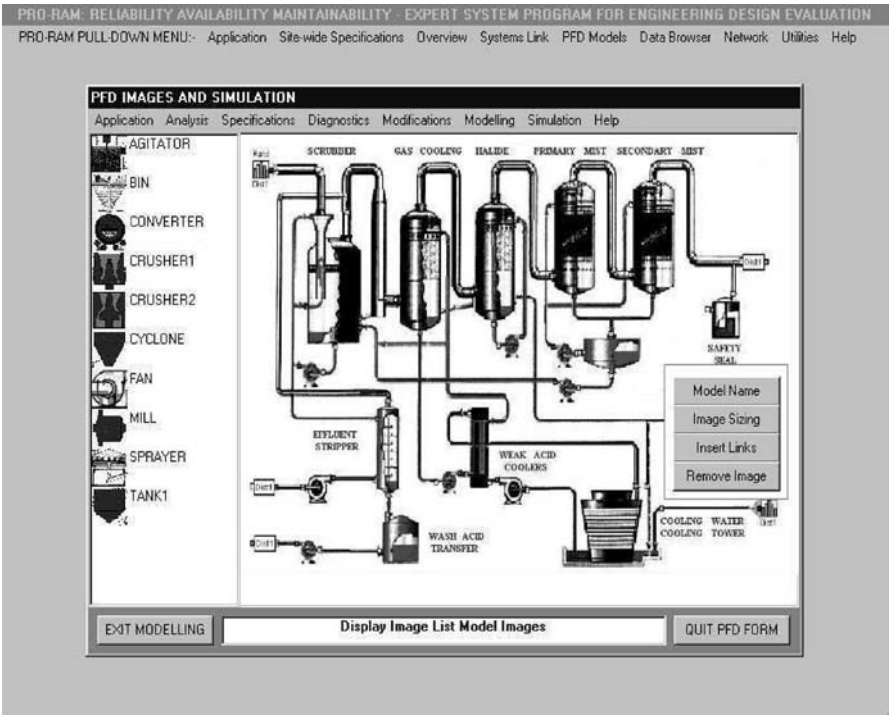


Fig. 5.85 Designing for safety using systems modelling

A selection facility in the *treeview*, alongside the selected component, enables the designer to directly access the component's specific technical specifications, or spares bill of materials (BOM).

Equipment technical data illustrated in Fig. 5.87 automatically format the technical attributes relevant to each type of equipment that is selected in the design process.

The equipment technical data document is structured into three sectors:

- technical data obtained from the technical data worksheet, relevant to the equipment's physical and rating data, as well as performance measures and performance operating, and property attributes that are considered during the design process,
- technical specifications obtained from an assessment and evaluation of the required process and/or system design specifications,
- acquisition data obtained from manufacturer/vendor data sheets, once equipment technical specifications have been finalised during the detail design phase of the engineering design process.

A feature of the systems modelling option in the AIB blackboard model is to determine system failure logic from *network diagrams* or *fault-tree diagrams*, through *Monte Carlo (MC) simulation*.

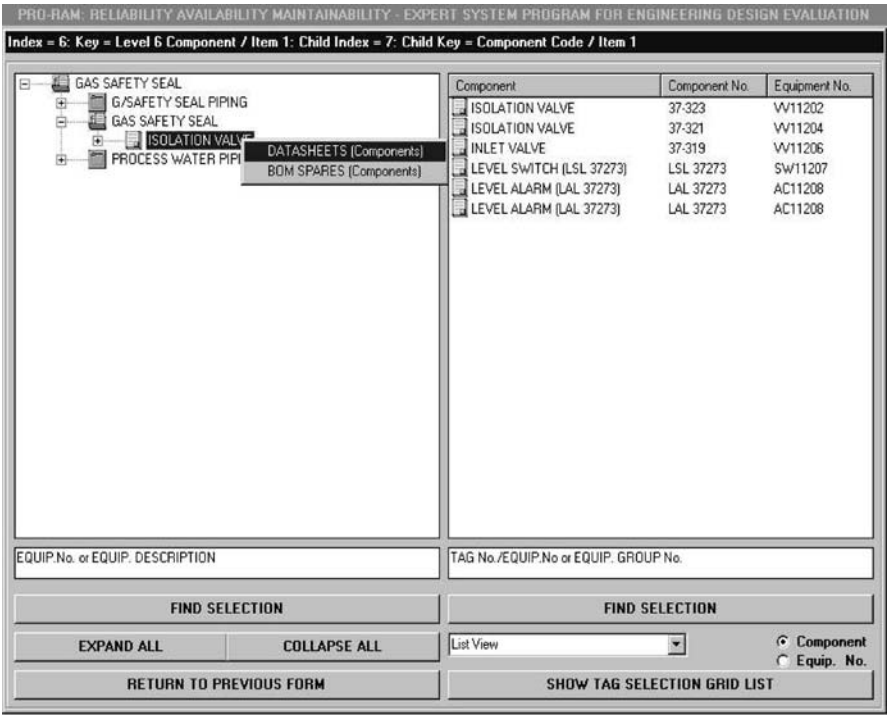


Fig. 5.86 Treeview of systems hierarchical structure

Figure 5.88 illustrates the use of the *network diagram* in determining potential system failures in a parallel control valve configuration of a high-integrity protection system (HIPS). Isograph's AvSim© Availability Simulation Model (Isograph 2001) has been imbedded in the AIB blackboard for its powerful network diagramming capability, especially in constructing block diagrams. The network diagram consists of blocks and nodes connected together in a parallel (and/or series) arrangement. The blocks in the network diagram usually represent potential component or sub-system failures, although they may also be used to represent other events such as operator actions, which may affect the reliability of the system under study. The nodes in the network diagram are used to position connecting lines and indicate voting arrangements. The complete system network diagram will consist of either a single node or block on the left-hand side of the diagram (input node or block) connected via intermediate nodes and blocks to a single node or block on the right-hand side of the diagram (output node or block). A complete system network diagram can have only one input node or block and one output node or block. In addition, all the intermediate nodes and blocks must be connected. The entire system network diagram represents ways in which component and sub-system failures will interact to cause the system to fail.

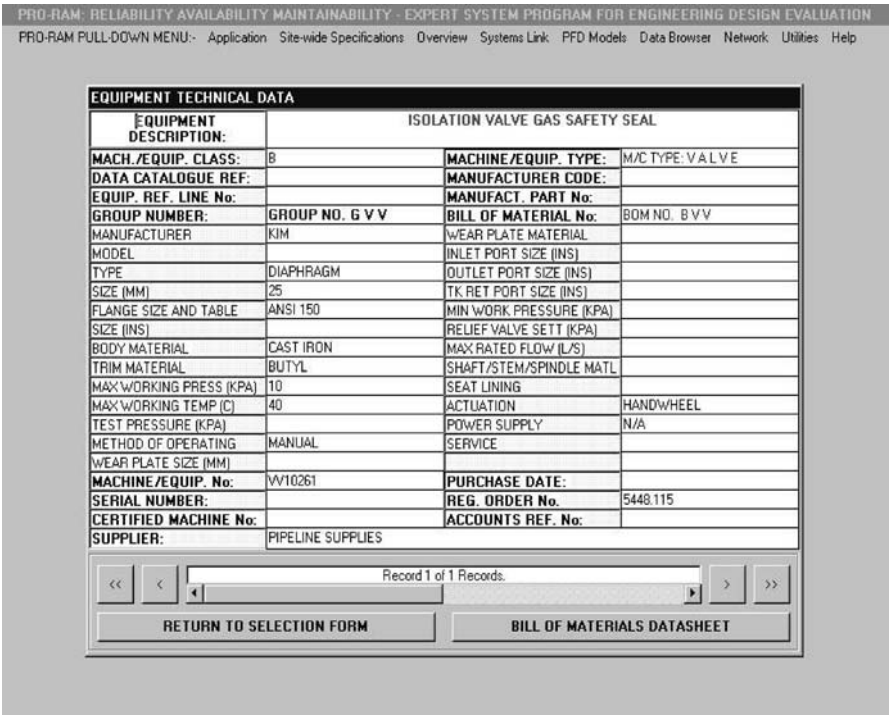


Fig. 5.87 Technical data sheets for modelling safety

Monte Carlo simulation is employed to estimate system and sub-system parameters such as number of expected failures, unavailability, system capacity, etc. The process involves synthesising system performance over a given number of simulation runs. In effect, each simulation run emulates how the system might perform in real life, based on the input data provided by the blackboard system's knowledge base. The input data can be divided into two categories: a *failure logic diagram*, and quantitative *failure and/or maintenance parameters*. The logic diagram (either a fault tree or a network diagram, in this case) informs the knowledge base how component failures interact to cause system failures. The failure and maintenance parameters indicate how often components are likely to fail and how quickly they should be restored to service. By performing many simulation runs, a statistical picture of the system performance is established. Monte Carlo simulation must emulate the chance variations that will affect system performance in real life. To do this, the model must generate random numbers that form a uniform distribution. Simulation methods are generally employed in reliability studies when deterministic methods are incapable of modelling strong dependencies between failures. In addition, simulation can readily assess the reliability behaviour of repairable components with non-constant failure or repair rates.

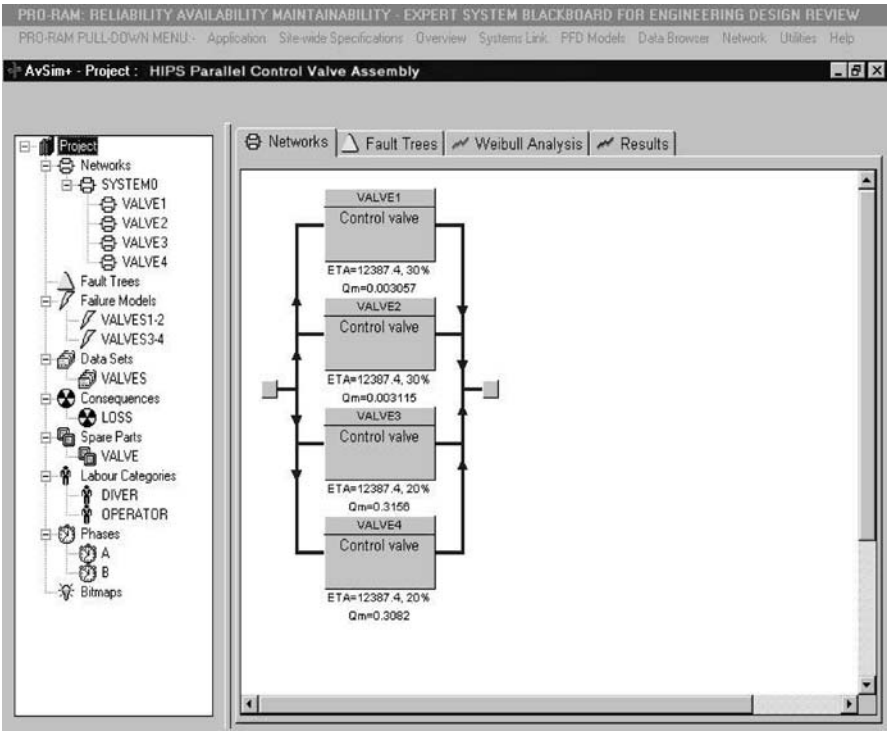


Fig. 5.88 Monte Carlo simulation of RBD and FTA models

During the simulation process, the model will be able to determine whether the system will fail, by examining the developed network diagram. The model does this by determining whether there are any open paths from the input node or block to the output node or block. An open path is a path that does not cross any failed component or sub-system blocks.

Network diagrams may also be used to represent *voting* arrangements. Nodes to the right of a parallel arrangement may be given a vote number to indicate how many success paths must be available through the parallel arrangement (if a vote number is not specified, then only one path need be available). The simple parallel arrangement of the four blocks 1, 2, 3 and 4 in Fig. 5.88, with a vote number (number of available paths required for success) of 2, would result in the *truth table* given in Table 5.27.

Figure 5.89 illustrates the use of the *fault-tree diagram* in determining potential system failures in a parallel control valve configuration of a high-integrity protection system (HIPS). This is developed from the imbedded Isograph AvSim© Availability Simulation Model (Isograph 2001). Fault-tree diagrams graphically represent the interaction of failures and other events within a system. Basic events at the bottom of the fault tree are linked via logic symbols (known as gates) to one or more TOP events. These TOP events represent identified hazards or system failure modes for

Table 5.27 Simple 2-out-of-4 vote arrangement truth table

Valve 1	Valve 2	Valve 3	Valve 4	System
Working	Working	Working	Working	Working
Failed	Working	Working	Working	Working
Working	Failed	Working	Working	Working
Working	Working	Failed	Working	Working
Working	Working	Working	Failed	Working
Working	Working	Failed	Failed	Working
Working	Failed	Working	Failed	Working
Working	Failed	Failed	Working	Working
Failed	Working	Working	Working	Failed
Failed	Working	Failed	Failed	Working
Failed	Failed	Working	Working	Working
Working	Failed	Failed	Failed	Failed
Failed	Working	Failed	Failed	Failed
Failed	Failed	Working	Failed	Failed
Failed	Failed	Failed	Working	Failed
Failed	Failed	Failed	Failed	Failed

which predicted reliability or availability data are required. Basic events at the bottom of the fault tree generally represent component failures, although they may also represent other events such as operator actions. Fault trees may be used to analyse large and complex systems, and are particularly adept at representing and analysing redundancy arrangements.

Figures 5.90 and 5.91 illustrate the Monte Carlo simulation results in the form of a Weibull cumulative failure probability graph, and an unavailability profile of the HIPS.

The *Weibull analysis* module (Isograph 2001) analyses the simulation data by assigning probability distributions that represent the failure or repair characteristics of a given failure mode. In the integration of complex systems, the purpose of determining equipment criticality, or combinations of critical equipment, is to assess the times to wear-out failures. The Weibull distribution is particularly useful because it can be applied to all three phases of the hazard rate curve. The failure distribution assigned to a given set of times to failure (known as a dataset) may be assigned to failure models that are attached to blocks in a network diagram or events in a fault-tree diagram. The model automatically fits the selected distribution to the data and displays the results graphically in the form of cumulative probability plots, unconditional probability density plots, and conditional probability density plots.

Figure 5.90 illustrates Monte Carlo simulation results of unreliability displayed in the form of a Weibull cumulative failure probability graph.

Unavailability profile graphs display the mean unavailability values for each time interval. Unavailability values may be displayed for several sub-systems, assemblies and components of a system, or integrated systems, which are concurrently being designed. Figure 5.91 illustrates the Monte Carlo simulation results in the form of an unavailability profile of the high-integrity protection system (HIPS).

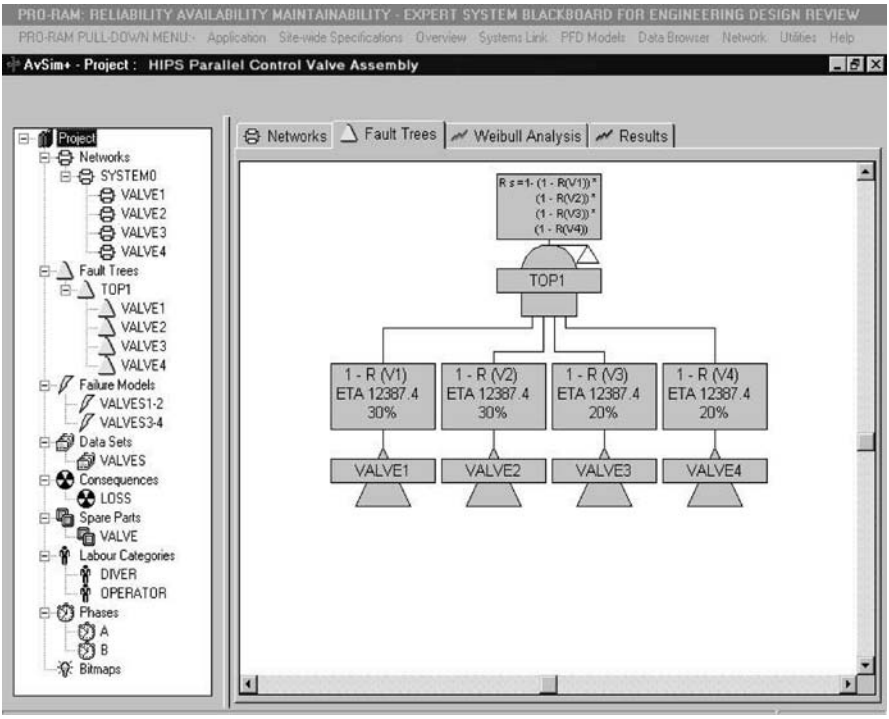
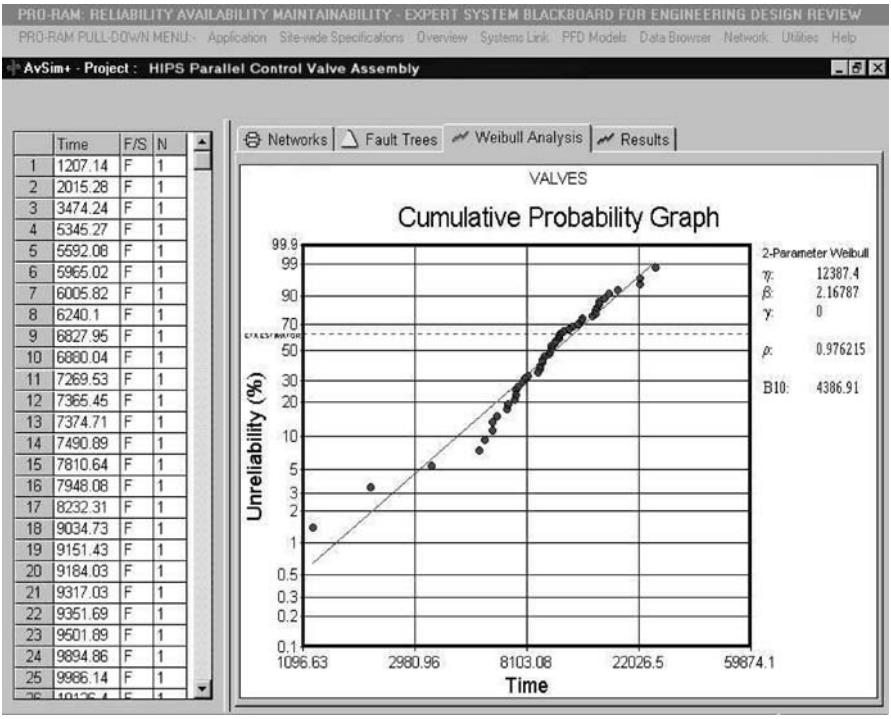


Fig. 5.89 FTA modelling in designing for safety

As stated in Sect. 4.4.1, dynamic system simulation in engineering design provides for *virtual prototyping* of engineering processes, making design verification faster and less expensive. To fully exploit the advantages of virtual prototyping, dynamic system simulation is the most efficient and effective. Dynamic system simulation provides various design teams in a collaborative design environment with immediate feedback on design decisions, allowing for a comprehensive exploration of design alternatives and for optimal final designs. However, dynamic simulation modelling can be very complex, resulting in a need for simulation models to be easy to create and analyse.

To take full advantage of virtual prototyping (i.e. developing PEMs), it is necessary for *dynamic system simulation modelling* to be integrated with the design environment (through the AIB blackboard), and to provide a simple and intuitive user interface that requires a minimum of analysis expertise. Figure 5.92 illustrates the AIB blackboard model selection menu with the process flow diagramming (PFD) option that includes systems modelling and systems simulation. Access to a simulation modelling capability by design engineers in a collaborative design environment is a powerful feature provided by the AIB blackboard.

Many engineered installations have a modular architecture that is based on the optimum selection and composition of systems, assemblies and components from



	Time	F/S	N
1	1207.14	F	1
2	2015.28	F	1
3	3474.24	F	1
4	5345.27	F	1
5	5592.08	F	1
6	5965.02	F	1
7	6005.82	F	1
8	6240.1	F	1
9	6827.95	F	1
10	6880.04	F	1
11	7269.53	F	1
12	7365.45	F	1
13	7374.71	F	1
14	7490.89	F	1
15	7810.64	F	1
16	7948.08	F	1
17	8232.31	F	1
18	9034.73	F	1
19	9151.43	F	1
20	9184.03	F	1
21	9317.03	F	1
22	9351.69	F	1
23	9501.89	F	1
24	9894.86	F	1
25	9986.14	F	1

Fig. 5.90 Weibull cumulative failure probability graph of HIPS

older designs. When the new design is created, these system compositions are selected and then connected together in a systems configuration. Figures 5.93 to 5.97 illustrate the overall systems configuration of an extend process simulation model with PEM blocks.

Multiple logical flow configurations can represent a particular system composition, and are bound to the system's configuration interface. The industrial systems simulation option of the Extend© Performance Modelling (Extend 2001) software has been modified and imbedded into the AIB blackboard to include a wide range of *process equipment models (PEMs)*. These PEMs are held in a general systems simulation database library that can be accessed by various programming options in the AIB blackboard (either imbedded as third-party software or as developed application software). A PEM system can be represented either as a single block (*model component*) or as a configuration of several blocks. These configurations are equivalent PEM specifications of the same blocks, and the choice of configuration is independent of the PEM system behaviour.

Figure 5.93 shows a specific section's *process flow diagram (PFD)* consisting of ten systems, each system graphically represented by a virtual prototype *process equipment model (PEM)*. The systems, or PEM blocks, are linked together with *logical flows*.

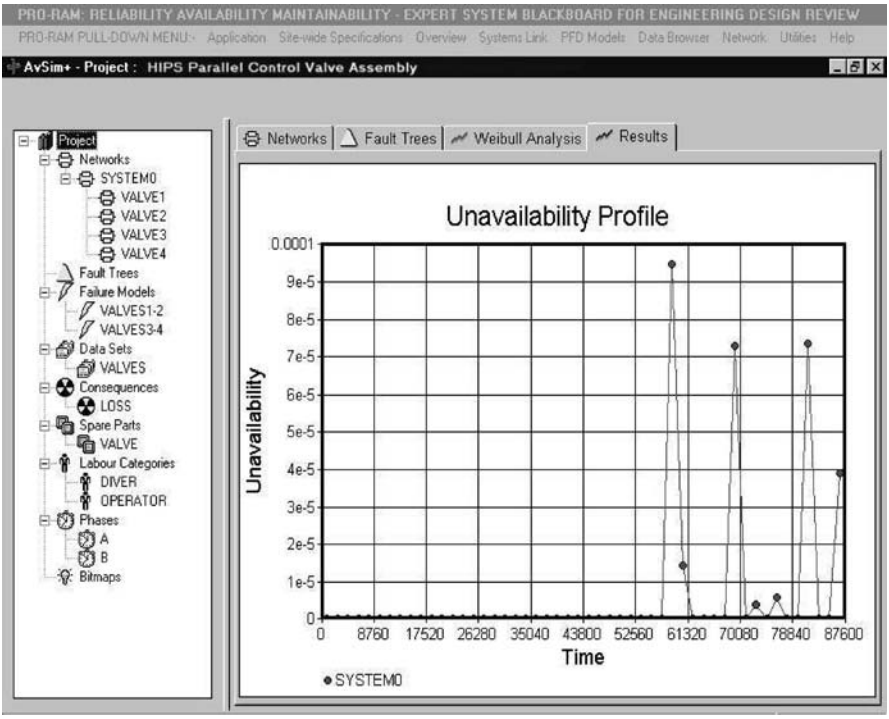


Fig. 5.91 Profile modelling in designing for safety

In many process designs, the physical or real-world systems are designed using *model components*. In such processes, these model components are selected, configured and assembled in such a way that the design specifications are met. A model component is a modular design entity with a complete specification describing how it may be connected to other model components in a *model configuration*. A model configuration is created when two or more model components are connected to each other via their *interfaces*. A model component can itself encapsulate a configuration of numerous model components, thus allowing for a hierarchical structure of sub-models as illustrated in Fig. 5.94.

Each block pertaining to a PEM has connectors that are the interface points of the block. Connections are lines used to specify the logical flow from one model component to another, as illustrated in Fig. 5.94. As will be shown later, a model component is instantiated in the design by specifying *instantiation parameters* that describe its specification.

Figures 5.95 and 5.96 illustrate the PEM simulation models process information. This information is generated either in a document layout of system performance variables (such as system contents, flows and surges, in the case of Fig. 5.95) or in a graphical display of system performance variables (such as in the case of Fig. 5.96).

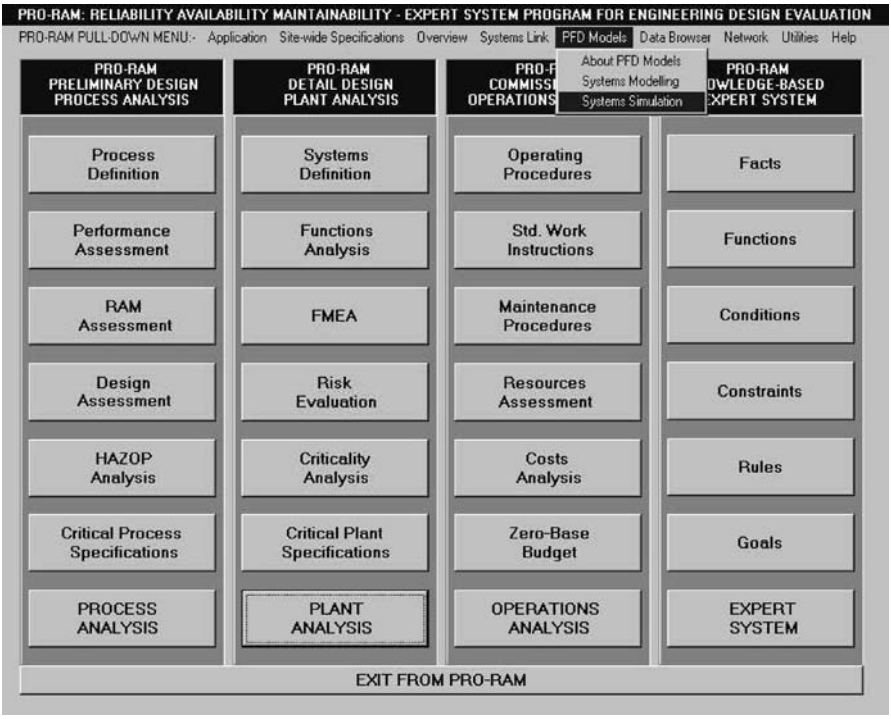


Fig. 5.92 AIB blackboard model with system simulation option

Figure 5.95 illustrates system performance variables that describe PEM specifications. In this case, the PEM specifications are represented by the modelling component called 'holding tank', relating to the PEM system, 'reverse jet scrubber'. These PEM specifications include performance variables such as operating contents, maximum contents, minimum contents, initial inflow, final inflow, initial outflow, final outflow, initial contents, final contents, initial flow surge, final flow surge, and accumulative surge. Several simulation run options are available, such as for operating contents going below minimum contents, or for steady-state flow (outflow=inflow).

The *graphical display* (plotter) shows both a graphical representation of the process values of a performance variable during a simulation run, as well as a table of the numerical values of the performance variable. A powerful feature of the graphical display in engineering design is that plots of a performance variable taken in previous simulation runs is 'remembered' (up to four previous simulation runs), to allow for a comparative analysis in the event a performance variable is changed for design cost/performance trade-off. Such a trade-off would not be considered in assessing safety criteria related to a specific performance variable, where an increase in safety might result in a decrease in performance as shown in previous simulation runs.

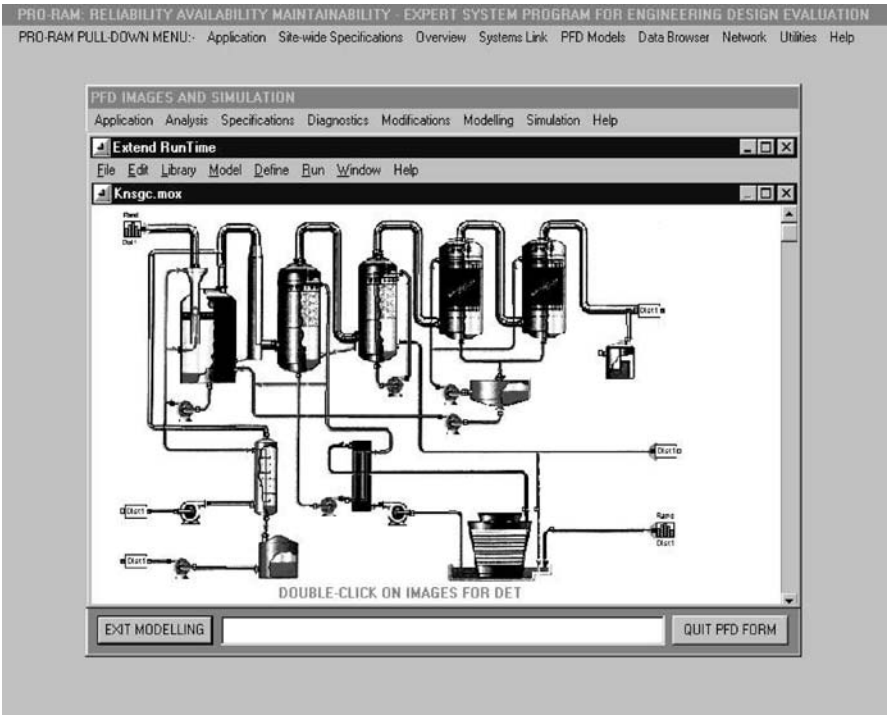


Fig. 5.93 PFD for simulation modelling

Figure 5.96 illustrates the graphical display model component for system behaviour of the performance variable 'operating contents' of the PEM system 'reverse jet scrubber', indicating a trend towards steady state.

Petri net-based optimisation algorithms are usefully applied in dynamic systems simulation—in this case, the determination of pressure surge through a continuous process flow line. Petri nets have been used as mathematical graphical tools for modelling and analysing systems of which the dynamic behaviours are characterised by synchronous and distributed operation, as well as non-determinism. A basic Petri net structure consists of *places* and *transitions* interconnected by directed *arcs*. Places are denoted by circles and represent *conditions*, while transitions are denoted by bars or rectangles and represent *events*. The directed arcs in a Petri net represent flow of control where the occurrence of events is controlled by a set of conditions that can be either instantaneous or gradual (averaged).

The pressure surge Petri net depicted in Fig. 5.97 includes *conditions* of flow surge criteria such as outlet diameter and fluid modulus, together with *events* representing the combination and manipulation of criteria in the flow surge algorithm to obtain results in graphical displays.

Design automation (DA) environments typically contain a design representation or design database through which the design is controlled. The design automation

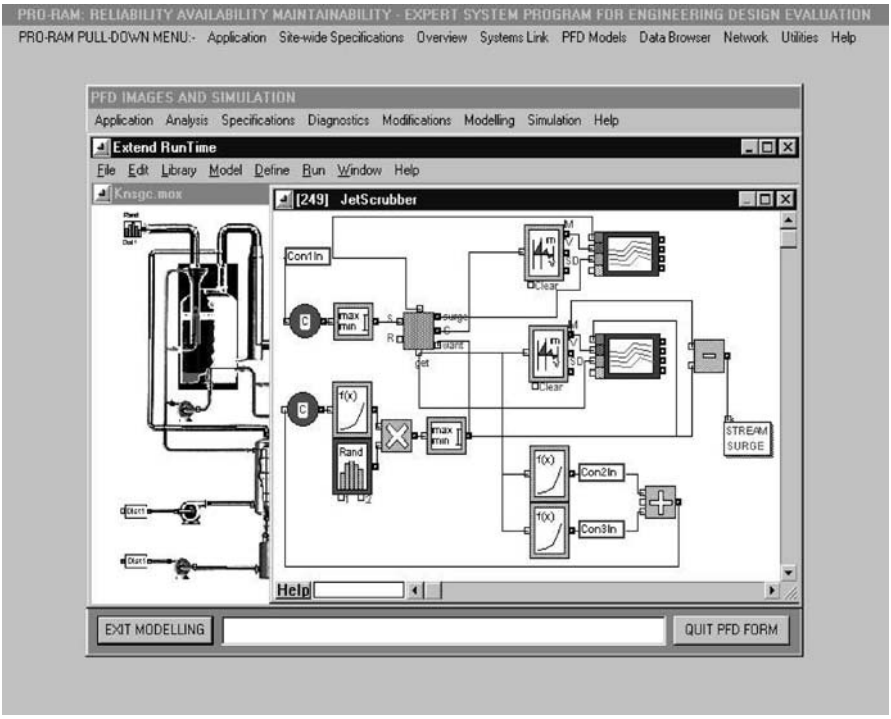


Fig. 5.94 PEMs for simulation modelling

environment usually interacts with a set of resident *computer aided design (CAD)* tools and will attempt to act as a manager of the CAD tools by handling input/output requirements, invocation parameters and, possibly, automatically sequencing the CAD tools. Thus, a DA environment provides a design framework that, in effect, shields the designer from cumbersome details and enables the designer to work at a high level of abstraction. Design automation environments have great potential in CAD because they can encapsulate expert design knowledge as well as rapidly changing domain knowledge, typical of process engineering design. Since they can be easily extended and modified, rule-based systems allow for limited automated design.

Figure 5.98 illustrates the AIB blackboard data browser option with access to a database library of integrated CAD data relevant to each PEM.

CAD models provide a comprehensive and detailed knowledge source for the AIB blackboard, which can be integrated with an expert systems knowledge base for process information. The most useful CAD model for knowledge integration is the three-dimensional CAD (3D CAD), which entails parametric solid modelling that requires the user to apply what is referred to as '*design intent*'. Some software packages provide the ability to edit parametric as well as non-parametric geometry without the need to understand or undo the design intent history of the

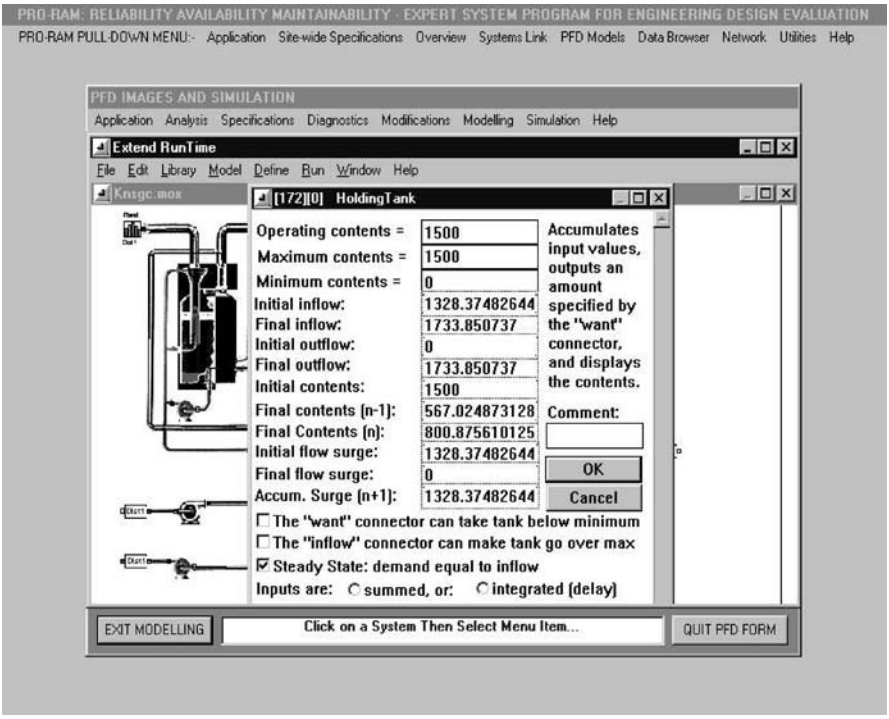


Fig. 5.95 PEM simulation model performance variables for process information

geometry by use of direct modelling functionality. Parametric designs require the user to consider the design sequence carefully, especially in a collaborative design environment. What may be a simple design now could be worst case later.

Figure 5.99 shows a three-dimensional CAD model of process configuration information, accessed from a database library of integrated CAD data relevant to each PEM in the AIB blackboard.

Knowledge training is an important application of three-dimensional CAD modelling, especially for training operators and engineers for the engineered installation, notably during the ramp-up and warranty stages. A CAD modelling system can be seen as built up from the interaction of a graphical user interface (GUI) with boundary representation data via a geometric modelling kernel. A geometry constraint engine is employed to manage the associative relationships between geometry, such as wire frame geometry in a schematic design or components in a detail design. Advanced capabilities of these associative relationships have led to a new form of prototyping called digital prototyping. In contrast to physical prototypes, digital prototypes allow for design verification and testing on screen, enabling three-dimensional CAD to be more than simply a documentation tool (representing designs in graphical format) but, rather, a more robust designing tool that assists in the design process as well as post-design testing and training.

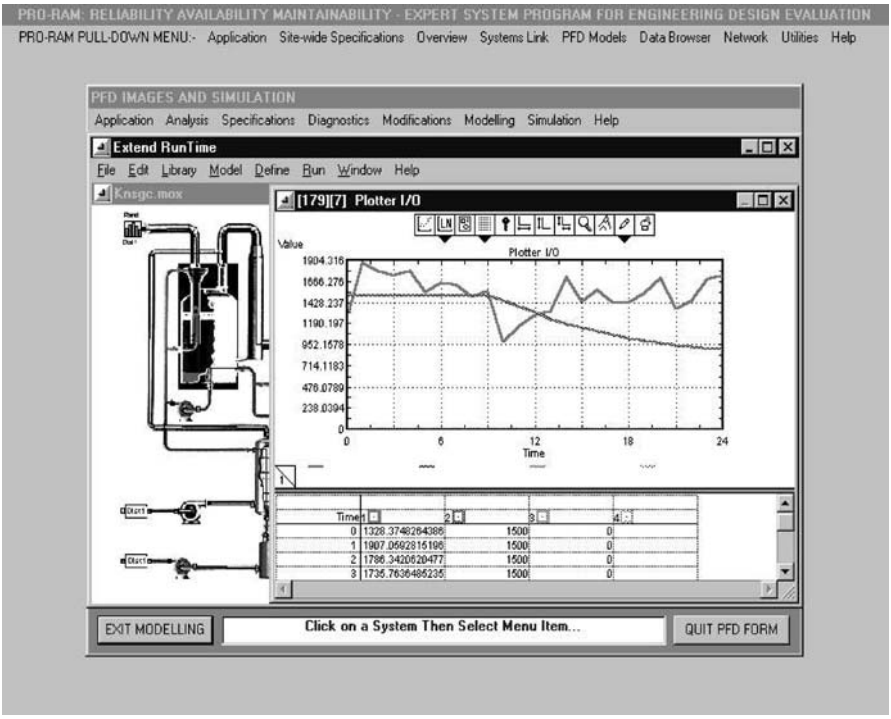


Fig. 5.96 PEM simulation model graphical display of process information

Figure 5.100 shows a typical CAD integrated training data library in the AIB blackboard of performance variable data relevant to each PEM.

Artificial neural network (ANN) computation, unlike more analytically based information processing methods, effectively explores the information contained within input data, without further assumptions. Statistical methods are based on certain assumptions about the input data (i.e. a priori probabilities, probability density functions, etc.). Artificial intelligence encodes deductive human knowledge with simple IF THEN rules, performing inference (search) on these rules to reach a conclusion. Artificial neural networks, on the other hand, identify relationships in the input datasets, through an iterative presentation of the data and intrinsic mapping characteristics of neural topologies (referred to as *learning*). There are two basic phases in neural network operation: the training or *learning phase*, where sample data are repeatedly presented to the network, while their *weights* are updated to obtain a desired response; and the recall or *retrieval phase*, where the trained network is applied to *prototype* data.

Figure 5.101 shows the AIB blackboard ANN computation option with access to an imbedded NeuralExpert© program (NeuroDimension 2001).

A *neural expert program* (Lefebvre et al. 2003) is a specific knowledge source of the AIB blackboard for processing time-varying information, such as non-linear

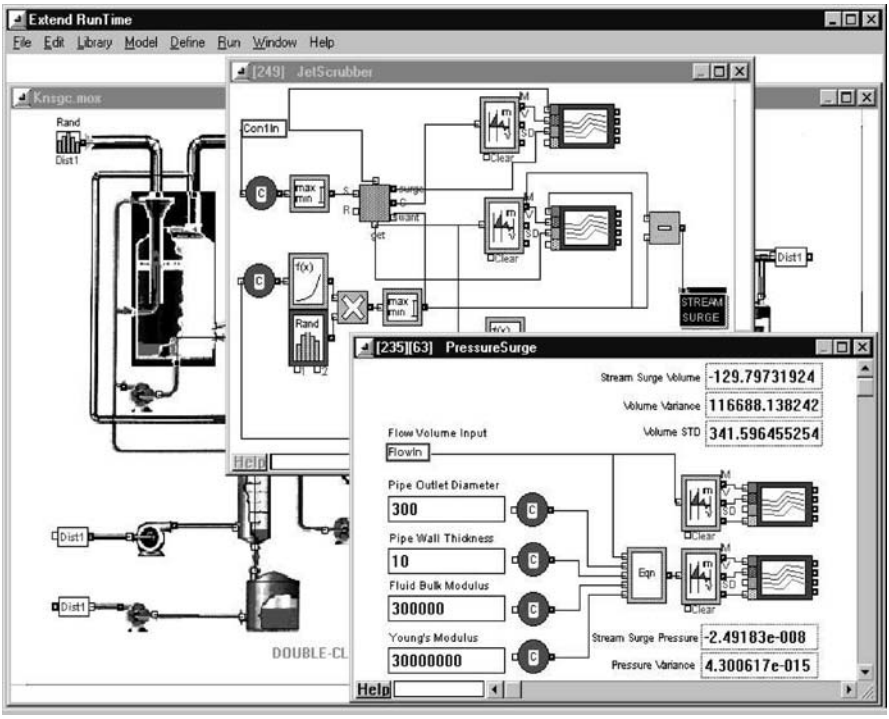


Fig. 5.97 Petri net-based optimisation algorithms in system simulation

dynamic modelling, time series prediction, adaptive control, etc., of various engineering design problems. A typical design problem that is ideal for ANN modelling is the formulation and evaluation of stream surge pressures in continuous flow processes, given in the simulation option of the AIB blackboard as illustrated in Fig. 5.97.

The NeuralExpert© (NeuroDimension 2001) program, imbedded in the AIB blackboard, asks specific questions and intelligently builds an ANN. The first step in building an ANN is the specification of the *problem type*, as illustrated in Fig. 5.102. The four currently available problem types in the NeuralExpert are classification, function approximation, prediction, and clustering. Once a problem type is selected, the program configures the parameters based on a description of the problem. These settings can be modified in the AIB blackboard, or in the NeuralExpert.

Input data selection is the next step in constructing an ANN model. The input file selection panel specifies where the input data file is located by choosing the 'browse' button and searching through the standard Windows tree structure to find the relevant file referenced in the AIB blackboard database, or by clicking on the triangle at the right edge of the text box to indicate a list of recently used text files in the NeuralExpert.

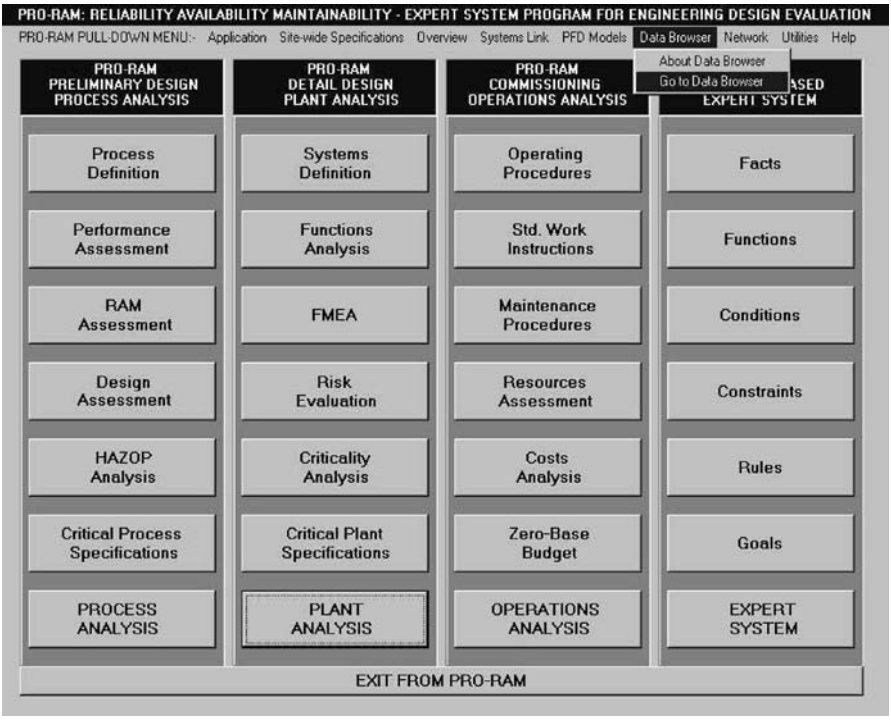


Fig. 5.98 AIB blackboard model with CAD data browser option

Figure 5.103 illustrates typical input data attributes for the example used to determine the stream surge pressure given in the simulation model option of the AIB blackboard illustrated in Fig. 5.97. In this case, the sample data would represent x_1, x_2, x_3, x_4 values of the input attributes. The pressure surge Petri net given in Fig. 5.97 includes *conditions* of flow surge criteria that now become the ANN input attributes, such as the pipe outlet diameter, pipe wall thickness, the fluid bulk modulus, and Young's modulus. The goal is to train the ANN to determine the stream surge pressure (desired output) based on these attributes.

Typical *computational problems* associated with artificial neural network programs, with regard to specific as well as general engineering design requirements, include the following:

Classification problems are those where the goal is to label each input with a specified classification. A simple example of a classification problem is to label process flows as 'fluids' and/or 'solids' for balancing (the two classes, also the desired output) using their volume, mass and viscosity (the input). The input can be either numeric or symbolic but the output is symbolic in nature. For example, the desired output in the process balancing problem is the ratio of fluids and solids, and not necessarily a numeric value of each.

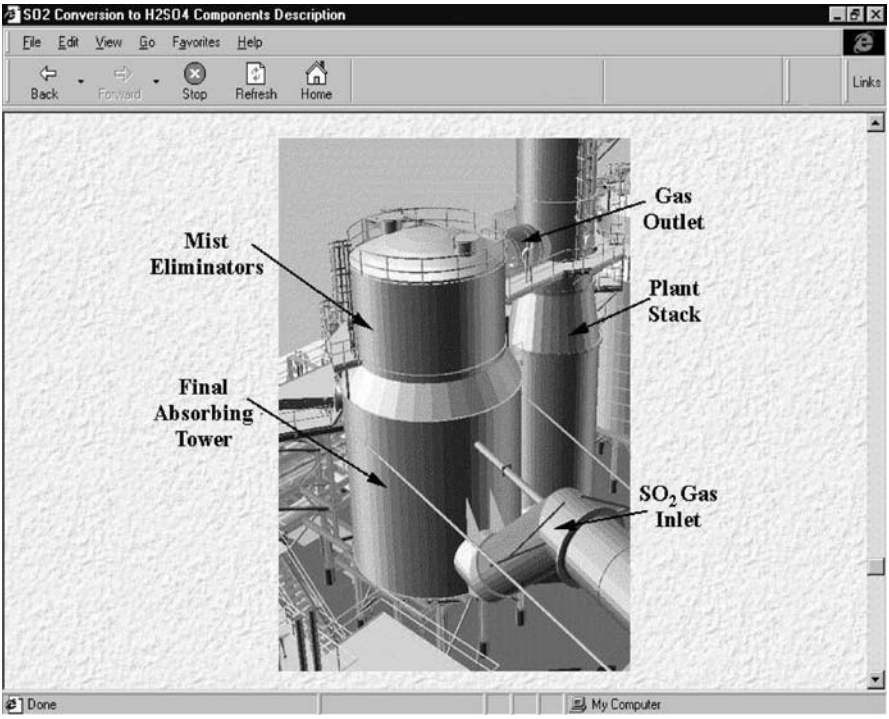


Fig. 5.99 Three-dimensional CAD integrated model for process information

Function approximation problems are those where the goal is to determine a numeric value, given a set of inputs. This is similar to classification problems, except that the output is numeric. An example is to determine the stream surge pressure (desired output) in numeric values, given the pipe outlet diameter, the pipe wall thickness, the fluid bulk modulus and Young's modulus. These problems are called function approximation because the ANN will try to approximate the functional relationship between the input and desired output. Prediction problems are also function approximation problems, except that they use *temporal* information (e.g. the past history of the input data) to make predictions of the available data.

Prediction problems are those where the goal is to determine an output, given a set of inputs and the past history of the inputs. The main difference between prediction problems and the others is that prediction problems use the current input and previous inputs (the temporal history of the input) to determine either the current value of the output or a future value of a signal. A typical example is to predict process pump operating performance (desired output) from motor current and delivery pressure performance values.

Clustering problems nformation is to be extracted from input data without any desired output. For example, in the analysis of process faults in designing for safety, the faults can be clustered according to the severity of hazard consequences risk.

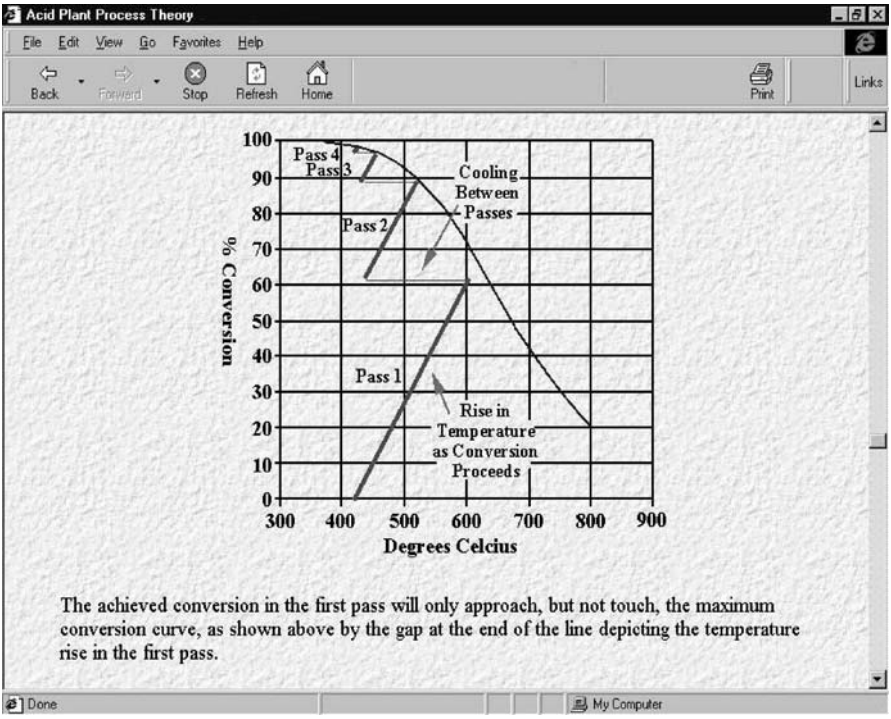


Fig. 5.100 CAD integrated models for process information

The fundamental difference between the clustering problem and the others is that there is no desired output (therefore, there is no error and the ANN model cannot be trained using back propagation).

For *classification problems* to label each input with a specified classification, the option is given to randomise the order of the data before presenting these to the network. Neural networks train better if the presentation of the data is not ordered. For example, if the design problem requires classifying between two classes, 'fluids' and/or 'solids', for balancing these two classes (as well as the desired output) using their volume, mass and viscosity (the input), the network will train much better if the fluids and solids data are intermixed. If the data are highly ordered, they should be randomised before training the artificial neural network.

One of the primary goals in training neural networks in the process of 'iterative prediction' is to ensure that the network performs well on data that it has not been trained on (called 'generalisation'). The standard method of ensuring good generalisation is to divide the training data into multiple datasets or samples, as indicated in Fig. 5.104. The most common datasets are the training, cross validation, and testing datasets.

The *cross validation dataset* is used by the network during training. Periodically while training on the training dataset, the network is tested for performance

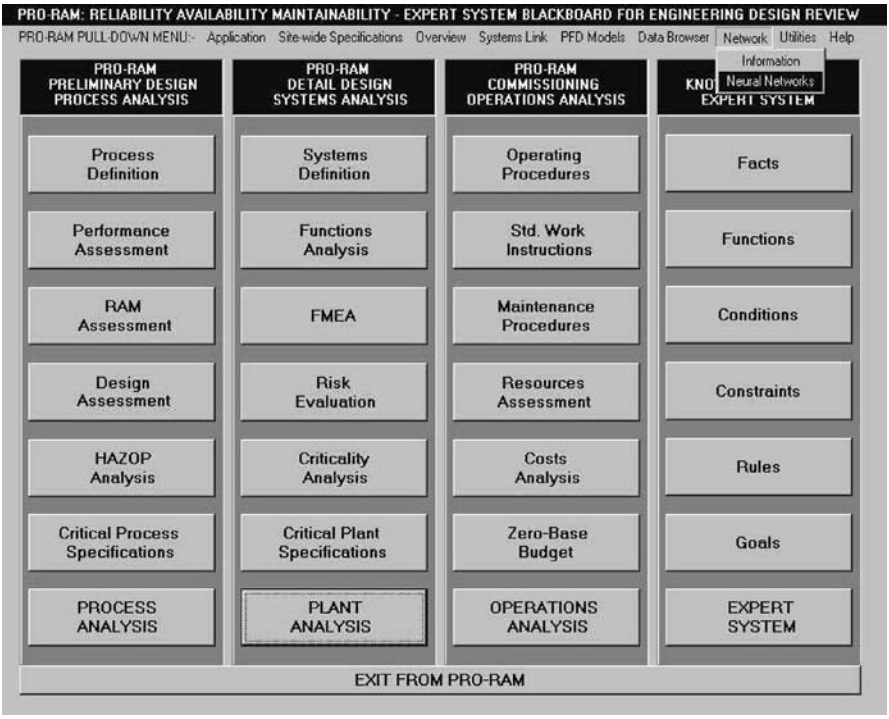


Fig. 5.101 ANN computation option in the AIB blackboard

on the cross validation set. During this testing, the weights are not trained but the performance of the network on the cross validation set is saved and compared to past values. The network shows signs of becoming *over-trained* on the training data when the cross validation performance begins to degrade. Thus, the cross validation dataset is used to determine when the network has been trained as best as possible, without over-training (i.e. maximum generalisation).

Although the network is not trained with the cross validation set, it uses the cross validation set to choose the best set of weights. Therefore, it is not truly an out-of-sample test of the network. For a true test of the performance of the network, an independent (i.e. out of sample) testing set is used. This provides a true indication of how the network will perform on new data. The 'out of sample testing' panel shown in Fig. 5.105 is used to specify the amount of data to set aside for the testing set.

It is important to find a *minimal network* with a minimum number of free weights that can still learn the problem. The minimal network is more likely to generalise well with new data. Therefore, once a successful training session has been achieved, the process of decreasing the size of the network should commence, and the training repeated until it no longer learns the problem effectively.

The genetic optimisation component shown in Fig. 5.106 implements a *genetic algorithm* to optimise one or more parameters within the neural network. The most common network parameters to optimise are the input columns, the number of

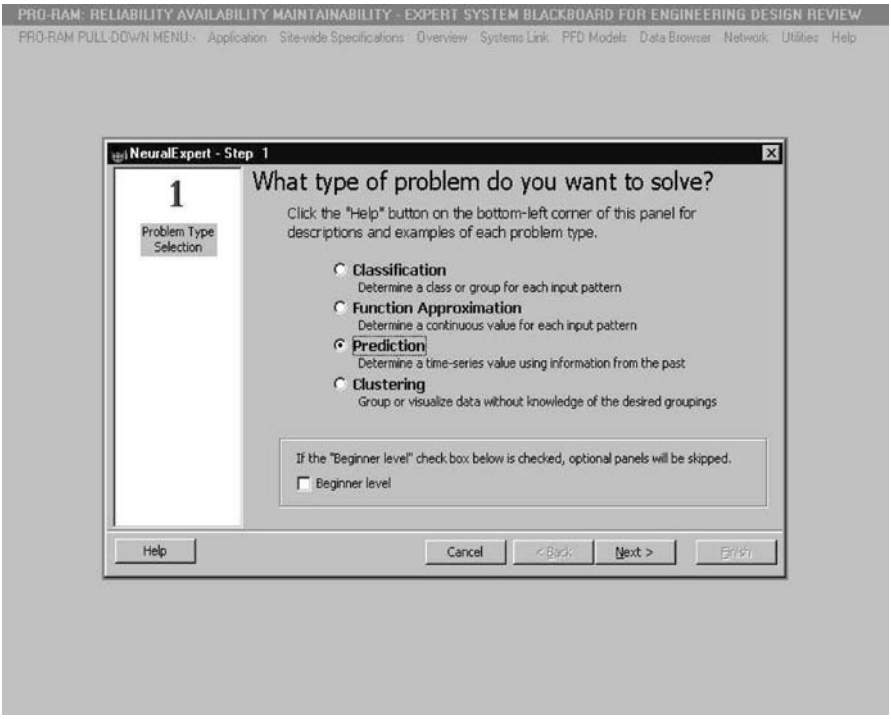


Fig. 5.102 ANN NeuralExpert problem selection

hidden *processing elements (PEs)*, the number of memory taps, and the learning rates. Genetic algorithms combine selection, crossover and mutation operators with the goal of finding the best solution to a problem. Genetic algorithms are general-purpose search algorithms that search for an optimal solution until a specified termination criterion is met.

Network complexity is determined by the size of the neural network in terms of hidden layers and processing elements (neurons). In general, smaller neural networks are preferable over large ones. If a small one can solve the problem sufficiently, then a large one will not only require more training and testing time but also may perform worse on new data. This is the generalisation problem—the larger the neural network, the more free parameters it has to solve the problem. Excessive free parameters may cause the network to over-specialise or to memorise the training data. When this happens, the performance of the training data will be much better than the performance of the cross validation or testing datasets.

The network complexity panel shown in Fig. 5.107 is used to specify the size of the neural network. It is essential to start ANN analysis with a ‘low-complexity’ network, after which analysis can progress to a medium- or high-complexity network to determine if the performance results are significantly better. A disadvantage is that medium- or high-complexity networks generally require a large amount of data.

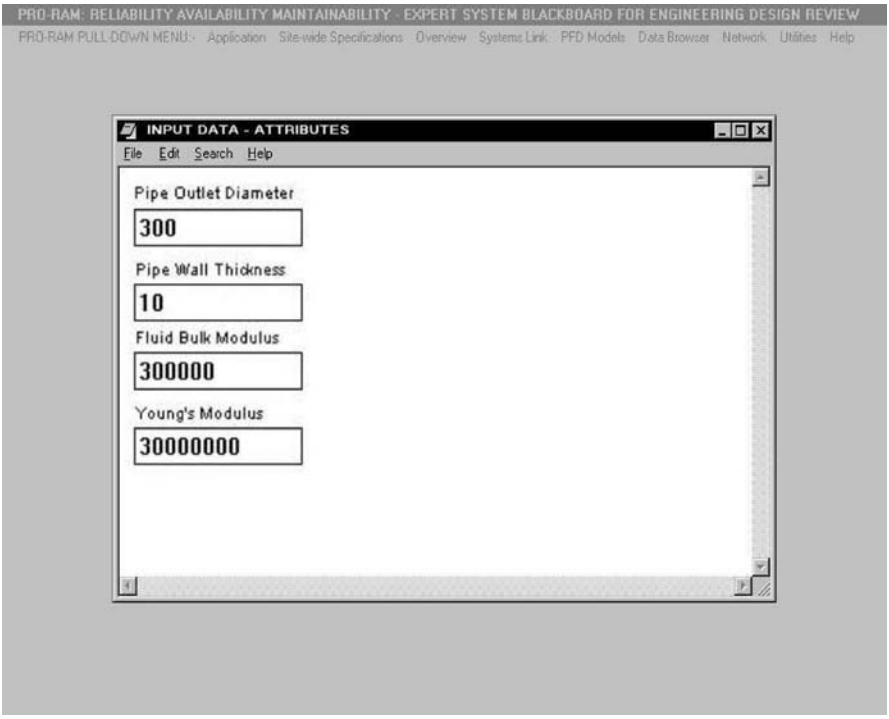


Fig. 5.103 ANN NeuralExpert example input data attributes

In the NeuralExpert© (NeuroDimension 2001) program, imbedded in the AIB blackboard, several criteria can be used to evaluate the *fitness* of each potential solution. The solution to a problem is called a *chromosome*. A chromosome is made up of a collection of *genes*, which are simply the neural network parameters to be optimised. A genetic algorithm creates an initial population (a collection of chromosomes) and then evaluates this population by training a neural network for each chromosome. It then evolves the population through multiple generations in the search for the best network parameters. Performance measures of the error criterion component provide several values that can be used to measure the performance results of the network for a particular dataset. These are:

- the *mean squared error (MSE)*,
- the *normalised mean squared error (NMSE)*,
- the *percent error (% error)*.

The *mean squared error* (MSE) is defined by the following formula

$$\mathrm{MSE} = \frac{\sum_{j=0}^{P} \sum_{i=0}^{N} (d_{ij} - y_{ij})^2}{NP} \tag{5.115}$$

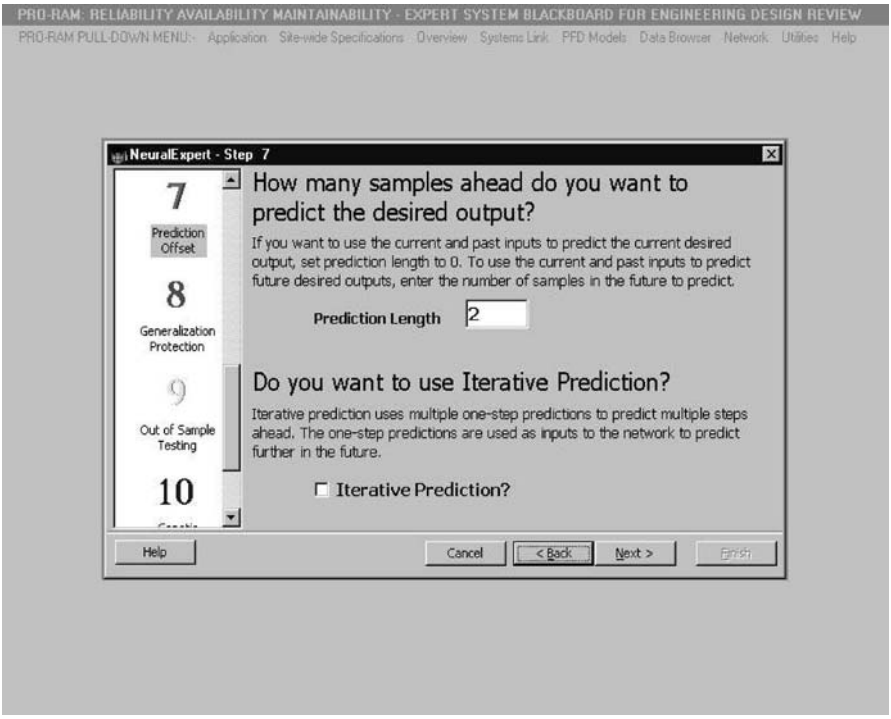


Fig. 5.104 ANN NeuralExpert sampling and prediction

where:

P = number of output processing elements
N = number of exemplars in the dataset
y_{ij} = network output for exemplar i at processing elements j
d_{ij} = desired output for exemplar i at processing elements j.

The *normalised mean squared error* (NMSE) is defined by

$$\text{NMSE} = \frac{P \cdot N \cdot \text{MSE}}{\sum_{j=0}^{P} \left(\frac{N \sum_{i=0}^{N} d_{ij}^2 - \left(\sum_{i=0}^{N} d_{ij} \right)^2}{N} \right)} \tag{5.116}$$

where:

P = number of output processing elements
N = number of exemplars in the dataset
MSE = mean squared error
d_{ij} = desired output for exemplar i at processing elements j.

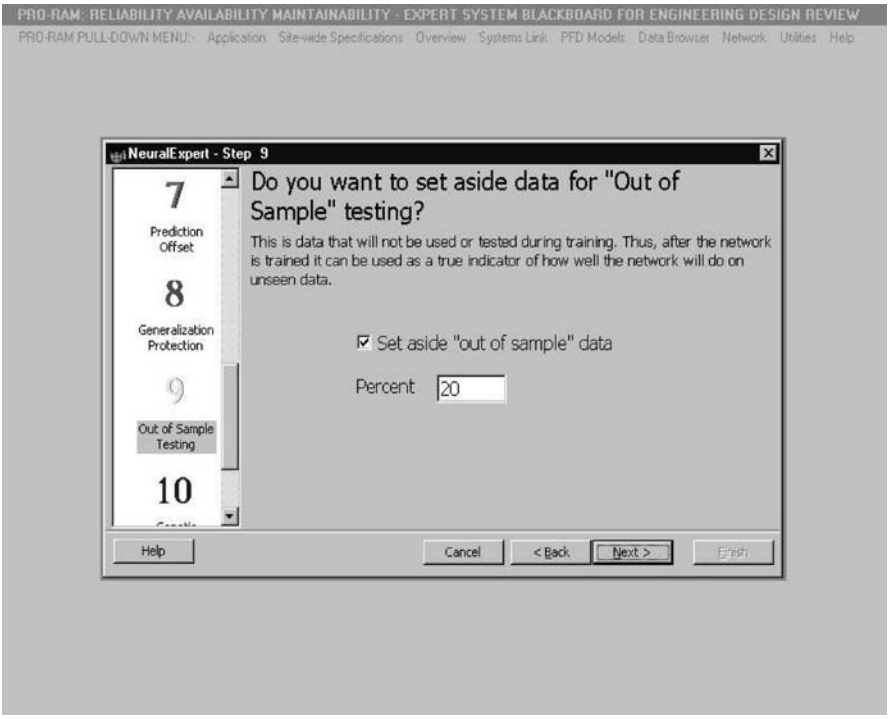


Fig. 5.105 ANN NeuralExpert sampling and testing

The *percent error* (%E) is defined by the following formula

$$\%E = \frac{100}{NP} \sum_{j=0}^{P} \sum_{i=0}^{N} \frac{|d_{ij} - dd_{ij}|}{dd_{ij}} \tag{5.117}$$

where:

P = number of output processing elements
N = number of exemplars in the dataset
d_{ij} = denormalised network output for exemplar i at elements j
dd_{ij} = denormalised desired output for exemplar i at elements j.

Knowledge-based expert systems Expert knowledge of how to solve complex engineering design problems is not often available. *Knowledge-based expert systems* are programs that capture that knowledge and allow its dissemination in the form of structured questions, to be able to determine the reasoning behind a particular design problem's solution. The knowledge-based expert systems incorporated in the AIB blackboard are based on the classical approach to expert systems methodology, which incorporates the following:

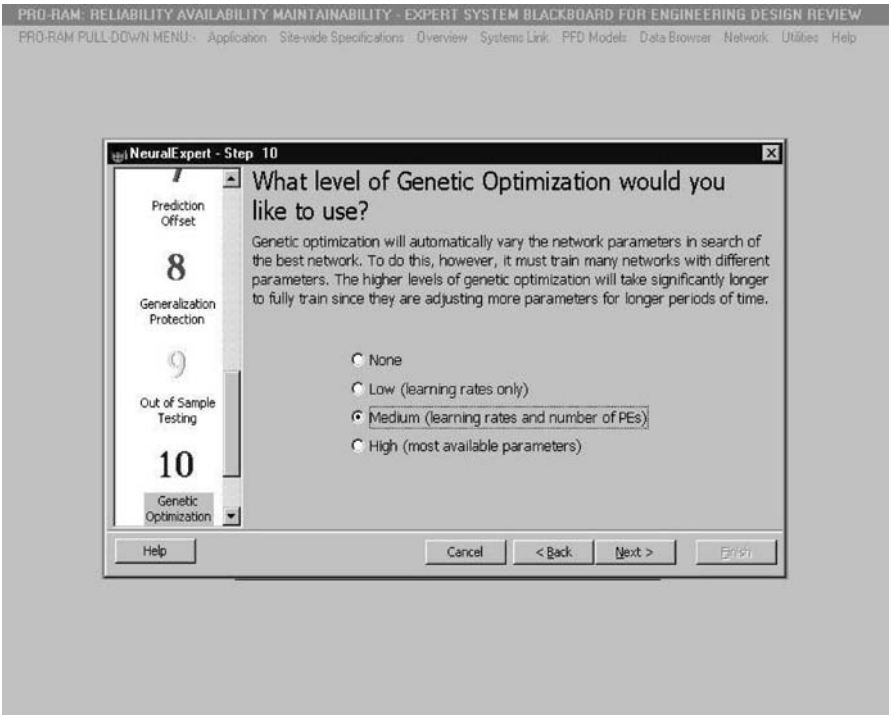


Fig. 5.106 ANN NeuralExpert genetic optimisation

- User interface,
- Working memory,
- Inference engine,
- Facts list,
- Agenda,
- Knowledge base,
- Knowledge acquisition facility.

The user interface is a mechanism by which the user and the expert system communicate. The working memory consists of a global database of facts used by rules. The inference engine controls the overall execution of queries or selections related to problems and their solutions based around the rules. The facts list contains the data on which inferences are derived. An agenda is a list of rules with priorities created by the inference engine, the patterns of which are satisfied by facts in the working memory. The knowledge base contains all the knowledge and rules. The knowledge acquisition facility is an automatic way for the user to enter or modify knowledge in the system, rather than by having all the knowledge explicitly coded at the onset of the expert systems design.

The *user interface* of the AIB blackboard is an object-oriented application in which the designer can point-and-click at digitised graphic process flow diagrams

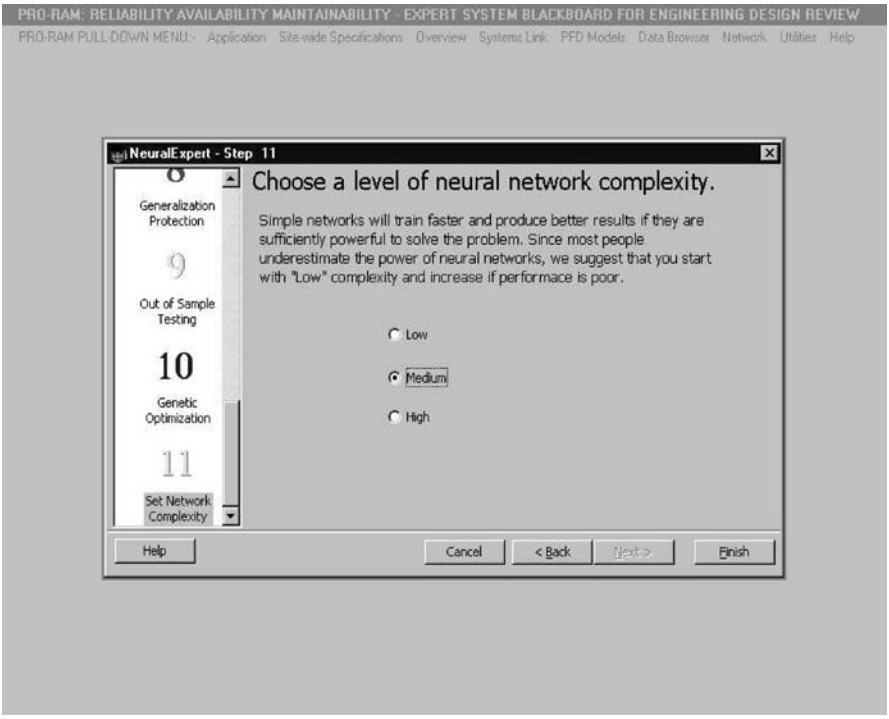


Fig. 5.107 ANN NeuralExpert network complexity

(PFD) of a plant to access specific details of any object shown on the PFD, as well as the object's detailed specifications, diagnostics or performance measures. The diagnostics inference engine contains diagnostic charts and queries relating to failure characteristics, failure conditions, equipment criticality, performance measures, and operating and maintenance strategies.

The knowledge base consists of *facts* and *functions* relating to all the technical data pertaining to process definition, systems definition, performance assessment and analysis, *conditions* and *constraints* relating to equipment failure modes and effects, the level of risk and mitigating maintenance procedures, as well as an assessment of the required resources. Figure 5.108 illustrates the AIB blackboard knowledge base user interface to access the various expert systems with their rules and goals.

A *knowledge-based expert system* emulates the interaction a group of multi-discipline design engineers will have in solving a design problem. The decision trees or *rules* used in a knowledge-based expert system contain the knowledge of the human specialist(s) in a particular field. The inference engine makes use of these rules to solve a problem in achieving set *goals* (design criteria). The end user (designer) asks structured questions until the expert system has reached an optimal solution in

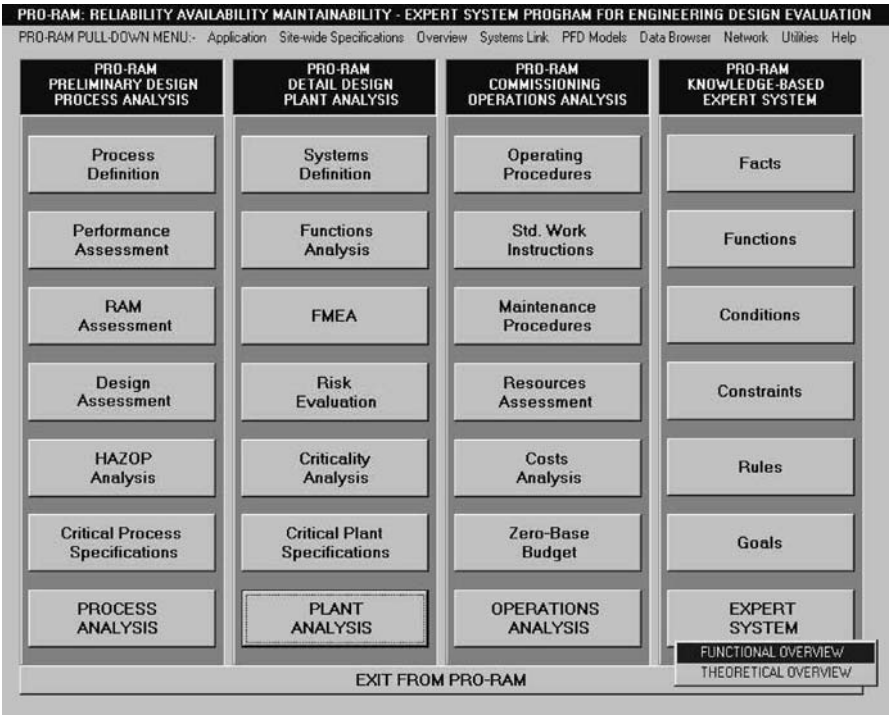


Fig. 5.108 Expert systems functional overview in the AIB blackboard knowledge base

meeting the specific design criteria, and gives information on how they were arrived at, and why.

As previously indicated, the diagnostics inference engine contains diagnostic charts and queries relating to equipment failure characteristics and failure conditions. Figure 5.109 illustrates a user's access to the AIB blackboard's diagnostics inference engine selection menu for assessment of equipment conditions, risk and criticality, as well as operating and maintenance costs and strategies and logistic support.

The first step in diagnostics of *equipment condition* is finding the *failure effect* on a process by determining the impact of an isolated failure on neighbouring and dependent components. This is the basic precursor to establishing a failure modes and effects analysis (FMEA). FMEA is a powerful design tool to analyse engineering systems, and may simply be described as an analysis of each potential *failure mode* in the system and examination of the results or *effects* of such failure modes on the system.

The strength of FMEA is that it can be applied at different systems hierarchy levels. In the specific case illustrated in Fig. 5.110, it is applied to determine the performance characteristics of the gas cleaning *process*, the functional failure probability of its critical *systems*, such as the halide tower, the failure-on-demand probability

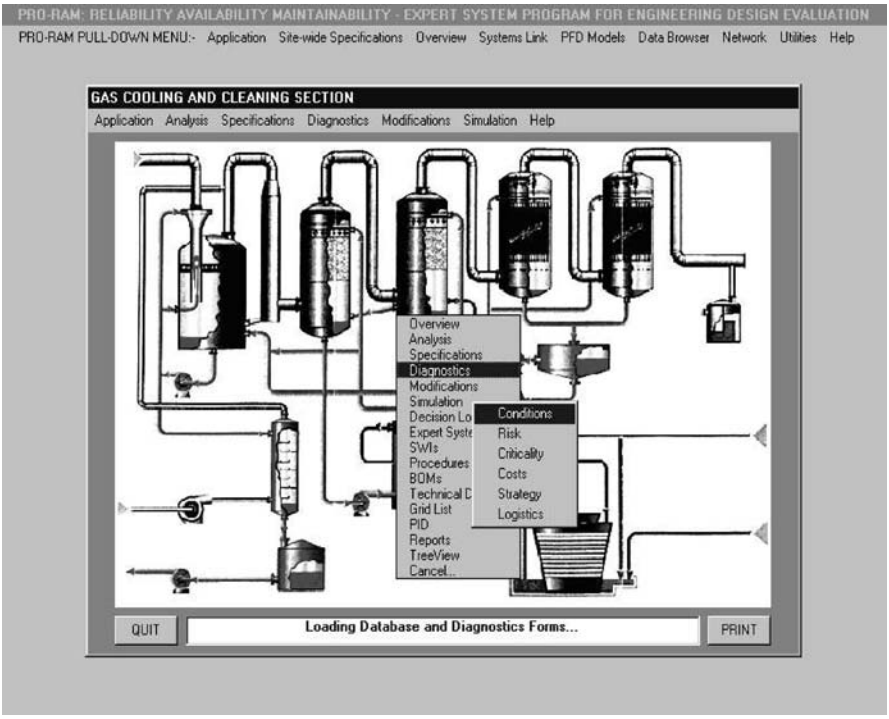


Fig. 5.109 Determining the conditions of a process

of the duty of a single pump *assembly*, namely the halide pump no. 1, down to an evaluation of the failure mechanisms (or failure modes) such as 'failure to open', failure effects and causes associated with a control valve *component*.

By the analysis of individual failure modes, the *effect* of each failure can be determined on the physical *condition* and operational *functionality* of the relevant systems hierarchy level, up to the *consequence* on the overall process. In preparation for establishing an expert system knowledge base pertaining to the diagnostics of *equipment condition*, the FMEA is performed in several steps, which are as follows:

- Identify the relevant hierarchical levels, and define systems and equipment.
- Establish ground rules and assumptions, i.e. operational phases.
- Describe systems and equipment functions and associated functional blocks.
- Identify and describe possible failure modes and their associated effects.
- Determine the effect of each item's failure for every possible failure mode.
- Determine the consequence of each item's failure on system performance.
- Determine the cause of each item's failure for every possible failure mode.

In this way, a knowledge base is built up of the *conditions* and *constraints* relating to failure characteristics and failure conditions.



Fig. 5.110 Determining the failure effect on a process

The advantages of performing a design-level *failure modes and effects analysis (FMEA)* in building up an *equipment condition* knowledge base include: identification of potential design-related failure modes at system/sub-system/component level; identification of important characteristics of a given design; documentation of the rationale for design changes to guide the development of future designs; help in the design requirement objective evaluation; assessment of design alternatives during the preliminary and detail phases of the engineering design process; and establishing priority for design improvement actions during the preliminary design phase. Furthermore, a design-level FMEA is a systematic approach to reduce risk and criticality, when the FMEA is extended to classify each potential failure effect according to its *severity* and *consequence of failure* on the system as a whole, in a systems-level *failure mode effects and criticality analysis (FMECA)*.

The *risk* of *common failure mode* is influenced by stress and time. As both stress and the time-at-stress increase, the risk increases. The point of maximum *common failure mode* risk occurs when both stress and time are at a maximum. However, this risk cannot be evaluated by either reliability analysis or high-stress exposure tests alone, and it becomes necessary to review design criteria conditions to evaluate risk in a *design-level FMEA*. The intention of this type of FMEA is to validate the

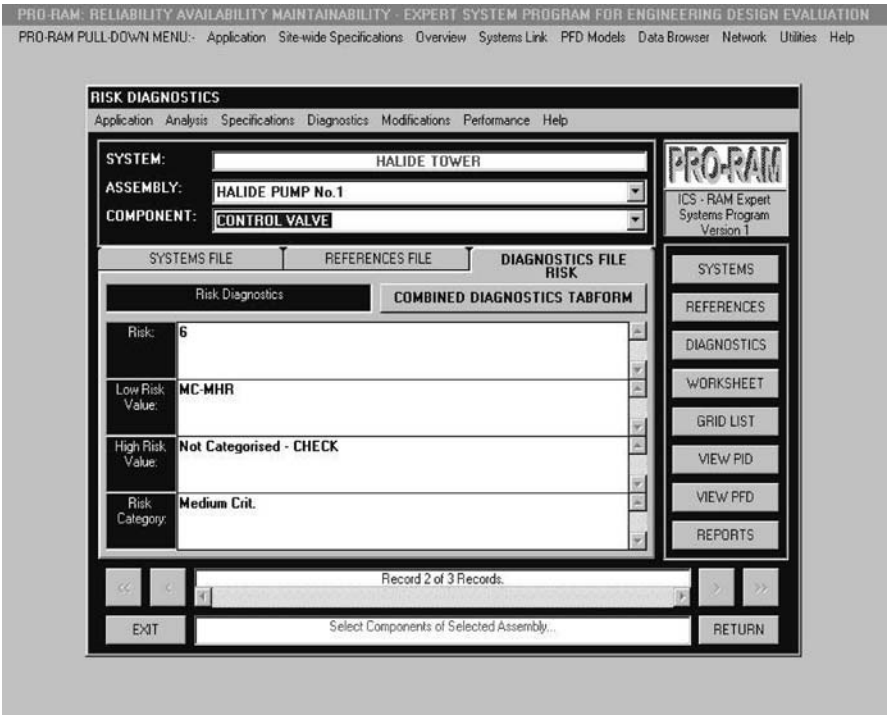


Fig. 5.111 Determining the risk of failure on a process

design parameters chosen for a specified functional performance requirement where the *risk* of *common failure mode* is at a maximum.

The risk evaluation of the *common failure mode* 'fail to open', associated with the example control valve, is illustrated in Fig. 5.111. Several risk categories are shown, specifically: a risk rating (value of 6 out of 10); a risk classification of a low risk value of MC–MHR (medium cost, and medium to high production risk); the grouping of the *common failure mode* risk into a risk category (medium criticality).

Thus, for making an assessment of equipment criticality, particularly at the component level, the priority for a component failure mode is calculated using three factors:

- Failure effect severity.
- Failure consequence likelihood.
- Failure mode occurrence probability.

Figure 5.112 illustrates the further development of an expert system knowledge base pertaining to the diagnostics of *equipment condition*, with the inclusion of determining the *criticality* of failure on a process. The objective of criticality assessment is to prioritise the failure modes identified during the FMEA on the basis of the severity

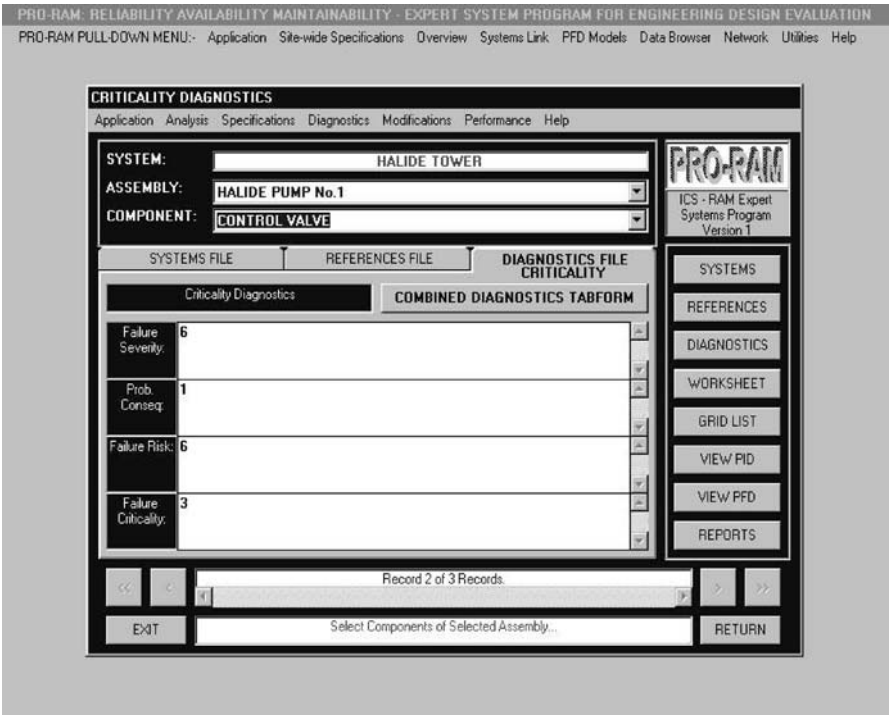


Fig. 5.112 Determining the criticality of consequences of failure

of their effects and consequences, and the likelihood of occurrence, i.e. the risk, as well as the estimated failure rate.

The assessment of *decision logic* in design problem solutions is to determine the required operating, maintenance and logistic strategies based on specific criteria related to system and equipment specifications, such as equipment technical specifications, process functional specifications, operating specifications, equipment function specifications, failure characteristics and failure conditions, equipment fault diagnostics, equipment criticality, equipment performance measures, operating and maintenance tasks, operating procedures, maintenance procedures, process cost models, critical spares, and spares logistic requirements.

Figure 5.113 illustrates decision logic assessment questions for building up a knowledge base of design problems pertaining to process functionality (cf. Fig. 5.96) and to design parameters (cf. Fig. 5.97). These questions serve to define the rules in a *rule-based expert system*. The questions are multiple-choice entries that are typically text and can contain several values. As an aid to the decision logic assessment, the FMECA results of the component under scrutiny are displayed.

Applying AI methodology to engineering design Aside from the use of intelligence in system components, there has been significant progress in its use

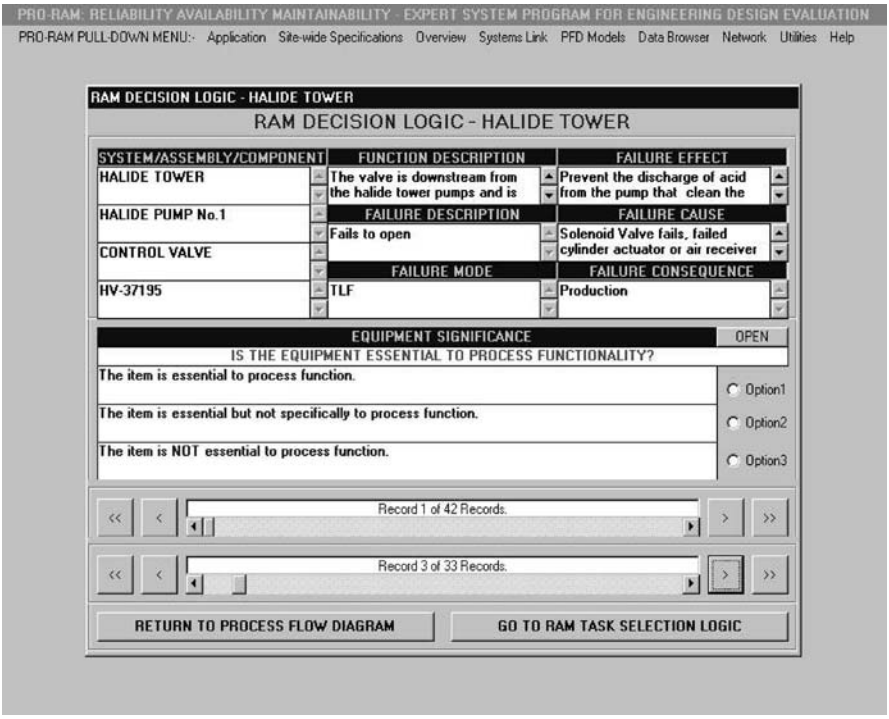


Fig. 5.113 Assessment of design problem decision logic

during design and evaluation of safety-related systems. Intelligent systems provide the safety engineer with valuable knowledge-based tools; the use of expert systems for verification and validation, or for use in FMEA studies, are typical examples. However, some deterministic systems have become so complex and sensitive to trivial input changes that complete analysis becomes a virtual impossibility. AI can support this process by providing experiential analysis of the system outputs, thereby eliminating false logical paths and reducing the amount of analysis required. There is also a move towards analysis whereby only a system's interface performance is assessed against benchmarks provided by an 'acceptable' system.

Figure 5.114 illustrates the options selection menu with 'expert systems' highlighted, which appears by clicking on a selected PEM in the PFD. This accesses the internal AIB blackboard knowledge-based expert systems.

A *facts frame* is a structure that represents a concept in knowledge-based expert systems. It can have any number of *attributes* or *properties* attached to it, some of which can be *relationships.* An attribute may have any number of *values* (i.e. no value, one value, several values, etc.). The types of relationships among frames include hierarchical, classification relations, time precedence, and resource dependent. The importance of being able to represent relations is that a given frame can inherit properties (attributes and/or values) from the frames to which it is related.

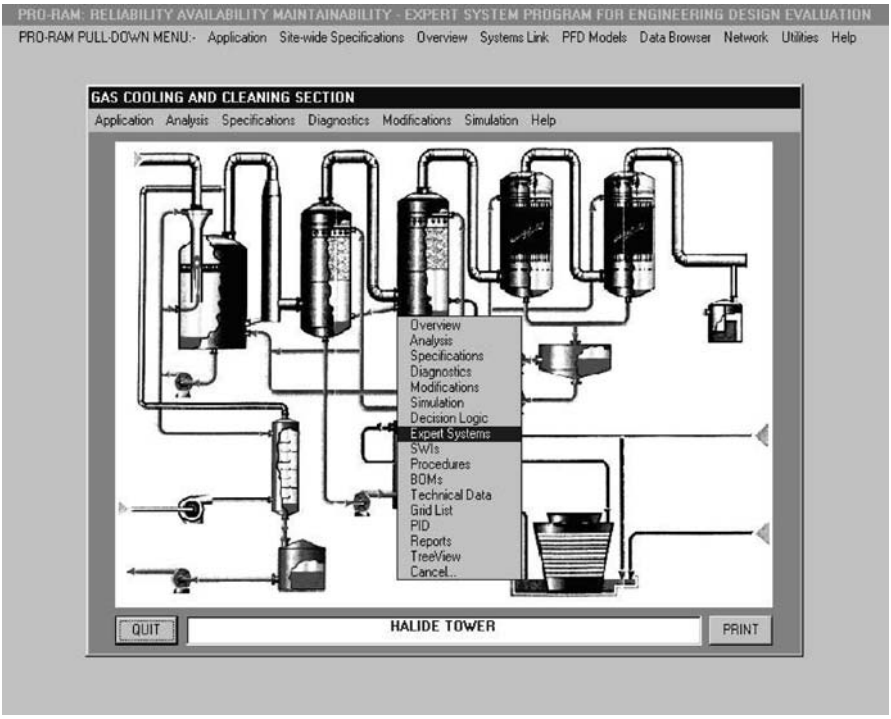


Fig. 5.114 AIB blackboard knowledge-based expert systems

Facts frames are a convenient and natural way to represent descriptive information, related objects, their properties and their relationships. They also represent the information carried in hierarchically structured domains. Relationship frames contain a number of slots representing attributes or relationships with the relevant questions, topic, class, problem statement and solution hypotheses relating to functions, conditions and consequences, as illustrated in Fig. 5.115. These frames represent the knowledge-based information of the design integrity of the equipment.

Frames are also known as *schemata* and *scripts*, and are abstractions of semantic network knowledge representation. As a result, frames are effective in expectation-driven processing, a technique often used in architecture and engineering design, where a knowledge-based expert system looks for expected data, based on context. Frames may inherit information from other frames. Frames are similar to forms that have a title (frame name) and a number of *slots* (frame slots) that accept only predetermined data types. A collection of nodes and links, or slots, together describe an object or event.

A frame is thus a format for expressing declarative knowledge, in which an object is represented by a data structure containing a number of *slots* (representing attributes or relationships of the object), with each slot filled with one or more values (representing specific values or relevant questions of attributes or other objects

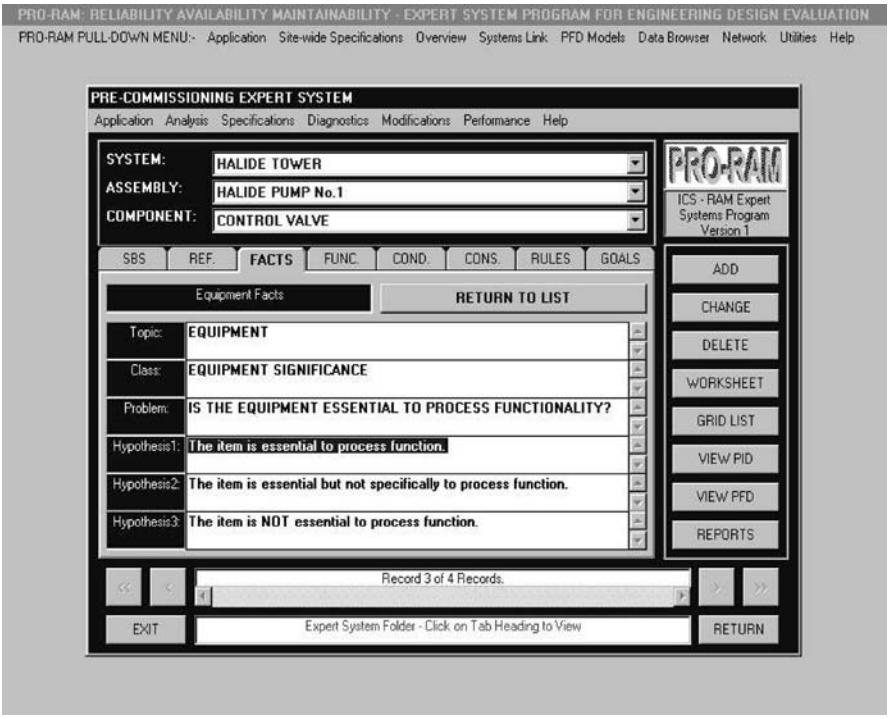


Fig. 5.115 Knowledge base facts frame in the AIB blackboard

that the object is related to). Figure 5.116 illustrates a typical frame *slot* representing attributes or relationships with relevant questions pertaining only to *condition*.

The system/assembly/component selection capability of a system breakdown structure (SBS) relates hierarchical systems data to a particular *hierarchical frame*. Hierarchical frames provide the means to efficiently represent certain types of data that have a hierarchical structure, such as engineering systems. Hierarchical frames allow for complex search criteria with Boolean operators in design optimisation. The data in such a frame can be read or updated by the expert system. These frames provide inheritance that allows a hierarchical set of frames to be created with data in 'parent' frames available to lower-level frames. The use of hierarchical frames provides a means for the AIB blackboard to manage hierarchically related data that are portable and maintainable in multiple expert systems.

Figure 5.117 illustrates the system breakdown structure (SBS) tab as part of a set of tabs (references, facts, functions, conditions, consequences, rules and goals) that contain various instructions in accessing data from the expert system knowledge database for application in an *expert system user interface*.

The *AIB blackboard* provides flexible use of *multiple expert systems*, and other knowledge source applications, to store and retrieve data for use by multi-disciplinary groups of design engineers in a collaborative design environment. The data are

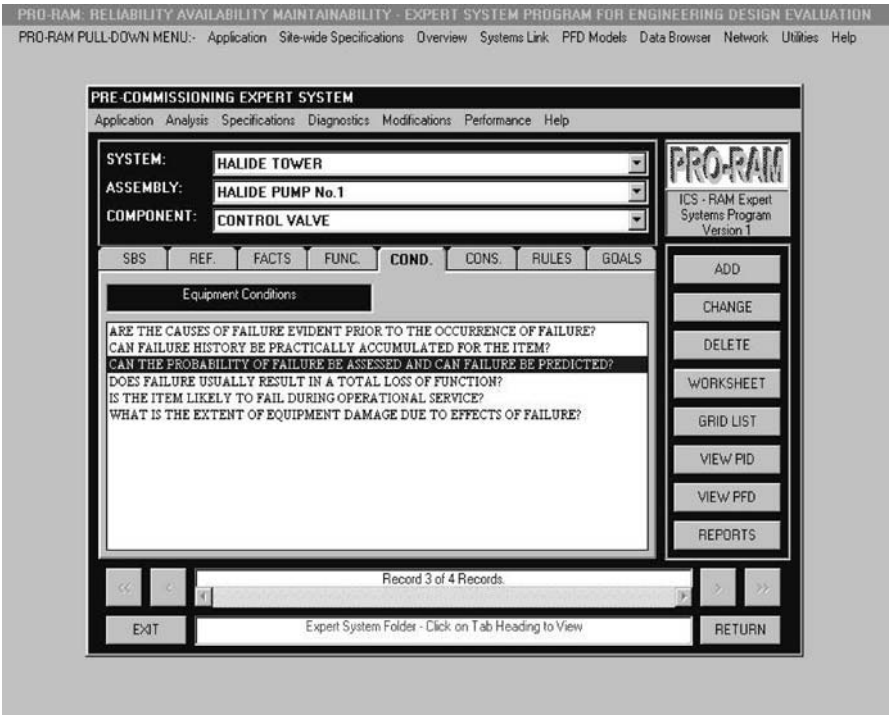


Fig. 5.116 Knowledge base conditions frame slot

shared among multiple knowledge-based expert systems as well as other knowledge sources. A set of blackboard commands enables designers to access specific frames to read or write design data to the blackboard. The blackboard files can be jointly read or created by the knowledge-based expert systems that automatically identify inappropriate design data or conflicting design specifications.

Figure 5.118 illustrates the 'goals' option tab of the imbedded ExSys© Expert System (ExSys 2000). Goals are the *design criteria* among which the expert systems will decide. An expert system is required to find solutions to a design problem subject to design criteria. A goal may be assigned a *confidence value* to determine its relative likelihood. Goals can be used only in the THEN part of trees, which are considered later.

Factors are text or numeric data items that are used to define the rules in a rule-based expert system (or the nodes of a decision tree). There are two types of factors: 'questions' and 'variables'.

Questions are multiple-choice lists that are typically text and can contain several values. A question condition is a statement in the rule (or tree) made up of the starting question text and several associated choices. Questions can be used in the IF part of a rule to *test* a value, or in the THEN part to *assign* a value.

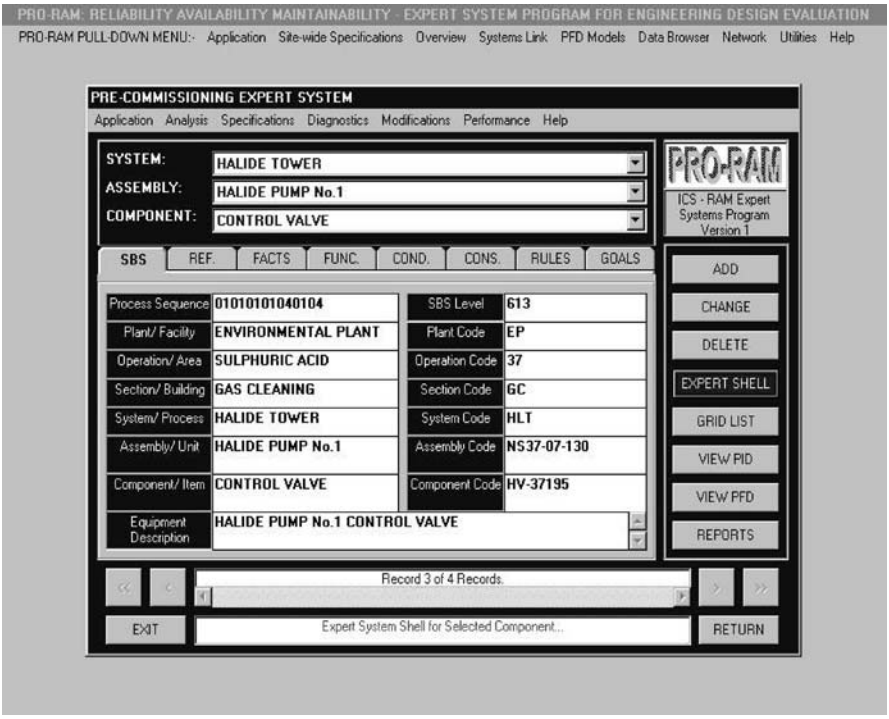


Fig. 5.117 Knowledge base hierarchical data frame

Figure 5.119 illustrates the 'questions' option tab of the imbedded ExSys© Expert System (ExSys 2000). In the illustration, one question relates to temperature, which refers to the expert system goal and is a design criteria constraint. Another question relates to a pressure constraint, where both constraints need to be considered concurrently, as the one impacts upon the other in a particular system design.

Figure 5.120 illustrates a *multiple-choice question editor* for application in the rule-based expert system. The multiple-choice question editor is used in establishing lists for questions relating to both temperature and pressure for application in the same rule-based expert system. The values of these variables are divided into ranges defined by the logical break points in the decision-making process (i.e. too low, within or exceeding specification, too high or critical).

Variables are both numeric or string variables, including expressions, parenthesis, Boolean operators, trig functions and exponential functions. A numeric variable may have any value between its upper and lower bounds. For the purposes of defining IF–THEN expert system rules, the value of the variable is divided into ranges defined by the logical break points in the decision-making process. For example, an IF part test expression might be the rule: IF (([X]<[Z]) AND (SIN([X]/2)>0.4)). The more standard MIN, MAX conditional assignment operators are also supported, as well as an approximately equal operator to handle round-off error problems.

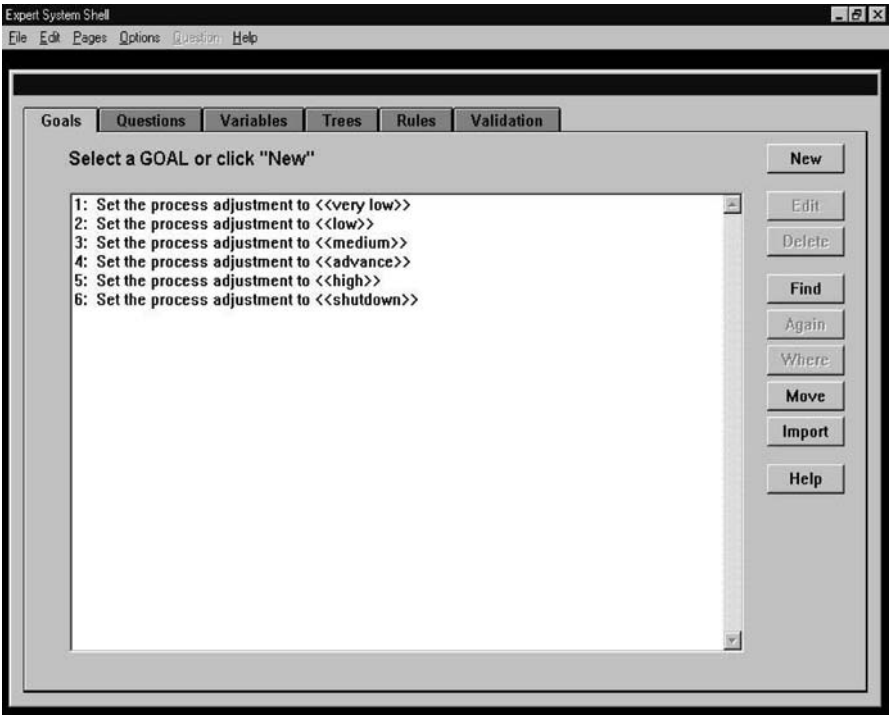


Fig. 5.118 The Expert System blackboard and goals

A *tree*, or *branched decision tree*, represents all or a portion of the decision-making instructions for input into a particular design scenario. Tree representation can be used for any design problem that involves a selection from among a definable group of goals (design criteria), where the decision is based on logical steps that can be described as a set of tree diagrams. The trees can involve relative probabilities of a goal being correct. Trees can also be used to derive data needed by other trees or rules. Individual rules are added to represent specific facts that cannot be represented as trees (usually rules requiring an ELSE part, or specific facts that are not part of an overall structure of information).

Figure 5.121 illustrates the 'trees' option tab of the imbedded ExSys© Expert System (ExSys 2000). The branched decision tree in the illustration relates to a process adjustment for the design criteria constraint of temperature for establishing decision-making instructions as input into a particular design scenario.

A *branched decision tree* is made up of *nodes* that represent decision branch points, and those that are assignments of value, as illustrated in Fig. 5.122. These correspond to IF and THEN conditions in a rule. The IF node has two or more values that are joined together in a block. The node values can be multiple-choice text items, ranges of a numeric variable or true/false tests of a mathematical expression. THEN nodes have a single value and assign a value to the goal of the expert

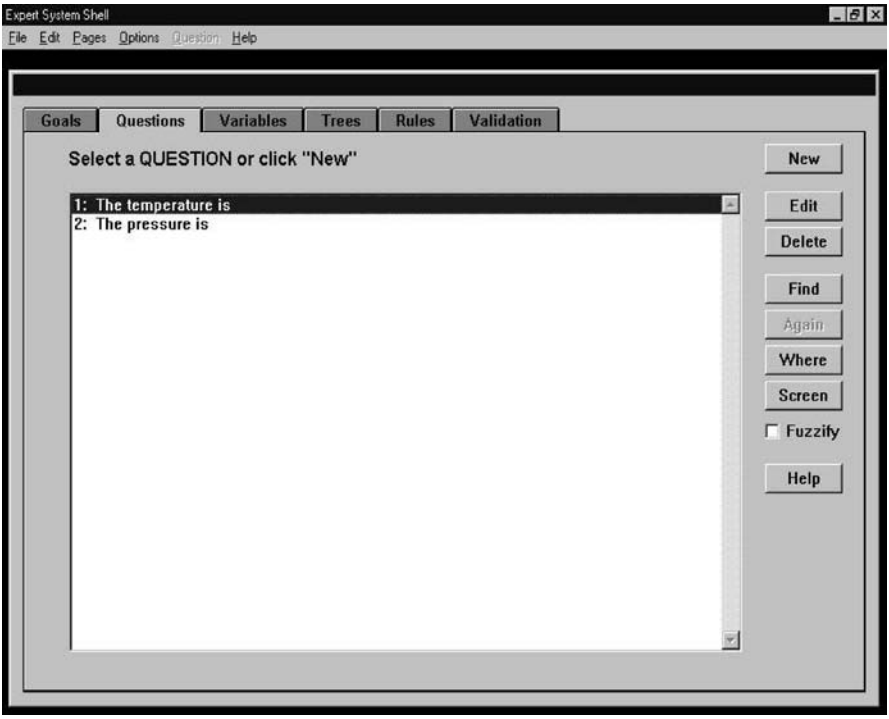


Fig. 5.119 Expert System questions factor—temperature

system, assign text or numeric data, or annotate the tree. Trees can assign values in their THEN part. If other tree nodes require these choices, the appropriate rules will automatically be called through *backward chaining*.

The ability to add nodes or automatically expand the tree, to consider all possible combinations of input, allows rules to be built very rapidly. The designer is prompted to consider all possible cases, which guarantees system completeness. In most applications there would be multiple trees, each representing a different aspect of the decision-making process.

Knowledge-based expert systems deal with knowledge, rather than data, and the files they use are often referred to as *knowledge bases*. This knowledge is represented as *rules*, as illustrated in Fig. 5.123. A rule is made up of a list of IF conditions (normal semantic sentences or algebraic expressions that can be tested to be TRUE or FALSE), and a list of THEN conditions (also semantic sentences or algebraic expressions) or statements about the probability of whether a particular value is the appropriate solution to the design problem. If the expert system determines that all IF conditions in a rule are true, it adds the rule's THEN conditions to what it knows to be true.

The ability of an expert system to *derive* information, rather than prompting the user, enables the expert system to combine many small pieces of knowledge to arrive

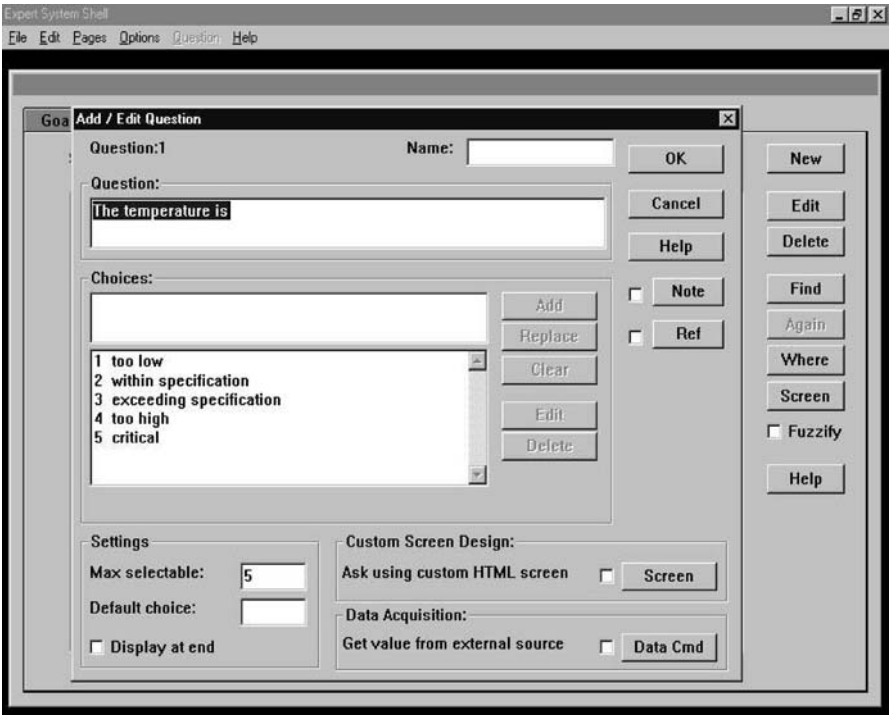


Fig. 5.120 Expert System multiple-choice question editor

at logical conclusions about complex problems. In this way, the expert system can manage the rules in a knowledge base from the point of view of conflicting meanings or values.

Although the use of decision trees are an ideal way to rapidly develop most systems, the logic of some complex systems cannot be described as trees. These systems require that individual *rules* be defined. A *rule editor* is provided for this ability. The expert system's rule editor is a powerful and flexible tool for knowledge-based expert system development, and enables the designer to rapidly write rules using the same data elements as the trees. As each rule is input, it is compiled to the knowledge base. This means that the rule editor has access to the logic of the rules already entered. Thus, if a rule is potentially in conflict with an existing rule, the editor will immediately highlight the conflict and give the designer the opportunity to correct it.

Figure 5.124 illustrates the rule editor of the imbedded ExSys© Expert System (ExSys 2000). Rules entered in an editor window are divided into three main parts: an IF part, a THEN part, an optional ELSE part, with an optional NOTE, and an optional REFERENCE.

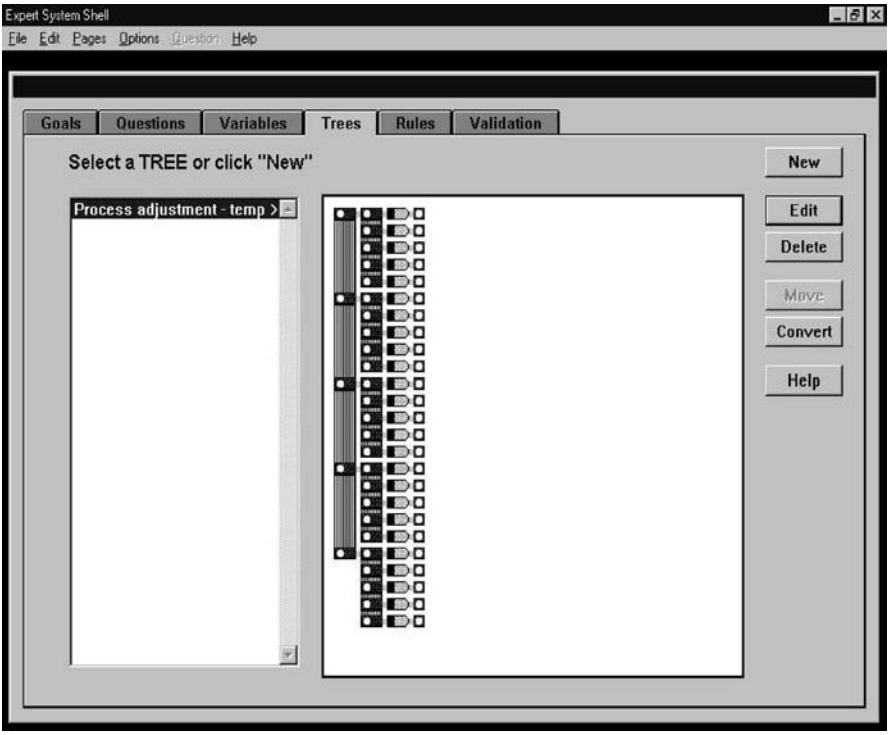


Fig. 5.121 Expert System branched decision tree

The basic *structure of a rule* has the following format:

IF – conditions
THEN – conditions and goals
ELSE – conditions and goals.

The IF part is simply a series of tests, expressed as semantic sentences or algebraic expressions. The IF conditions are tested against the data provided by the knowledge base, information that can be derived from other rules, or data obtained from other knowledge sources. In the IF part, the tests can be combined with either AND or OR. Boolean operators can also be used to build complex logical tests.

The THEN part can contain conditions similar to those of the IF part. However, in the THEN part, they are not tests but statements of fact. In the IF part, a statement would be a test that might be true or false. The same statement in the THEN part would be considered to be a valid fact, if the IF conditions in the rule were true. When the IF conditions in a rule are determined to be true, the expert system assumes the THEN part is true and adds any facts in the THEN conditions to what it knows. The THEN part can also contain the possible goals that the expert system will decide among, along with their assigned probability values. The expert system keeps track of the value each goal receives, and calculates a final confidence value

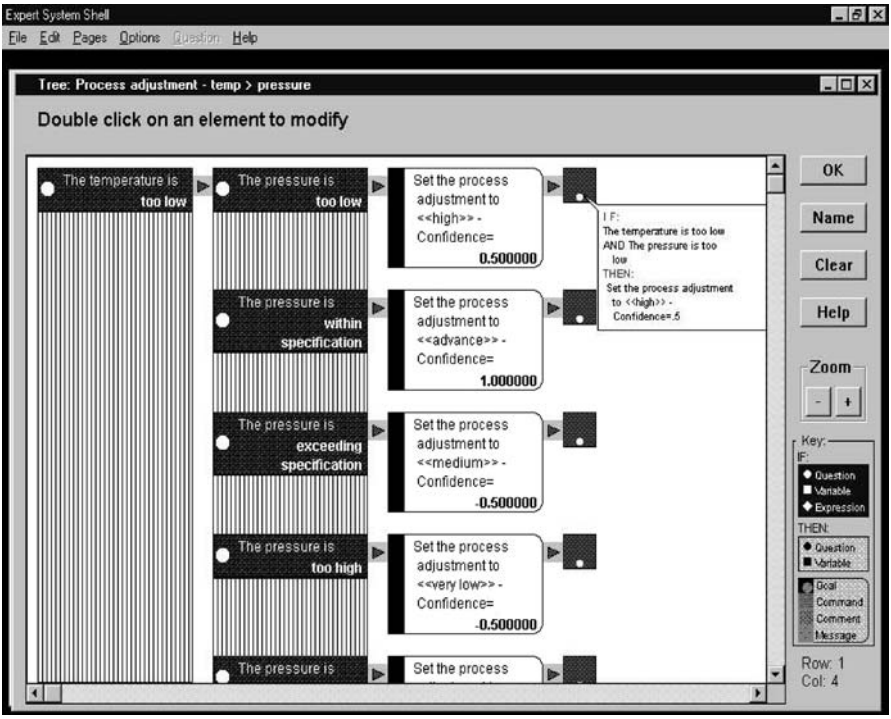


Fig. 5.122 Expert System branched decision tree: nodes

for each of the choices. The THEN conditions may also include statements that assign a value to a numeric or string variable. This allows values to be calculated during a run and displayed at the end.

The ELSE part is the same as the THEN part and is applied if any of the IF conditions are FALSE. The ELSE part is optional and not needed in most rules. (Rules built in tree structures have only IF and THEN parts.)

In some cases, it is desirable to add a NOTE to a rule to provide some special information to the knowledge base. If there is a NOTE added, it will be displayed with the rule. The NOTES from rules that fired (i.e. are activated) can also be applied as information output, using the report generator. The expert system knowledge base may also include a REFERENCE for a rule. This is intended to assist in finding the source of the knowledge contained within a rule, or for more information relating to the rule. As with the NOTE, the REFERENCE is optional and only for containing information. It has no effect on the running of the program.

The difference between the NOTE and the REFERENCE is that the NOTE is displayed whenever the rule is displayed. The REFERENCE is displayed only if it is requested. The REFERENCES from rules that fired can also be applied as output using the report generator. Both NOTE and REFERENCE elements can contain

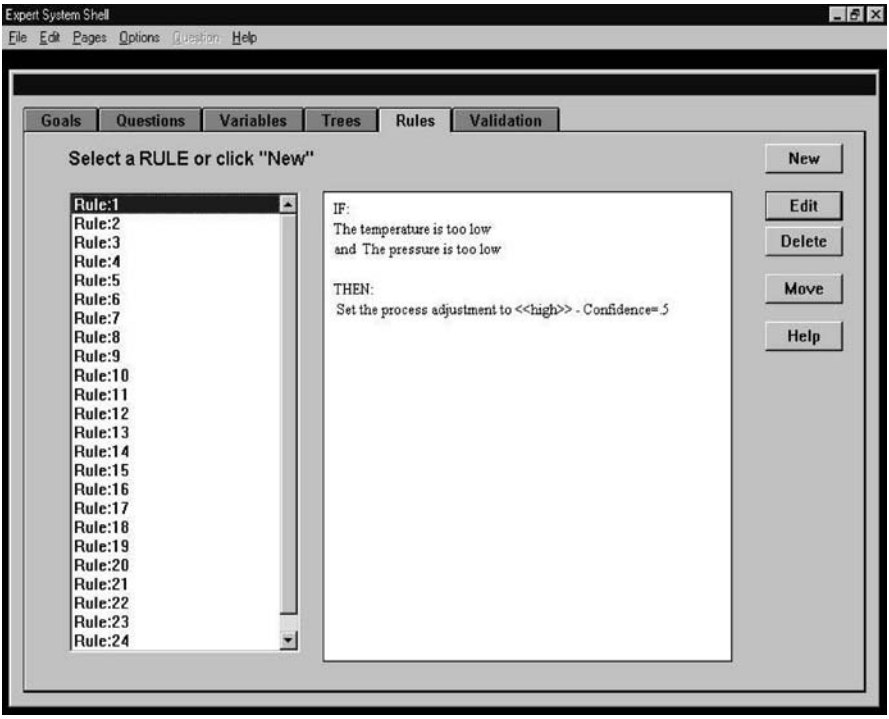


Fig. 5.123 Expert System rules of the knowledge base

links to other blackboards, which is useful to display different parts of a complex design.

Backward chaining is a term used to describe running the rules in a goal-driven manner. In backward chaining, if an item of information is needed, the expert system will automatically check all of the rules to see if there is a rule that could provide the needed information. The system will then 'chain' to this new rule before completing the first rule. This new rule may require information that can be found in yet another rule. The expert system will then again automatically test this new rule. The logical reasoning of why the information is needed goes backwards through the chain of rules. The called rules can be anywhere in the expert system. It is not necessary to specify which rules apply to which information. Backward chaining simplifies the development of the expert system. Each rule can simply state an individual fact. Unlike some expert system models, the relationships between rules do not have to be explicitly assigned in the blackboard. Expert systems incorporated in the blackboard will automatically find the relevant rules and use these. In a backward chaining system, the rules can be in any order. As new facts are added to the design, rules are simply added and the expert system will automatically determine when and how to use the new items of information.

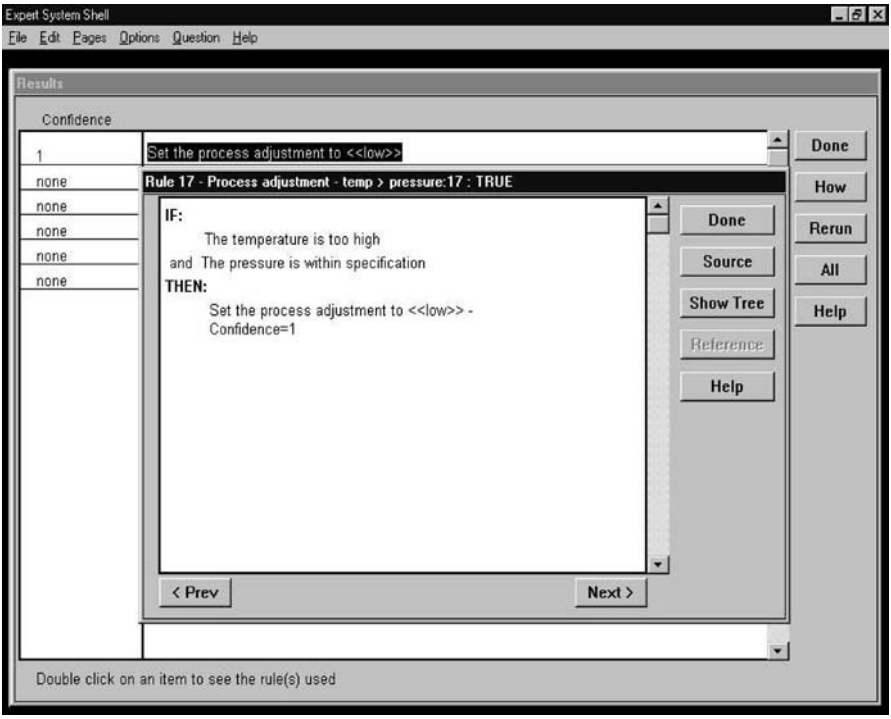


Fig. 5.124 Expert System rule editor

Forward chaining is a data-driven method of running the rules (unlike the goal-driven approach of backward chaining), and is an alternative to backward chaining. In backward chaining, there is always a goal to be satisfied and a specific reason why rules are tested. In pure forward chaining, rules are simply tested in the order they occur based on available data. If information is needed, other rules are not invoked—instead, the designer is asked for the information. Consequently, forward chaining systems are more dependent on rule order. However, since time is not spent determining if information can be derived from other rules, forward chaining is much faster. The blackboard expert system also provides a hybrid between backward and forward chaining, where the basic approach is data-driven but information needed by rules is derived through backward chaining. Another technique is to divide an expert system into subsets of rules and run some in forward chaining and some in backward chaining with procedural command language.

Testing and validating a knowledge-based expert system must be a major part of any expert system development project. It is important to make sure that end users will get valid answers to any input. The automatic validation function greatly simplifies and automates this process. The expert system automatically tests the design application, unattended, for a variety of errors. There are two methods of validation testing, namely systematic and random testing. Systematic testing allows

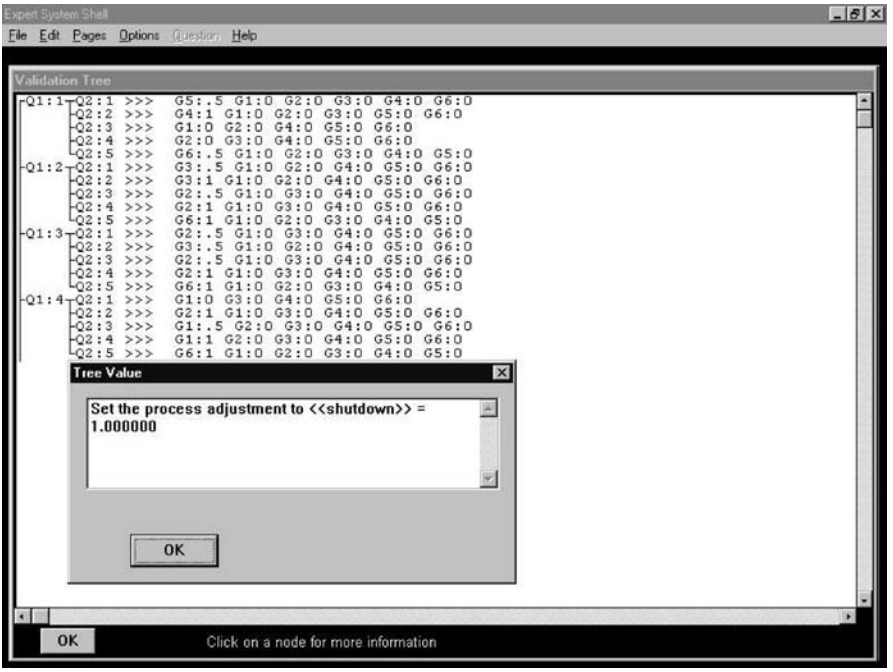


Fig. 5.125 Testing and validating Expert System rules

all possible combinations of input to be tested for a variety of possible errors. If the expert system is very large and systematic testing of the entire system would take too long, testing of portions of the system, or random testing of the entire system, can be performed. The blackboard system has built-in functions to automatically validate a particular design application, and to check for a variety of common errors.

Figure 5.125 illustrates the validation test facility of the imbedded ExSys© Expert System (ExSys 2000). A validation test run is illustrated of the example HIPS system with specific temperature and pressure design constraints.

Confidence methods for managing uncertain data The AIB blackboard expert system provides several ways to manage uncertain data, and the confidence or probability factors within each expert system. The different systems are designed to provide a range from simple and intuitive confidence systems, to complex methods of assessing and evaluating confidence. These confidence methods are:

YES/NO system If the system does not require any estimate of probability, the YES/NO (0/1) system is the easiest to apply. This confidence system is very easy to use, since the first rule that fires for a choice sets the value to 1 for 'yes', or 0 for 'no'. No intermediate values are assigned. This confidence system is good for selecting choices from a list, an automated questionnaire, or other systems where the choices used by such systems are 'yes' or 'no'.

0–10 system The 0–10 system provides confidence values on a scale of 0–10. This is often quite compatible with the intuitive knowledge used in the development of an expert system. A value of 0 locks the value for the goal at 0 (no—not possible) and a value of 10 locks the value for the goal at 10 (yes—definitely selected). Confidence values between 1 and 9 are averaged to give a *relative likelihood*.

This system can positively select or reject a goal (with a value of 10 or 0) but can also allow intermediate values to indicate goals that may also be appropriate. When obtaining a designer's intuitive recommendation in an event, a 0–10 scale is often easy to use. Unless valid statistical data are available, precision higher than 0–10 is difficult to obtain from intuitive knowledge, especially for conceptual design. Despite its simplistic calculations, the 0–10 system is quite suitable for many expert systems and has been used to build thousands of real-world applications.

–100 to 100 system Values can be assigned to goals in the range of -100 to 100. This provides greater resolution than the 0–10 system. However, there is no value that locks the value at 'yes' or 'no', and values of 0, 100 and -100 are treated like any other value. There are three methods of combining the confidence values—average, dependent probabilities, or independent probabilities. The system is effective if the nature of the problem requires that the confidence factors be combined as dependent or independent probabilities.

Fuzzy logic is a very powerful technique that enables the expert systems to manipulate imprecise data (i.e. the temperature is too high) and more closely reflect the real world. In the fuzzy logic confidence mode, fuzzy membership functions are defined that assign confidence to items based on the value of a variable. These confidence values are propagated through the rules to the confidence assigned to the goals. Specific values can be *defuzzified* out of the results, to have the expert system give precise recommendations.

Figure 5.126 illustrates the application of fuzzy logic for managing uncertainty concerning the design constraint of temperature, and can similarly be done for the pressure design constraint. The fuzzy membership functions are represented by a triangular distribution mapping (t-norm), established by input of membership functions against specific confidence levels. The resulting rules developed with fuzzy logic inference are similar in construct to Fig. 5.124, except with $<$ or $\ll$ notation.

Plant analysis, with specific reference to the integrity of engineering design, focuses on equipment functional failure, their causes and effects, and the overall consequences that affect safety, operations, quality and the environment. It includes the identification of critical equipment with regard to safety, risk, operations downtime, product quality and environmental impact, as well as costs of downtime. The outcome of plant analysis determines maintenance procedures, plant isolation procedures (with the establishment of statutory requirements), plant shutdown procedures (shutdown and start-up), standard work instructions, maintenance and operating resource requirements, and logistical spares requirements, for the effective care of plant and equipment to ensure safety, operational performance, production output, product quality and environmental protection. Figure 5.127 illustrates the plant analysis functional overview option in the AIB blackboard. It also provides

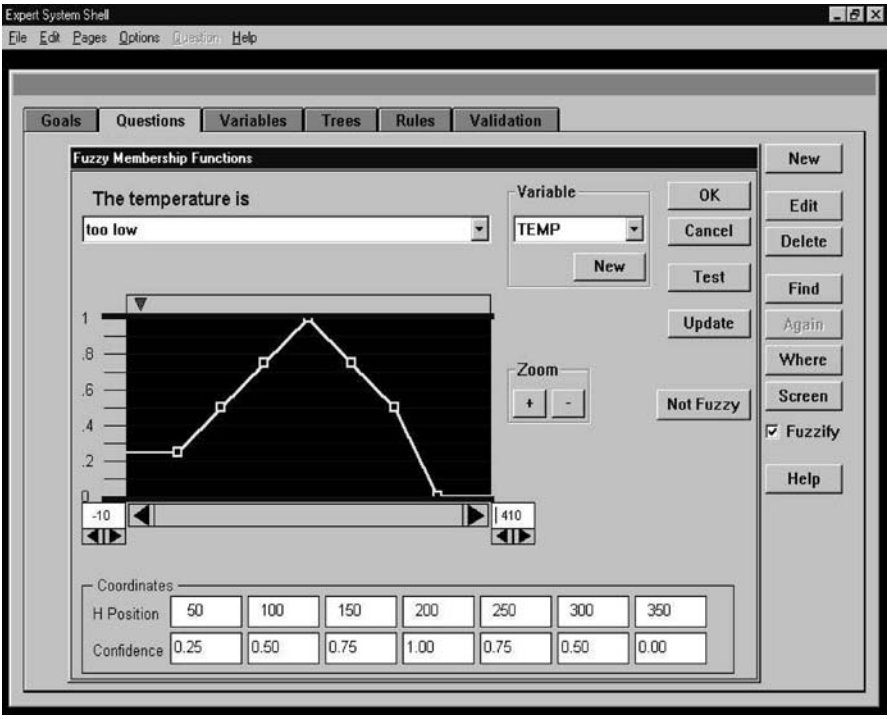


Fig. 5.126 Fuzzy logic for managing uncertain data

a theoretical overview of reliability, availability, maintainability and safety in engineering design—the methodology presented in this handbook.

Plant analysis in the AIB blackboard is the working memory of the knowledge-based expert systems, consisting of a global database of facts relating to the integrity of engineering design, which are used for establishing *automated continual design reviews*. The basic aims of automated continual design reviews are to automatically assess system requirements and allocations to ensure that the design specifications are complete; to automatically compare the design output against design specifications; to automatically present the risks associated with a collaborative and continuous design effort; and to continually allow for decision-making in selecting the most suitable design amongst the current design solutions.

Figures 5.128 and 5.129 illustrate the typical AIB blackboard format of an automated continual design review. Figure 5.128 shows the blackboard systems hierarchy navigation and selection format whereby critical components can be viewed with regard to their systems relationships.

Figure 5.129 shows a typical criticality assessment of a component, based on condition and performance obtained from an FMECA analysis.

The artificial intelligence blackboard model—overview Artificial intelligence-based strategies for decision-making and, in particular, for decisions concerning the

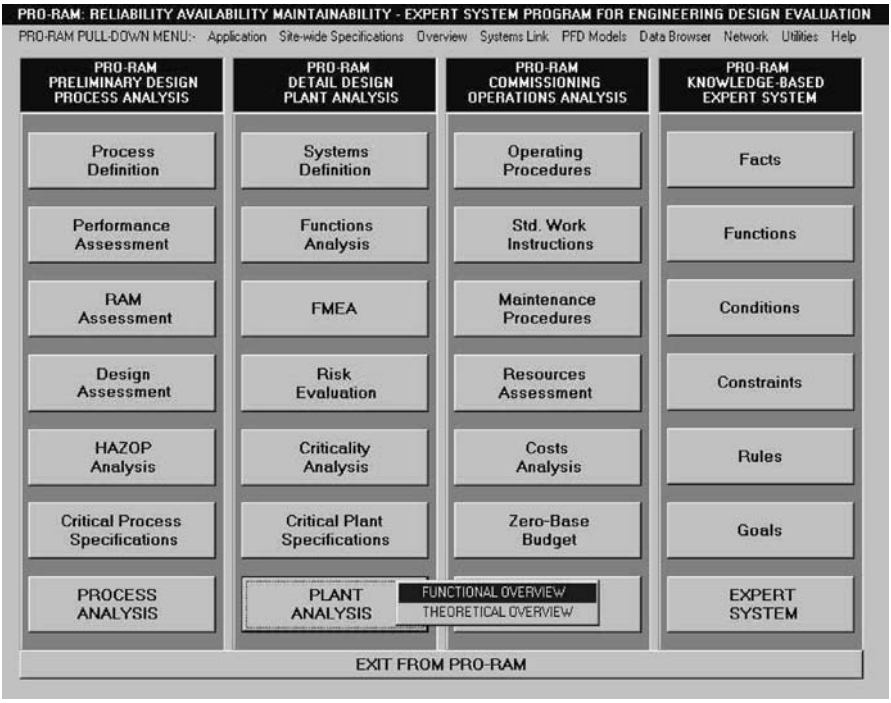


Fig. 5.127 AIB blackboard model with plant analysis overview option

integrity of engineering design are centred around three approaches termed *deterministic knowledge*, *probabilistic knowledge* and *possibilistic knowledge*.

Deterministic knowledge, in engineering design integrity formulation, is based on a well-defined systems structure and definition of the operational and physical functions of equipment, the usefulness of which depends on the ability to relate the information specifically to failure conditions (or failure modes) in identifying problems of equipment failure consequences.

Probabilistic knowledge is gained mainly from a statistical analysis of the probable occurrences of events, such as component failures, in order to predict the expected occurrence of these events in the future to be able to design-out problems or to implement some form of preventive action.

Possibilistic knowledge focuses primarily on imprecision or uncertainty that is intrinsic to equipment degradation. Imprecision here is meant to express a sense of vagueness, rather than the lack of any knowledge at all about predicted equipment condition, particularly its physical condition. In other words, possibilistic knowledge concerns the concept of ‘fuzziness’, and not ‘randomness’.

The application of *fuzzy logic expert systems* focuses on the use of expert systems technology and fuzzy logic to achieve intelligent computer automated methodology to determine the integrity of engineering design. The most important impact areas of expert systems on the integrity of engineering design are:

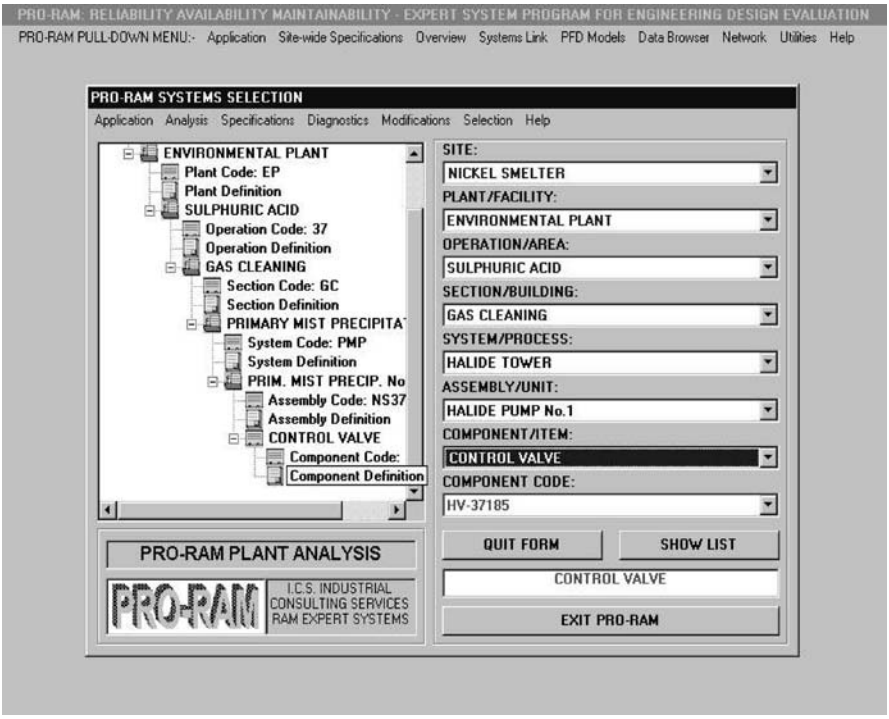


Fig. 5.128 Automated continual design review: component SBS

- automatic checking of design constraints that affect the design's integrity, allowing for alternatives to be considered in a collaborative design environment;
- automation of complex tasks and activities for determining design integrity where expertise is specialised and technical;
- strategies for searching in the space of alternative designs, and monitoring of progress towards the targets of achieving the required design integrity;
- integration of diverse knowledge sources in an AIB blackboard system, with expertise applied concurrently to the problem of ensuring design integrity;
- provision of intelligent computer automated methodology for determining the integrity of engineering design through automated continual design reviews.

5.4.2 Evaluation of Modelling Results

As previously indicated, blackboard systems consist mainly of a set of *knowledge sources* and a blackboard *data structure*. A blackboard knowledge source is a highly specialised, highly independent process that takes inputs from the blackboard data structure, performs a computation, and places the results of the computation back in

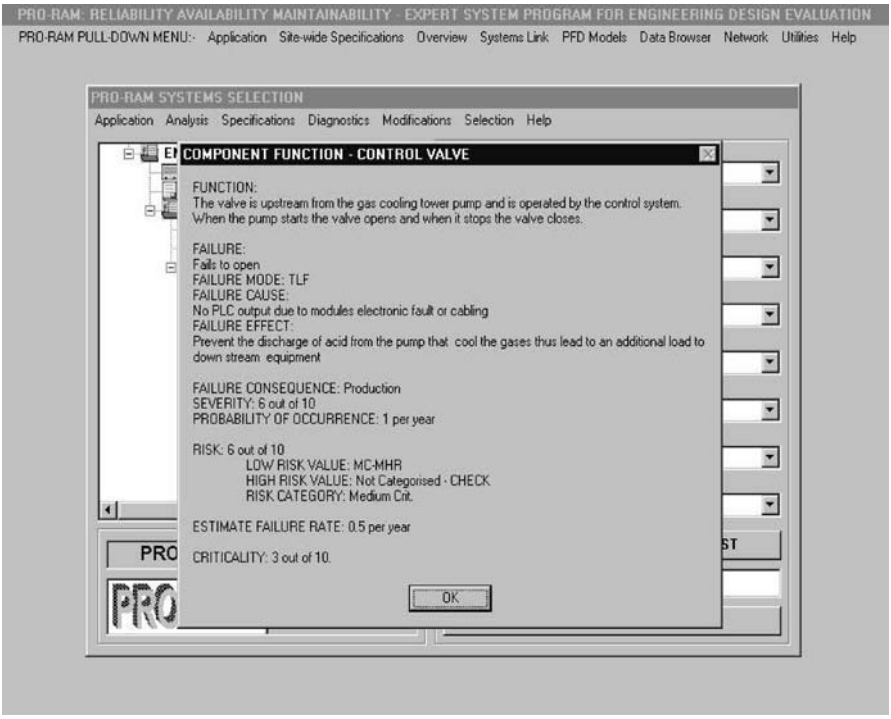


Fig. 5.129 Automated continual design review: component criticality

the blackboard data structure. This blackboard data structure is a centralised global data structure partitioned in a hierarchical manner and used to represent the problem domain (in this case, the engineering design problem), and acts as a shared memory visible to all of the knowledge sources to allow intercommunication between the knowledge sources. The blackboard data structure contains shared blackboard data objects and can be accessed by all of the knowledge sources. This design allows for an opportunistic control strategy that enables a knowledge source to contribute towards the solution of the current problem without knowing which of the other knowledge sources will use the information.

Blackboard systems are a natural progression of *expert systems* into a more powerful problem-solving technique. They generally provide a way for several highly specialised knowledge sources to cooperate to solve larger and more complex problems. Due to the hierarchical structure of the blackboard, each data object on the blackboard will usually have only one knowledge source that can update it. Although these knowledge sources are often referred to as 'experts', knowledge sources are not restricted to expert systems such as the ExSys© Expert System (ExSys 2000) or other AI systems, and include the ability to add conventionally coded software such as the artificial intelligence-based (AIB) model, to cooperate in solving problems.

Many knowledge sources are numeric or algorithmic in nature (i.e. the AIB blackboard knowledge source for artificial neural network (ANN) computation that is specifically applied for processing time-varying information, such as non-linear dynamic modelling, time series prediction, adaptive control, etc. of various engineering design problems). The use of multiple, independent knowledge sources allows each knowledge source to use the data representation scheme and problem-solving strategy that best suit the specific purpose of that knowledge source. These specialised knowledge sources are thus easier to develop and can be hosted on distributed hardware.

The use of opportunistic problem-solving and highly specialised knowledge sources allows a set of distributed knowledge sources to cooperate concurrently to solve large, complex design problems. However, blackboard systems are not easily developed, especially where a high degree of concurrent knowledge source execution must be achieved while maintaining knowledge consistency on the blackboard. In general, blackboard systems have not attained their apparent potential, because there are no established tools or methods to analyse their performance.

The lack of a coherent set of performance analysis tools has in many cases resulted in the revision of a poorly designed system to be ignored once the system had been implemented. This lack of the appropriate performance analysis tools for evaluating blackboard system design is one of the reasons why incorporating concurrency into the blackboard problem-solving model has not generally been successful. Consequently, a method for the validation of blackboard system design has been developed (McManus 1991). This method has been applied to the AIB blackboard system for determining the integrity of process engineering design.

Knowledge source *connectivity analysis* is a method for evaluating blackboard system performance using a *formalised model* for blackboard systems design. A description of the blackboard data structure, the function computed by each knowledge source, and the knowledge source's input and output variables are sufficient to create a formalised model of a blackboard system (McManus 1992). Connectivity analysis determines the data transfers between the knowledge sources and data migration across the blackboard.

The attributes of *specialisation*, *serialisation* and *interdependence* are evaluated for each knowledge source. This technique allows for the evaluation of a blackboard design specification before the blackboard system is developed. This also allows the designer to address knowledge source connectivity problems, feedback loops and interdependence problems as a part of the initial design process. Knowledge source connectivity analysis measures the *output set overlap*, *functional connectivity*, and *output to input connectivity* between pairs of knowledge sources. Output set overlap is a measure of the *specialisation* of pairs of knowledge sources, whereas functional connectivity between pairs of knowledge sources is a measure of their *serialisation*, and output to input connectivity is a measure of their *interdependence*.

a) The Formalised Model for Blackboard Systems Design

Knowledge source connectivity analysis requires a specification of the system developed using a formalised model for blackboard systems (McManus 1992). Blackboard systems can be modelled as a blackboard data structure containing shared *blackboard data objects*, and a set of cooperating knowledge sources that can access all of the blackboard data objects. These knowledge sources are processes that take inputs from the blackboard, perform some computation, then place the results back on the blackboard for other design teams in a collaborative design environment.

Blackboard data structure A blackboard data structure is a global data structure consisting of a set of blackboard data objects, $\{d_1, \ldots, d_j\}$, used to represent the problem domain.

Blackboard data object Each blackboard data object is a predefined data object type with a point value or a range of values. A blackboard data object, d_j, is thus an object that has a single value or multiple values.

Knowledge source A knowledge source, ks_j, of a set of knowledge sources, $\beta = \{ks_1, \ldots, ks_j\}$, consists of the following:

- a set of input variables, IV $= \{iv_1, \ldots, iv_n\}$,
- a set of input conditions, IC $= \{ic_1, \ldots, ic_n\}$,
- a set of output variables, OV $= \{ov_1, \ldots, ov_m\}$,
- a description of the computation delivered by the knowledge source,
- a set of preconditions, PR $= \{pr_1, \ldots, pr_k\}$,
- a set of post-conditions, PT $= \{pt_1, \ldots, pt_k\}$ and
- an input queue, IQ.

A knowledge source's input conditions are a set of Boolean variables used to notify a knowledge source when one of its input variables has been updated. The preconditions are a set of Boolean functions that all must be TRUE for a knowledge source to be activated, and the post-conditions are a set of Boolean functions that all must be TRUE for a knowledge source to post the result of its computation to the blackboard. If all of a knowledge source's activation conditions are met while it is executing, the input queue stores the knowledge source's input variables.

There are two classes of input variables pertaining to knowledge sources: *explicit input variables* and *generic input variables*. An explicit input variable specifies a single, unique blackboard data object that is used as the input variable to a knowledge source. A knowledge source can use only the blackboard data object specified by the explicit input variable as a valid input. A generic input variable specifies a class or type of blackboard data object that can be used as the input variable to the knowledge source. The knowledge source can accept an instance of a blackboard data object of the specified class as an input variable. The use of generic input variables allows development of knowledge sources that function on a class of blackboard data objects.

Knowledge sources can be classified by their input variables:

- *Explicit knowledge sources* have only explicit input variables;
- *Mixed knowledge sources* have both explicit and generic input variables;
- *Generic knowledge sources* have only generic input variables.

Blackboard system A blackboard system is used to allow intercommunication of knowledge sources, and acts as a shared memory that is visible to all of the knowledge sources. A blackboard system, B, is a tuple $\langle X, P, \beta, \mathbf{Id}, \theta \rangle$, where:

- X is a set of blackboard data objects, $X = \{d_1, \ldots, d_i\}$;
- P is the set of blackboard data object states, $P = V_1 \cdot V_2 \cdot \ldots \cdot V_i$, where V_i is a set of all valid values for blackboard data object d_i;
- β is the set of knowledge sources, $\beta = \{ks_1, \ldots, ks_j\}$;
- each knowledge source's domain is a subset of P, and its range is a subset of P;
- **Id** is an i-vector describing the i initial values of the blackboard data objects, $\mathbf{Id} \in P$;
- θ is a relation on β, where $\theta \subset \beta \cdot \beta$ and $\langle ks_j, ks_k \rangle \in \theta$ if and only if $\exists d_j \in X$ where: $d_j \in \text{OV}$ and $(ks_j) \wedge d_j \in \text{IV}(ks_k)$;
- If $\langle ks_j, ks_k \rangle \in \theta$, then ks_k is a *successor* of ks_j, and ks_j is a *predecessor* of ks_k.

b) Performance Analysis of the Blackboard Systems Design

The performance of a blackboard system design can be analysed in the following manner (McManus 1991): for each knowledge source ks_j in β is an input set, Ψ_j, containing all of the input variables of ks_j and an output set, Φ_j, containing all of the output variables of ks_j

$$\begin{aligned} \Psi_j &= \{iv_1, iv_2, \ldots, iv_n\} \\ \Phi_j &= \{ov_1, ov_2, \ldots, ov_m\} \end{aligned} \tag{5.118}$$

Once Ψ_j and Φ_j have been established for all ks_j in β, the sets $\Gamma_{j,k}$ and $\theta_{j,k}$ can be computed for all knowledge source pairs $\{ks_j, ks_k\}$ in β $(j \neq k)$

$$\begin{aligned} \Gamma_{j,k} &= \Phi_j \cap \Phi_k \\ \theta_{j,k} &= \Phi_j \cap \Psi_k \end{aligned} \tag{5.119}$$

As indicated, *output set overlap* is a measure of the specialisation of pairs of knowledge sources, whereas functional connectivity between the pairs of knowledge sources is a measure of their serialisation, and output to input connectivity is a measure of their interdependence.

Specialisation value The *output set overlap* is a measure of the *specialisation* of pairs of knowledge sources, whereby the set $\Gamma_{j,k}$ is computed to assess functional specialisation. The *cardinality* of the set $\Gamma_{j,k}$ for each pair $\{ks_j, ks_k\}$ in β is a measure of the *output overlap* for the pair $\{ks_j, ks_k\}$ (i.e. a measure of the specialisation

of pairs of knowledge sources). Knowledge source pairs $\{ks_j, ks_k\}$ with a large output overlap imply that ks_j and ks_k share a large number of output variables and, thus, have similar functions. Knowledge source pairs $\{ks_j, ks_k\}$ with a low overlap imply that ks_j and ks_k have different functions. A proposed heuristic to measure *knowledge source specialisation* is to compute a *specialisation value*, $\Omega_{j,k}$, for each pair $\{ks_j, ks_k\}$ in β. Specialisation values measure the output set overlap of a pair of knowledge sources, $\{ks_j, ks_k\}$. The specialisation value is computed using the following (McManus 1992):

$$\Omega_{j,k} = \frac{\text{card}(\Gamma_{j,k})}{\min(\text{card}(\Phi_j), \text{card}(\Phi_k))} \tag{5.120}$$

The cardinality of the set $\Gamma_{j,k}$ divided by the minimum of the cardinalities of the sets Φ_j and Φ_k computes a percentage of overlap between the set $\Gamma_{j,k}$ and the smaller of the sets Φ_j and Φ_k. As $\Omega_{j,k}$ approaches 1.0, the output overlap between ks_j and ks_k *increases*. As $\Omega_{j,k}$ approaches 0.0, the output overlap between ks_j and ks_k *decreases*. For the limiting cases, where $\Phi_j \supset \Phi_k$ or $\Phi_k \supset \Phi_j$, we know that $\Omega_{j,k} = 1.0$, and ks_j and ks_k compute the same outputs—thus, the knowledge sources are not specialised. However, if $\Gamma_{j,k} = \phi$ (where ϕ is the null value), then $\Omega_{j,k} = 0.0$, and the two knowledge sources have no common outputs and are highly specialised in relation to each other.

Serialisation value The *functional connectivity* between pairs of knowledge sources is a measure of their *serialisation*, whereby the set $\theta_{j,k}$ is computed to assess serialisation. The *cardinality* of the set $\theta_{j,k}$ for each pair $\{ks_j, ks_k\}$ in β, compared to the cardinality of the set Ψ_k, is a measure of the *input overlap* for the pair $\{ks_j, ks_k\}$ (i.e. a measure of the serialisation of pairs of knowledge sources). Knowledge source pairs $\{ks_j, ks_k\}$ with a large input overlap imply that ks_j and ks_k share a large number of output to input variables and, thus, form serialised execution. Knowledge source pairs $\{ks_j, ks_k\}$ with a low input overlap imply that ks_j and ks_k can execute separately. A serialisation value measures the functional connectivity between a pair of knowledge sources where the functional connectivity is the relative output to input ratio. A proposed heuristic, therefore, to measure *knowledge source serialisation* is to compute a *serialisation value*, $\Sigma_{j,k}$, for each pair $\{ks_j, ks_k\}$ in β. Serialisation values measure the functional connectivity of a pair of knowledge sources, $\{ks_j, ks_k\}$.

The serialisation value is computed using (McManus 1992):

$$\Sigma_{j,k} = \frac{(\text{card}\theta_{j,k})}{(\text{card}\Psi_k)} \tag{5.121}$$

This heuristic computes the percentage of the input data objects for knowledge source ks_k that are provided by knowledge source ks_j. The cardinality of the set $\theta_{j,k}$ divided by the cardinality of the set Ψ_k computes a percentage of input overlap between $\theta_{j,k}$ and Ψ_k.

As $\Sigma_{j,k}$ approaches 1.0, the percentage of overlap between $\theta_{j,k}$ and Ψ_k is greater, and the serialisation between ks_j and ks_k strengthens. As $\Sigma_{j,k}$ approaches 0.0, the serialisation between ks_j and ks_k weakens. For the limiting cases, if $\Psi_k \supset \Phi_j$, $\Pi_{j,k} = 1.0$, and ks_j and ks_k have direct serialisation. If $\theta_{j,k} = \phi$ (where ϕ is the null value), then $\Sigma_{j,k} = 0.0$, and the two knowledge sources are independent and can execute concurrently.

Strongly connected knowledge sources have high serialisation values. These knowledge sources form serialised execution *pipelines*, with each knowledge source blocking completion of any computation for the same input data objects by other knowledge sources. Unless multiple copies of the serialised knowledge sources are developed, the serial pipelines reduce the blackboard's capability for *concurrent execution*. Weakly connected knowledge sources reduce knowledge source serialisation and increase the opportunity for concurrent knowledge source execution. Knowledge source pairs that have high serialisation values are best suited for knowledge source integration whereby the first knowledge source provides all of the inputs to the second knowledge source. Such a *serially connected* pair of knowledge sources can be reduced to a single knowledge source that combines the functionality of the two.

Interdependence value The *output to input connectivity* between pairs of knowledge sources is a measure of their *interdependence*, whereby the set $\theta_{j,k}$ is computed to assess interdependence. The *cardinality* of the set $\theta_{j,k}$ for each pair $\{ks_j, ks_k\}$ in β is a measure of the output to input connectivity for the pair $\{ks_j, ks_k\}$. Knowledge source pairs $\{ks_j, ks_k\}$ with a high output to input connectivity imply that ks_k is *highly dependent* on ks_j for its input variables. Knowledge source pairs $\{ks_j, ks_k\}$ with a low output to input connectivity imply that ks_k's inputs are *independent* of ks_j's outputs.

A proposed heuristic to measure *knowledge source interdependence* is to compute an *interdependence value*, $\Pi_{j,k}$, for each pair $\{ks_j, ks_k\}$ in β. Interdependence values measure the output to input connectivity between knowledge sources, $\{ks_j, ks_k\}$. The interdependence value is computed using the following (McManus 1992):

$$\Pi_{j,k} = \frac{(\text{card}\theta_{j,k})}{\min(\text{card}(\Phi_j), \text{card}(\Psi_k))} \tag{5.122}$$

This heuristic computes the percentage of overlap between the sets Φ_j and Ψ_k, or the percentage of output data objects of ks_j that are used as input data objects by ks_k. The cardinality of the set $\theta_{j,k}$ divided by the minimum of the cardinalities of the sets Φ_j and Ψ_k computes a percentage of overlap between the set $\theta_{j,k}$ and the smaller of the sets Φ_j and Ψ_k. As $\Pi_{j,k}$ approaches 1.0, the output to input connectivity between ks_j and ks_k strengthens and the knowledge sources become more *interdependent*. As $\Pi_{j,k}$ approaches 0.0, the output to input connectivity between ks_j and ks_k weakens and the knowledge sources become *independent*. For the limiting cases, if $\Phi_j \supset \Psi_k$, $\Pi_{j,k} = 1.0$, and ks_j and ks_k have direct output to input connectivity and are interdependent. If the set $\theta_{j,k} = \phi$ (where ϕ is the null value), then $\Pi_{j,k} = 0.0$, and the two knowledge sources have no output to input connectivity and are independent.

c) Evaluation of the AIB Blackboard Model for Determining the Integrity of Engineering Design

The AIB blackboard model for determining the integrity of engineering design includes subsets of the knowledge sources and blackboard data objects that are used by the knowledge-based expert system section. This knowledge-based expert system section allows for the development of various expert systems, and is structured into facts, functions, conditions, constraints, rules and goals related to the subsets of the knowledge sources and blackboard data objects of process analysis, plant analysis and operations analysis sections. The primary subsets of the knowledge sources for the process analysis and plant analysis sections are described below in accordance with Fig. 5.82 illustrating the AIB blackboard model for engineering design integrity.

Process analysis section

- Let Ks_1 be the *process definition* module. This knowledge source makes use of six global data object inputs—d_{i1}, d_{i2}, d_{i3}, d_{i4}, d_{i5} and d_{i6}, which can be represented by the set of input variables $\mathrm{IV}_6 = \{iv_1, \ldots, iv_n\}$—as well as a *process description* input, and computes five data object outputs that can be represented by the set of output variables $\mathrm{OV}_5 = \{iv_1, \ldots, iv_n\}$, for the data object outputs d_{o1} to d_{o5}.

The data object *inputs* d_{i1} to d_{i6} and data object *outputs* d_{o1} to d_{o5}:

d_{i1} = Plant/facility
d_{i2} = Operation/area
d_{i3} = Section/building
d_{i4} = System/process
d_{i5} = Assembly/unit
d_{i6} = Component/item
d_{i7} = Process description
d_{o1} = Process sequence
d_{o2} = Mass balance
d_{o3} = Heat balance
d_{o4} = Energy balance
d_{o5} = Utilities balance.

- Let Ks_2 be the *performance assessment* module. This knowledge source makes use of the six global data object inputs d_{i1}, d_{i2}, d_{i3}, d_{i4}, d_{i5} and d_{i6}, as well as a *performance specification* set, d_{i8}, and computes a *performance output* variable set, d_{o6}.
  The *performance specification* set d_{i8} can be represented by the set of input variables $\mathrm{IV}_8 = \{iv_1, \ldots, iv_n\}$, where d_{i8} =performance specification data object with $\mathrm{IV}_8 = \{$efficiency, flow, precipitation, throughput, output, pressure, viscosity, absorption, temperature, losses, etc.$\}$.
  The *performance output* variable set d_{o6} can be represented by the set of output variables $\mathrm{OV}_6 = \{ov_1, \ldots, ov_m\}$, where d_{o6} is the performance output data object with $\mathrm{OV}_6 = \{$efficiency rating, flow rating, throughput rating, output rating, yield, pressure rating, consistency, temperature rating, productivity, etc.$\}$.
- Let Ks_3 be the *RAM assessment* module. This knowledge source makes use of the six global data object inputs d_{i1}, d_{i2}, d_{i3}, d_{i4}, d_{i5} and d_{i6}, as well as a *conditions description* set, d_{i9}, and computes a *conditions failure* output variable set, d_{o7}.

The *conditions description* set d_{i9} can be represented by the set of input variables $\mathrm{IV}_9 = \{iv_1, \ldots, iv_n\}$, where d_{i9} = conditions description data object with IV_9 = {function description, failure description, failure effects, failure consequences, failure causes, failure mode description, failure frequency, restoration tasks description, procedure description, maintainability, etc.}.
The *conditions failure* output variable set d_{o7} can be represented by the set of output variables $\mathrm{OV}_7 = \{ov_1, \ldots, ov_m\}$, where d_{o7} is the conditions failure data object with OV_7 = {failure severity, probability of consequence, failure risk, failure criticality, failure downtime, restoration downtime, availability, etc.}.

- Let Ks_4 be the *design assessment* module. This knowledge source makes use of the six global data object inputs d_{i1}, d_{i2}, d_{i3}, d_{i4}, d_{i5} and d_{i6}, as well as a *design specification* set, d_{i10}, and computes a *design criteria* output variable set, d_{o8}.
  The *design specification* set d_{i10} can be represented by the set of following input variables $\mathrm{IV}_{10} = \{iv_1, \ldots, iv_n\}$, where d_{i10} = design specification data object with IV_{10} = {mass, volume, capacity, circulation, agitation, fluids, solids, consumption, heat input, energy input, etc.}.
  The *design criteria* output variable set d_{o8} can be represented by the set of output variables $\mathrm{OV}_8 = \{ov_1, \ldots, ov_m\}$, where d_{o8} is the design criteria data object with OV_8 = {efficiency, flow, precipitation, throughput, output, pressure, viscosity, absorption, temperature, losses, etc.}.
- Let Ks_5 be the *hazardous operations (HazOp) assessment* module. This knowledge source makes use of the six global data object inputs d_{i1}, d_{i2}, d_{i3}, d_{i4}, d_{i5} and d_{i6}, as well as the *operational hazards* set d_{i11}, and computes an *operational risk* output variable set, d_{o9}.
  The *operational hazards* set d_{i11} can be represented by the set of input variables $\mathrm{IV}_{11} = \{iv_1, \ldots, iv_n\}$, where d_{i11} = operational hazards data object with IV_{11} = {efficiency rating, flow rating, throughput rating, output rating, pressure rating, temperature rating, design torque, design stress, etc.}.
  The *operational risk* output variable set d_{o9} can be represented by the set of output variables $\mathrm{OV}_9 = \{ov_1, \ldots, ov_m\}$, where d_{o9} is the operational risk data object with OV_9 = {operational failure description, operational failure effects, operational failure consequences, operational failure causes, etc.}.

Systems analysis section

- Let Ks_6 be the *systems definition* module. This knowledge source makes use of the six global data object inputs d_{i1}, d_{i2}, d_{i3}, d_{i4}, d_{i5} and d_{i6}, as well as a *systems description* input, d_{i12}, and computes a *systems definition* output variable set, d_{o10}.
  There is no output variable set for *systems description* input.
  The *systems definition* output variable set d_{o10} can be represented by the set of output variables $\mathrm{OV}_{10} = \{ov_1, \ldots, ov_m\}$, where d_{o10} is the systems definition data object with OV_{10} = {system efficiency rating, system flow rating, system output rating, system pressure rating, system temperature rating, etc.}.
- Let Ks_7 be the *functions analysis* module. This knowledge source makes use of the six global data object inputs d_{i1}, d_{i2}, d_{i3}, d_{i4}, d_{i5} and d_{i6}, as well as a *func-*

tions description input, d_{i13}, and computes a *functions definition* output variable set, d_{o11}.
There is no output variable set for *functions description* input.
The *functions definition* output variable set d_{o11} can be represented by the set of output variables $\mathrm{OV}_{11} = \{ov_1, \ldots, ov_m\}$, where d_{o11} is the functions definition object with $\mathrm{OV}_{11} = \{$type, make, size, weight, capacity, cooling, insulation, power rating, power source, governing, rotation, speed, acceleration, torque, stress, voltage, current, etc.$\}$.

- Let Ks_8 be the *FMEA* module. This knowledge source makes use of the six global data object inputs d_{i1}, d_{i2}, d_{i3}, d_{i4}, d_{i5} and d_{i6}, as well as a *failure modes* set, d_{i14}, and computes a *failure effects* output variable set, d_{o12}.
  The *failure modes* set d_{i14} can be represented by the set of input variables $\mathrm{IV}_{14} = \{iv_1, \ldots, iv_n\}$, where d_{i14} is the failure modes data object with $\mathrm{IV}_{14} = \{$system failure description, system failure mode description, etc.$\}$.
  The *failure effects* output variable set d_{o12} is represented by the set of output variables $\mathrm{OV}_{12} = \{ov_1, \ldots, ov_m\}$, where d_{o12} is the failure effects data object with $\mathrm{OV}_{12} = \{$system failure effects, system failure severity, etc.$\}$.
- Ks_9 is the *risk evaluation* module. This knowledge source makes use of the six global data object inputs d_{i1}, d_{i2}, d_{i3}, d_{i4}, d_{i5} and d_{i6}, as well as a *risk identification* set, d_{i15}, and computes a *failure risk* output variable, d_{o13}.
  The *risk identification* set d_{i15} can be represented by the set of input variables $\mathrm{IV}_{15} = \{iv_1, \ldots, iv_n\}$, where d_{i15} is the risk identification data object with $\mathrm{IV}_{15} = \{$system failure effects, system failure consequences, system failure mode description, system probability of consequence, system failure severity, system failure frequency, system failure risk, etc.$\}$.

Table 5.28 The AIB blackboard data object construct

Data object input variables	Data object output variables
d_{i1} = Plant/facility	d_{o1} = Process sequence
d_{i2} = Operation/area	d_{o2} = Mass balance
d_{i3} = Section/building	d_{o3} = Heat balance
d_{i4} = System/process	d_{o4} = Energy balance
d_{i5} = Assembly/unit	d_{o5} = Utilities balance
d_{i6} = Component/item	d_{o6} = Performance output
d_{i7} = Process description	d_{o7} = Conditions failure
d_{i8} = Performance specification	d_{o8} = Design criteria
d_{i9} = Conditions description	d_{o9} = Operational risk
d_{i10} = Design specification	d_{o10} = Systems definition
d_{i11} = Operational hazards	d_{o11} = Functions definition
d_{i12} = Systems description	d_{o12} = Failure effects
d_{i13} = Functions description	d_{o13} = Failure risk
d_{i14} = Failure modes	d_{o14} = Failure criticality
d_{i15} = Risk identification	
d_{i16} = Failure identification	

- Let Ks_{10} be the *criticality analysis* module. This knowledge source makes use of the six global data object inputs d_{i1}, d_{i2}, d_{i3}, d_{i4}, d_{i5} and d_{i6}, as well as a *failure identification* set, d_{i16}, and computes a *failure criticality* output variable, d_{o14}.
  The *failure identification* set d_{i16} can be represented by the set of input variables $\mathrm{IV}_{16} = \{iv_1, \ldots, iv_n\}$, where d_{i16} is the failure identification data object with $\mathrm{IV}_{16} = \{$system function description, system failure description, system failure consequences, system failure causes, system failure mode description, system failure frequency, system probability of consequence, system failure severity, system failure frequency, system failure risk, etc.$\}$.

The input and output variable sets are summarised in Table 5.28.

d) The AIB Blackboard Model Specifications

The AIB blackboard model developed for determining the integrity of engineering design, has basically three levels of application which in effect divides the blackboard model into three separate blackboard sections: a *process design blackboard section (B1)*, a *systems design blackboard section (B2)*, and a *systems procedures blackboard section (B3)*. The process design blackboard section, (B1), is constrained to the input and output variables directly related to the *process analysis* section, while the systems design blackboard section, (B2), is constrained to the input and output variables directly related to the *plant analysis* section, and the systems procedures blackboard section, (B3), is constrained to the input and output variables directly related to the *operations analysis* section.

Specification of the process design blackboard section (B1)

$$
\begin{aligned}
X_i &= \{d_{i1}, d_{i2}, d_{i3}, d_{i4}, d_{i5}, d_{i6}, d_{i7}, d_{i8}, d_{i9}, d_{i10}, d_{i11}\};\\
X_o &= \{d_{o1}, d_{o2}, d_{o3}, d_{o4}, d_{o5}, d_{o6}, d_{o7}, d_{o8}, d_{o9}\};\\
P_i &= \{\mathrm{IV}_6 \times \mathrm{IV}_8 \times \mathrm{IV}_9 \times \mathrm{IV}_{10} \times \mathrm{IV}_{11}\};\\
P_o &= \{\mathrm{OV}_6 \times \mathrm{OV}_7 \times \mathrm{OV}_8 \times \mathrm{OV}_9\};\\
\beta &= \{ks_1, ks_2, ks_3, ks_4, ks_5\};
\end{aligned}
$$

where d_{i1}, d_{i2}, d_{i3}, d_{i4}, d_{i5}, $d_{i6} = \mathrm{IV}_6$, d_{o1}, d_{o2}, d_{o3}, d_{o4}, $d_{o5} = \mathrm{OV}_5$;

$$
\begin{aligned}
ks_1 &= \{\mathrm{IV}_6, d_{i7}, \mathrm{OV}_5\}\\
ks_2 &= \{\mathrm{IV}_6, d_{i8}, d_{o6}\} = \{\mathrm{IV}_6, \mathrm{IV}_8, \mathrm{OV}_6\}\\
ks_3 &= \{\mathrm{IV}_6, d_{i9}, d_{o7}\} = \{\mathrm{IV}_6, \mathrm{IV}_9, \mathrm{OV}_7\}\\
ks_4 &= \{\mathrm{IV}_6, d_{i10}, d_{o8}\} = \{\mathrm{IV}_6, \mathrm{IV}_{10}, \mathrm{OV}_8\}\\
ks_5 &= \{\mathrm{IV}_6, d_{i11}, d_{o9}\} = \{\mathrm{IV}_6, \mathrm{IV}_{11}, \mathrm{OV}_9\}\ .
\end{aligned}
$$

For each knowledge source ks_j in β is an input set, Ψ_j, containing all of the input variables of ks_j, and an output set, Φ_j, containing all of the output variables of ks_j:

$$
\begin{aligned}
\Psi_1 &= \{\mathrm{IV}_6, d_{i7}\} \qquad \Phi_1 = \{\mathrm{OV}_5\}\\
\Psi_2 &= \{\mathrm{IV}_6, \mathrm{IV}_8\} \qquad \Phi_2 = \{\mathrm{OV}_6\}\\
\Psi_3 &= \{\mathrm{IV}_6, \mathrm{IV}_9\} \qquad \Phi_3 = \{\mathrm{OV}_7\}\\
\Psi_4 &= \{\mathrm{IV}_6, \mathrm{IV}_{10}\} \qquad \Phi_4 = \{\mathrm{OV}_8\}\\
\Psi_5 &= \{\mathrm{IV}_6, \mathrm{IV}_{11}\} \qquad \Phi_5 = \{\mathrm{OV}_9\}\ .
\end{aligned}
$$

Table 5.29 Computation of $\Gamma_{j,k}$ and $\theta_{j,k}$ for blackboard B1

$\Gamma_{1,2} = \Phi_1 \cap \Phi_2 = \{\text{OV}_5\} \cap \{\text{OV}_6\} = 0$	$\theta_{1,2} = \Phi_1 \cap \Psi_2 = \{\text{OV}_5\} \cap \{\text{IV}_6, \text{IV}_8\} = 0$
$\Gamma_{1,3} = \Phi_1 \cap \Phi_3 = \{\text{OV}_5\} \cap \{\text{OV}_7\} = 0$	$\theta_{1,3} = \Phi_1 \cap \Psi_3 = \{\text{OV}_5\} \cap \{\text{IV}_6, \text{IV}_9\} = 0$
$\Gamma_{1,4} = \Phi_1 \cap \Phi_4 = \{\text{OV}_5\} \cap \{\text{OV}_8\} = 0$	$\theta_{1,4} = \Phi_1 \cap \Psi_4 = \{\text{OV}_5\} \cap \{\text{IV}_6, \text{IV}_{10}\} = 0$
$\Gamma_{1,5} = \Phi_1 \cap \Phi_5 = \{\text{OV}_5\} \cap \{\text{OV}_9\} = 0$	$\theta_{1,5} = \Phi_1 \cap \Psi_5 = \{\text{OV}_5\} \cap \{\text{IV}_6, \text{IV}_{11}\} = 0$
$\Gamma_{2,1} = \Phi_2 \cap \Phi_1 = \{\text{OV}_6\} \cap \{\text{OV}_5\} = 0$	$\theta_{2,1} = \Phi_2 \cap \Psi_1 = \{\text{OV}_6\} \cap \{\text{IV}_6, \text{IV}_7\} = 0$
$\Gamma_{2,3} = \Phi_2 \cap \Phi_3 = \{\text{OV}_6\} \cap \{\text{OV}_7\} = 0$	$\theta_{2,3} = \Phi_2 \cap \Psi_3 = \{\text{OV}_6\} \cap \{\text{IV}_6, \text{IV}_9\} = 0$
$\Gamma_{2,4} = \Phi_2 \cap \Phi_4 = \{\text{OV}_6\} \cap \{\text{OV}_8\} = 0.7$	$\theta_{2,4} = \Phi_2 \cap \Psi_4 = \{\text{OV}_6\} \cap \{\text{IV}_6, \text{IV}_{10}\} = 0$
$\Gamma_{2,5} = \Phi_2 \cap \Phi_5 = \{\text{OV}_6\} \cap \{\text{OV}_9\} = 0$	$\theta_{2,5} = \Phi_2 \cap \Psi_5 = \{\text{OV}_6\} \cap \{\text{IV}_6, \text{IV}_{11}\} = 0.7$
$\Gamma_{3,1} = \Phi_3 \cap \Phi_1 = \{\text{OV}_7\} \cap \{\text{OV}_5\} = 0$	$\theta_{3,1} = \Phi_3 \cap \Psi_1 = \{\text{OV}_7\} \cap \{\text{IV}_6, d_{i7}\} = 0$
$\Gamma_{3,2} = \Phi_3 \cap \Phi_2 = \{\text{OV}_7\} \cap \{\text{OV}_6\} = 0$	$\theta_{3,2} = \Phi_3 \cap \Psi_2 = \{\text{OV}_7\} \cap \{\text{IV}_6, \text{IV}_8\} = 0$
$\Gamma_{3,4} = \Phi_3 \cap \Phi_4 = \{\text{OV}_7\} \cap \{\text{OV}_8\} = 0$	$\theta_{3,4} = \Phi_3 \cap \Psi_4 = \{\text{OV}_7\} \cap \{\text{IV}_6, \text{IV}_{10}\} = 0$
$\Gamma_{3,5} = \Phi_3 \cap \Phi_5 = \{\text{OV}_7\} \cap \{\text{OV}_9\} = 0$	$\theta_{3,5} = \Phi_3 \cap \Psi_5 = \{\text{OV}_7\} \cap \{\text{IV}_6, \text{IV}_{11}\} = 0$
$\Gamma_{4,1} = \Phi_4 \cap \Phi_1 = \{\text{OV}_8\} \cap \{\text{OV}_5\} = 0$	$\theta_{4,1} = \Phi_4 \cap \Psi_1 = \{\text{OV}_8\} \cap \{\text{IV}_6, d_{i7}\} = 0$
$\Gamma_{4,2} = \Phi_4 \cap \Phi_2 = \{\text{OV}_8\} \cap \{\text{OV}_6\} = 0.6$	$\theta_{4,2} = \Phi_4 \cap \Psi_2 = \{\text{OV}_8\} \cap \{\text{IV}_6, \text{IV}_8\} = 1.0$
$\Gamma_{4,3} = \Phi_4 \cap \Phi_3 = \{\text{OV}_8\} \cap \{\text{OV}_7\} = 0$	$\theta_{4,3} = \Phi_4 \cap \Psi_3 = \{\text{OV}_8\} \cap \{\text{IV}_6, \text{IV}_9\} = 0$
$\Gamma_{4,5} = \Phi_4 \cap \Phi_5 = \{\text{OV}_8\} \cap \{\text{OV}_9\} = 0$	$\theta_{4,5} = \Phi_4 \cap \Psi_5 = \{\text{OV}_8\} \cap \{\text{IV}_6, \text{IV}_{11}\} = 0.6$
$\Gamma_{5,1} = \Phi_5 \cap \Phi_1 = \{\text{OV}_9\} \cap \{\text{OV}_5\} = 0$	$\theta_{5,1} = \Phi_5 \cap \Psi_1 = \{\text{OV}_9\} \cap \{\text{IV}_6, d_{i7}\} = 0$
$\Gamma_{5,2} = \Phi_5 \cap \Phi_2 = \{\text{OV}_9\} \cap \{\text{OV}_6\} = 0$	$\theta_{5,2} = \Phi_5 \cap \Psi_2 = \{\text{OV}_9\} \cap \{\text{IV}_6, \text{IV}_8\} = 0$
$\Gamma_{5,3} = \Phi_5 \cap \Phi_3 = \{\text{OV}_9\} \cap \{\text{OV}_7\} = 0$	$\theta_{5,3} = \Phi_5 \cap \Psi_3 = \{\text{OV}_9\} \cap \{\text{IV}_6, \text{IV}_9\} = 1.0$
$\Gamma_{5,4} = \Phi_5 \cap \Phi_4 = \{\text{OV}_9\} \cap \{\text{OV}_8\} = 0$	$\theta_{5,4} = \Phi_5 \cap \Psi_4 = \{\text{OV}_9\} \cap \{\text{IV}_6, \text{IV}_{10}\} = 0$

Once Ψ_j and Φ_j have been established for all ks_j in β, the sets $\Gamma_{j,k}$ and $\theta_{j,k}$ can be computed for all knowledge source pairs $\{ks_j, ks_k\}$ in β $(j \neq k)$ where $\Gamma_j, k = \Phi_j \cap \Phi_k$ and $\theta_{j,k} = \Phi_j \cap \Psi_k$. The set $\Gamma_{j,k}$ is computed to assess functional specialisation, whereas the set $\theta_{j,k}$ is computed to assess serialisation and interdependence (Table 5.29).

Knowledge source specialisation $\Omega_{j,k}$ is computed from (Eq. 5.120), knowledge source serialisation $\Sigma_{j,k}$ is computed from (Eq. 5.121), and knowledge source interdependence $\Pi_{j,k}$ is computed from (Eq. 5.122) (McManus 1992).

From Table 5.29, the sets $\Gamma_{j,k}$ and $\theta_{j,k}$ for the pairs of data objects that are zero indicate that their specialisation, serialisation and interdependence are also zero, with the conclusion that the relevant knowledge sources are highly specialised with no serialisation and total independence, making these suitable for *concurrent execution*.

However, the sets $\Gamma_{j,k}$ and $\theta_{j,k}$ for certain pairs of data objects that are not zero indicate that their specialisation, serialisation *or* interdependence will also not be zero, resulting in a diminished capability for concurrent execution. These sets' values are given below (Table 5.30).

Table 5.30 Computation of non-zero $\Omega_{j,k}, \Sigma_{j,k}$ and $\Pi_{j,k}$ for blackboard B1

$\Gamma_{2,4} = 0.7$	$\theta_{2,4} = 0$	$\Omega_{2,4} = 0.67$	$\Sigma_{2,4} = 0$	$\Pi_{2,4} = 0$
$\Gamma_{2,5} = 0$	$\theta_{2,5} = 0.7$	$\Omega_{2,5} = 0$	$\Sigma_{2,5} = 0.43$	$\Pi_{2,5} = 0.67$
$\Gamma_{4,2} = 0.6$	$\theta_{4,2} = 1.0$	$\Omega_{4,2} = 0.67$	$\Sigma_{4,2} = 1.0$	$\Pi_{4,2} = 1.0$
$\Gamma_{4,5} = 0$	$\theta_{4,5} = 0.6$	$\Omega_{4,5} = 0$	$\Sigma_{4,5} = 0.75$	$\Pi_{4,5} = 0.75$
$\Gamma_{5,3} = 0$	$\theta_{5,3} = 1.0$	$\Omega_{5,3} = 0$	$\Sigma_{5,3} = 0.40$	$\Pi_{5,3} = 1.0$

Specification of the systems design blackboard section (B2)

$$
\begin{aligned}
X_i &= \{d_{i1}, d_{i2}, d_{i3}, d_{i4}, d_{i5}, d_{i6}, d_{i12}, d_{i13}, d_{i14}, d_{i15}, d_{i16}\};\\
X_o &= \{d_{o10}, d_{o11}, d_{o12}, d_{o13}, d_{o14}\};\\
P_i &= \{\mathrm{IV}_6 \times \mathrm{IV}_{14} \times \mathrm{IV}_{15} \times \mathrm{IV}_{16}\};\\
P_o &= \{\mathrm{OV}_{10} \times \mathrm{OV}_{11} \times \mathrm{OV}_{12}\};\\
\beta &= \{ks_6, ks_7, ks_8, ks_9, ks_{10}\};
\end{aligned}
$$

where d_{i1}, d_{i2}, d_{i3}, d_{i4}, d_{i5}, $d_{i6} = \mathrm{IV}_6$ and $d_{o10} = \mathrm{OV}_{10}$;

$$
\begin{aligned}
ks_6 &= \{\mathrm{IV}_6, d_{i12}, \mathrm{OV}_{10}\}\\
ks_7 &= \{\mathrm{IV}_6, d_{i13}, d_{o11}\} = \{\mathrm{IV}_6, d_{i13}, \mathrm{OV}_{11}\}\\
ks_8 &= \{\mathrm{IV}_6, d_{i14}, d_{o12}\} = \{\mathrm{IV}_6, d_{i14}, \mathrm{OV}_{12}\}\\
ks_9 &= \{\mathrm{IV}_6, d_{i15}, d_{o13}\} = \{\mathrm{IV}_6, \mathrm{IV}_{15}, d_{o13}\}\\
ks_{10} &= \{\mathrm{IV}_6, d_{i16}, d_{o14}\} = \{\mathrm{IV}_6, \mathrm{IV}_{16}, d_{o14}\}\ .
\end{aligned}
$$

For each knowledge source ks_j in β is an input set, Ψ_j, containing all of the input variables of ks_j and an output set, Φ_j, containing all of the output variables of ks_j:

$$
\begin{aligned}
\Psi_6 &= \{\mathrm{IV}_6, d_{i12}\} & \Phi_6 &= \{\mathrm{OV}_{10}\}\\
\Psi_7 &= \{\mathrm{IV}_6, d_{i13}\} & \Phi_7 &= \{\mathrm{OV}_{11}\}\\
\Psi_8 &= \{\mathrm{IV}_6, \mathrm{IV}_{14}\} & \Phi_8 &= \{\mathrm{OV}_{12}\}\\
\Psi_9 &= \{\mathrm{IV}_6, \mathrm{IV}_{15}\} & \Phi_9 &= \{d_{o13}\}\\
\Psi_{10} &= \{\mathrm{IV}_6, \mathrm{IV}_{16}\} & \Phi_{10} &= \{d_{o14}\}\ .
\end{aligned}
$$

Once Ψ_j and Φ_j have been established for all ks_j in β, the sets $\Gamma_{j,k}$ and $\theta_{j,k}$ can be computed for all knowledge source pairs $\{ks_j, ks_k\}$ in β $(j \neq k)$ where $\Gamma_{j,k} = \Phi_j \cap \Phi_k$ and $\theta_{j,k} = \Phi_j \cap \Psi_k$. The set $\Gamma_{j,k}$ is computed to assess functional specialisation, whereas the set $\theta_{j,k}$ is computed to assess serialisation and interdependence.

From Table 5.31, the sets $\Gamma_{j,k}$ and $\theta_{j,k}$ for the pairs of data objects that are zero indicate that their specialisation, serialisation and interdependence are also zero, with the conclusion that the relevant knowledge sources are highly specialised with no serialisation and total independence, making these suitable for *concurrent execution*.

However, the sets $\Gamma_{j,k}$ and $\theta_{j,k}$ for certain pairs of data objects that are not zero indicate that their specialisation, serialisation *or* interdependence will also not be zero, resulting in a diminished capability for concurrent execution.

These sets' values are given below (Table 5.32).

e) Findings of Specialisation, Serialisation or Interdependence Computation

As previously indicated, the set $\Gamma_{j,k}$ is computed to assess functional specialisation and the cardinality of the set $\Gamma_{j,k}$ for each pair $\{ks_j, ks_k\}$ in β is a measure of the output overlap for the pair $\{ks_j, ks_k\}$ (i.e. a measure of the specialisation of pairs of knowledge sources). Knowledge source pairs $\{ks_j, ks_k\}$ with a large output overlap imply that ks_j and ks_k share a large number of output variables and, thus, have similar functions. Knowledge source pairs $\{ks_j, ks_k\}$ with a low overlap imply that ks_j and ks_k have different functions.

Table 5.31 Computation of $\Gamma_{j,k}$ and $\theta_{j,k}$ for blackboard B2

$\Gamma_{6,7} = \Phi_6 \cap \Phi_7 = \{\text{OV}_{10}\} \cap \{\text{OV}_{11}\} = 0$	$\theta_{6,7} = \Phi_6 \cap \Psi_7 = \{\text{OV}_{10}\} \cap \{\text{IV}_6, \text{IV}_{14}\} = 0$
$\Gamma_{6,8} = \Phi_6 \cap \Phi_8 = \{\text{OV}_{10}\} \cap \{\text{OV}_{12}\} = 0$	$\theta_{6,8} = \Phi_6 \cap \Psi_8 = \{\text{OV}_{10}\} \cap \{\text{IV}_6, \text{IV}_{14}\} = 0$
$\Gamma_{6,9} = \Phi_6 \cap \Phi_9 = \{\text{OV}_{10}\} \cap \{d_{o13}\} = 0$	$\theta_{6,9} = \Phi_6 \cap \Psi_9 = \{\text{OV}_{10}\} \cap \{\text{IV}_6, \text{IV}_{15}\} = 0$
$\Gamma_{6,10} = \Phi_6 \cap \Phi_{10} = \{\text{OV}_{10}\} \cap \{d_{o14}\} = 0$	$\theta_{6,10} = \Phi_6 \cap \Psi_{10} = \{\text{OV}_{10}\} \cap \{\text{IV}_6, \text{IV}_{16}\} = 0$
$\Gamma_{7,6} = \Phi_7 \cap \Phi_6 = \{\text{OV}_{11}\} \cap \{\text{OV}_{10}\} = 0$	$\theta_{7,6} = \Phi_7 \cap \Psi_6 = \{\text{OV}_{11}\} \cap \{\text{IV}_6, d_{i12}\} = 0$
$\Gamma_{7,8} = \Phi_7 \cap \Phi_8 = \{\text{OV}_{11}\} \cap \{\text{OV}_{12}\} = 0$	$\theta_{7,8} = \Phi_7 \cap \Psi_8 = \{\text{OV}_{11}\} \cap \{\text{IV}_6, \text{IV}_{14}\} = 0$
$\Gamma_{7,9} = \Phi_7 \cap \Phi_9 = \{\text{OV}_{11}\} \cap \{d_{o13}\} = 0$	$\theta_{7,9} = \Phi_7 \cap \Psi_9 = \{\text{OV}_{11}\} \cap \{\text{IV}_6, \text{IV}_{15}\} = 0$
$\Gamma_{7,10} = \Phi_7 \cap \Phi_{10} = \{\text{OV}_{11}\} \cap \{d_{o14}\} = 0$	$\theta_{7,10} = \Phi_7 \cap \Psi_{10} = \{\text{OV}_{11}\} \cap \{\text{IV}_6, \text{IV}_{16}\} = 0$
$\Gamma_{8,6} = \Phi_8 \cap \Phi_6 = \{\text{OV}_{12}\} \cap \{\text{OV}_{10}\} = 0$	$\theta_{8,6} = \Phi_8 \cap \Psi_6 = \{\text{OV}_{12}\} \cap \{\text{IV}_6, d_{i12}\} = 0$
$\Gamma_{8,7} = \Phi_8 \cap \Phi_7 = \{\text{OV}_{12}\} \cap \{\text{OV}_{11}\} = 0$	$\theta_{8,7} = \Phi_8 \cap \Psi_7 = \{\text{OV}_{12}\} \cap \{\text{IV}_6, d_{i13}\} = 0$
$\Gamma_{8,9} = \Phi_8 \cap \Phi_9 = \{\text{OV}_{12}\} \cap \{d_{o13}\} = 0$	$\theta_{8,9} = \Phi_8 \cap \Psi_9 = \{\text{OV}_{12}\} \cap \{\text{IV}_6, \text{IV}_{15}\} = 1.0$
$\Gamma_{8,10} = \Phi_8 \cap \Phi_{10} = \{\text{OV}_{12}\} \cap \{d_{o14}\} = 0$	$\theta_{8,10} = \Phi_8 \cap \Psi_{10} = \{\text{OV}_{12}\} \cap \{\text{IV}_6, \text{IV}_{16}\} = 1.0$
$\Gamma_{9,6} = \Phi_9 \cap \Phi_6 = \{d_{o13}\} \cap \{\text{OV}_{10}\} = 0$	$\theta_{9,6} = \Phi_9 \cap \Psi_6 = \{d_{o13}\} \cap \{\text{IV}_6, d_{i12}\} = 0$
$\Gamma_{9,7} = \Phi_9 \cap \Phi_7 = \{d_{o13}\} \cap \{\text{OV}_{11}\} = 0$	$\theta_{9,7} = \Phi_9 \cap \Psi_7 = \{d_{o13}\} \cap \{\text{IV}_6, d_{i13}\} = 0$
$\Gamma_{9,8} = \Phi_9 \cap \Phi_8 = \{d_{o13}\} \cap \{\text{OV}_{12}\} = 0$	$\theta_{9,8} = \Phi_9 \cap \Psi_8 = \{d_{o13}\} \cap \{\text{IV}_6, \text{IV}_{14}\} = 0$
$\Gamma_{9,10} = \Phi_9 \cap \Phi_{10} = \{d_{o13}\} \cap \{d_{o14}\} = 0$	$\theta_{9,10} = \Phi_9 \cap \Psi_{10} = \{d_{o13}\} \cap \{\text{IV}_6, \text{IV}_{16}\} = 1.0$
$\Gamma_{10,6} = \Phi_{10} \cap \Phi_6 = \{d_{o14}\} \cap \{\text{OV}_{10}\} = 0$	$\theta_{10,6} = \Phi_{10} \cap \Psi_6 = \{d_{o14}\} \cap \{\text{IV}_6, d_{i12}\} = 0$
$\Gamma_{10,7} = \Phi_{10} \cap \Phi_7 = \{d_{o14}\} \cap \{\text{OV}_{11}\} = 0$	$\theta_{10,7} = \Phi_{10} \cap \Psi_7 = \{d_{o14}\} \cap \{\text{IV}_6, d_{i13}\} = 0$
$\Gamma_{10,8} = \Phi_{10} \cap \Phi_8 = \{d_{o14}\} \cap \{\text{OV}_{12}\} = 0$	$\theta_{10,8} = \Phi_{10} \cap \Psi_8 = \{d_{o14}\} \cap \{\text{IV}_6, \text{IV}_{14}\} = 0$
$\Gamma_{10,9} = \Phi_{10} \cap \Phi_9 = \{d_{o14}\} \cap \{\text{OV}_{13}\} = 0$	$\theta_{10,9} = \Phi_{10} \cap \Psi_9 = \{d_{o14}\} \cap \{\text{IV}_6, \text{IV}_{15}\} = 0$

Table 5.32 Computation of non-zero $\Omega_{j,k}, \Sigma_{j,k}$ and $\Pi_{j,k}$ for blackboard B2

$\Gamma_{8,9} = 0$	$\theta_{8,9} = 1.0$	$\Omega_{8,9} = 0$	$\Sigma_{8,9} = 0.28$	$\Pi_{8,9} = 1.0$
$\Gamma_{8,10} = 0$	$\theta_{8,10} = 1.0$	$\Omega_{8,10} = 0$	$\Sigma_{8,10} = 0.18$	$\Pi_{8,10} = 1.0$
$\Gamma_{9,10} = 0$	$\theta_{9,10} = 1.0$	$\Omega_{9,10} = 0$	$\Sigma_{9,10} = 0.64$	$\Pi_{9,10} = 1.0$

From Table 5.30, the knowledge sources $ks_2 = \{\text{IV}_6, \text{IV}_8, \text{OV}_6\}$ and $ks_4 = \{\text{IV}_6, \text{IV}_{10}, \text{OV}_8\}$ have a *relatively low level of functional specialisation* with a large output overlap, where ks_2 and ks_4 share a large number of output variables and, thus, have similar functions.

The knowledge source $ks_2 =$ the performance assessment module with output variable set $\text{OV}_6 = \{$efficiency rating, flow rating, throughput rating, output rating, yield, pressure rating, consistency, temperature rating, productivity, etc.$\}$.

The knowledge source $ks_4 =$ the design assessment module with output variable set $\text{OV}_8 = \{$efficiency, flow, precipitation, throughput, output, pressure, viscosity, absorption, temperature, losses, etc.$\}$.

Similarly, the set $\theta_{j,k}$ is computed to assess serialisation and interdependence.

The cardinality of the set $\theta_{j,k}$ for each pair $\{ks_j, ks_k\}$ in β, compared to the cardinality of the set Ψ_k, is a measure of the input overlap for the pair $\{ks_j, ks_k\}$ (i.e. a measure of the serialisation of pairs of knowledge sources). Knowledge source pairs $\{ks_j, ks_k\}$ with a large input overlap imply that ks_j and ks_k share a large number of output to input variables and, thus, form serialised execution. Knowledge source pairs $\{ks_j, ks_k\}$ with a low input overlap imply that ks_j and ks_k can execute separately.

Knowledge sources $ks_2 = \{IV_6, IV_8, OV_6\}$, $ks_4 = \{IV_6, IV_{10}, OV_8\}$ and $ks_5 = \{IV_6, IV_{11}, OV_9\}$ have a *relatively high level of serialisation and interdependence* with a large input overlap, and share a large number of output to input variables, thus forming serialised execution in the blackboard section (B1), related to the process analysis section.

Knowledge sources $ks_8 = \{IV_6, d_{i14}, OV_{12}\}$, $ks_9 = \{IV_6, IV_{15}, d_{o13}\}$ and $ks_{10} = \{IV_6, IV_{16}, d_{o14}\}$ also *have a relatively high level of serialisation and interdependence* with an input overlap, and share a varied number of output to input variables, thus forming serialised execution in the blackboard section (B2), related to the systems analysis section.

The relative input overlaps for knowledge sources ks_8 and ks_9 are small compared to that for knowledge source ks_{10}, which requires a significant effort for re-design of the knowledge source resulting in concentrated focus on ks_{10}.

Knowledge source ks_8 = the FMEA module with the input variable set $IV_{14} = \{$system failure description, system failure mode description, etc.$\}$. Knowledge source ks_9 = the risk evaluation module with the input variable set $IV_{15} = \{$system failure effects, system failure consequences, system failure mode description, system probability of consequence, system failure severity, system failure frequency, system failure risk, etc.$\}$. Knowledge source ks_{10} = the criticality analysis module with the input variable set $IV_{16} = \{$system function description, system failure description, system failure effects, system failure consequences, system failure causes, system failure mode description, system failure frequency, system probability of consequence, system failure severity, system failure frequency, system failure risk, etc.$\}$.

It is quite apparent that these knowledge sources share the same input variables, not necessarily requiring serialised execution based on their serialisation value, $\Sigma_{j,k}$, but having a tight output to input connectivity (value=1.0) where the knowledge sources *are totally interdependent*.

5.4.3 Application Modelling Outcome

Of the ten knowledge sources evaluated in the two blackboard sections, B1 and B2, for the process analysis section and the systems analysis section of the AIB blackboard model respectively, several knowledge sources failed to meet stringent constraints of specialisation, serialisation *or* interdependence. This prompted re-design of some of the knowledge sources' interconnectivity to minimise serialised execution in the AIB blackboard model, whereby *automated continual design reviews* could be conducted throughout the engineering design process on the basis of concurrent evaluations of design integrity in an integrated collaborative engineering design environment.

The performance assessment module and the design assessment module of the process analysis section were found to have a relatively low level of functional specialisation with a large output overlap, indicating that a large number of output vari-

ables were common and, thus, had similar functions. This necessitated combining the two knowledge sources both in access and in application during re-design of the knowledge sources, thereby enhancing *functional specialisation* of the *process design blackboard section (B1)*.

The FMEA module, risk evaluation module, and criticality analysis module of the systems analysis section of the AIB blackboard model had a relative input overlap, indicating that they shared a varied number of output to input variables, thus forming serialised execution. However, the relative input overlap for the FMEA and risk evaluation knowledge sources were small compared to the criticality analysis knowledge source. The relatively low serialisation value for the FMEA and risk evaluation modules indicated that these knowledge sources shared the same input variables but did not necessarily have complete serialised execution. The criticality analysis module had a relatively high serialisation value (64%), indicating the need for a high level of serialised execution. All three knowledge sources had a tight output to input connectivity (value=1.0), where the knowledge sources were totally interdependent. This necessitated combining the three knowledge sources both in access and in application during re-design of the knowledge sources, thereby enhancing *functional independence* of the *systems design blackboard section (B2)*.

5.5 Review Exercises and References

Review Exercises

1. Discuss and compare fault-tree analysis (FTA), root cause analysis (RCA), and event tree analysis (ETA) for determining system safety in engineering design.
2. Discuss the general application of cause-consequence analysis for determining system safety in engineering design.
3. Give a brief account of the process of hazardous operability (HazOp) studies in designing for safety, considering concepts such as design representations, entities and their attributes, guidewords and interpretations, process parameter selection, point of reference, consequences and safeguards, and deriving recommendations.
4. Explain deviations from design intent and screening for causes of deviations.
5. Discuss the significance of safety and risk analysis in engineering design.
6. Describe the use of cost risk models, considering feature-based costing, parametric costing and risk analysis in designing for safety.
7. Discuss traditional cost estimating and consider comparisons between parametric cost estimating and qualitative cost estimating.
8. Discuss the significance of risk cost analysis in designing for safety.
9. Discuss process operational risk modelling and give an overview of developing a risk hypothesis and risk equation and measures.
10. Give a brief account of the application of hazard and operability (HazOp) studies for risk prediction in designing for safety.

11. Give an example of primary and secondary keywords in a HazOp study for risk prediction in engineering design.
12. Briefly describe the steps in the HazOp study methodology.
13. Consider the concept of hazard and operability modelling.
14. Describe qualitative modelling for hazard identification in contrast to a quantitative representation of uncontrolled processes.
15. Discuss checking safety by reachability analysis.
16. Give a brief description of the application of Markov point processes in designing for safety.
17. Define point process parameters.
18. Explain Markov chains and critical risk in safety analysis.
19. Briefly discuss the application of Kolmogorov differential equations.
20. Describe the Q-matrix.
21. Discuss critical risk theory in designing for safety.
22. Explain the concept of delayed fatalities.
23. Give a brief account of fault-tree analysis (FTA) for safety systems design and assessment of safety protection systems.
24. Discuss design optimisation in designing for safety.
25. Describe the process of assessment of safety systems with FTA.
26. Describe common cause failures in root cause analysis (RCA).
27. Define CMF and CCF and consider problems with applying CCF in safety and risk analysis for engineering design
28. Explain point process event tree analysis in designing for safety by determining the source of risk and designing for safety requirements.
29. Define probabilistic safety evaluation (PSE)
30. Explain point process consequence analysis.
31. Discuss the relationship between cause-consequence analysis, FTA and reliability analysis.
32. Give a brief account of fault tree, reliability block diagram, and event tree transformations.
33. Briefly describe the process of RBD to fault tree transformation.
34. Briefly describe fault tree to RBD transformation.
35. Briefly describe RBD and fault tree to event tree transformation.
36. Briefly describe event tree to RBD and fault tree transformation.
37. Give a brief description of structuring the cause-consequence diagram with event ordering and cause-consequence diagram construction.
38. Discuss failure modes and safety effects (FMSE) evaluation.
39. Define safety criticality analysis.
40. Define risk-based maintenance.
41. Discuss the significance of safety criticality analysis and risk-based maintenance in designing for safety.
42. Discuss risk analysis and decision criteria in designing for safety.
43. Define qualitative criticality analysis.
44. Describe residual life evaluation.

45. Consider the concepts of failure probability, reliability and residual life in designing for safety.
46. Define sensitivity testing.
47. Consider establishing an analytic basis for developing an intelligent computer automated system, including concepts such as a computer automated design space.
48. Discuss preferences and fuzzy rules, and dynamic constraints and scenarios in developing an intelligent computer automated system.
49. Discuss evolutionary computing and evolutionary design.
50. Define evolutionary algorithms (EA).
51. Describe the fundamentals of evolutionary algorithms.
52. Define genetic algorithms (GA).
53. Describe the fundamentals of genetic algorithms (GA).
54. Consider genetic algorithms in optimal safety system design.
55. Give a brief account of safety design considerations in the design optimisation problem.
56. Discuss systems analysis with GAs and fault trees.
57. Describe the concepts of algorithm description and binary decision diagrams in GA methodology for optimal safety system design.
58. Give an example of a genetic algorithm application in designing for safety, with typical results expected of the GA methodology.
59. Briefly describe artificial neural network (ANN) modelling in designing for safety.
60. Give a brief description of the building blocks of artificial neural networks (ANNs) and consider a typical structure of the ANN.
61. Briefly describe the process of learning in artificial neural networks.
62. Consider back propagation in artificial neural networks.
63. Briefly discuss the application of fuzzy neural rule-based systems in designing for safety.
64. Give a brief account of the significance of artificial neural networks in engineering design.
65. Describe the various ANN computational architectures.

References

AFSC DH 1-6 (1967) System safety design handbook. United States Air Force Systems Command

AIChE (1985) Guidelines for event tree analysis. American Institute of Chemical Engineers, Center for Chemical Process Safety, New York

AIChE (1992) Guidelines for hazard evaluation procedures. American Institute of Chemical Engineers, Center for Chemical Process Safety, New York

Akers SB (1978) Binary decision diagrams. IEEE Trans Computers vol C-27, no 6, June

Andrews JD (1994) Optimal safety system design using fault tree analysis. Proc Inst Mech Engrs 208 I Mech E:123–131

Andrews JD, Morgan JM (1986) Application of the digraph method of fault tree construction to process plant. Reliability Eng 14:85–106

Andrews JD, Moss TR (1993) Reliability and risk assessment. American Society of Mechanical Engineers
Andrews JD, Pattison RL (1997) Optimal safety system performance. In: Proc Reliability and Maintainability Symp, Philadelphia, PA, pp 76–83
ANSTO (1994) The safety of nuclear power reactors. Nuclear Services Section Background Paper, Australian Nuclear Science and Technology Organisation
APT Maintenance (1999) Cost/risk evaluation & optimisation of planned maintenance. Asset Performance Tools, Berkshire
Aven T (1992) Reliability and risk analysis, 1st edn. Elsevier, Amsterdam
Bäck T (1994) Parallel optimisation of evolutionary algorithms. In: Proc Int Conf Evolutionary Computation. Springer, Berlin Heidelberg New York, pp 418–427
Beaumont GP (1986) Probability and random variables. Ellis Horwood, New York
Bellman RE, Dreyfus E (1962) Applied dynamic programming. Princeton University Press, Princeton, NJ
Ben Brahim S, Smith A, Bidanda B (1992) Estimating product performance and quality from design parameters via neural networks. In: Proc IIE Research Conf, pp 319–323
Blandford A, Butterworth B, Duke D, Good J, Milner R, Young R (1999) Programmable user modelling applications: incorporating human factors concerns into the design and safety engineering of complex control systems. Middlesex University Work Pap WP22, EPSRC Res Pap GR/L00391
Bourne AJ, Edwards GT, Watson IA (1981) Defences against common mode failures in redundancy systems. SRD R196, UKAEA
Bowles JB, Bonnell RD (1994) Failure mode effects and criticality analysis. In: Proc Annu Reliability and Maintainability Symp, pp 1–34
Bradley J (2001) A risk hypothesis and risk measures for throughput capacity in systems. Rep Department of Computer Science, University of Calgary
Bryant RE (1986) Graph-based algorithms for Boolean function manipulation. IEEE Trans Computers 35(8)
Chryssolouris G, Lee M, Pierce J, Domroese M (1989) Use of neural networks for the design of manufacturing systems. Proc American Society of Mechanical Engineers, pp 57–63
Coit DW, Smith AE (1994) Use of a genetic algorithm to optimize a combinatorial reliability design problems. In: Proc 3rd Int Engineering Research Conf, pp 467–472
Coit DW, Smith AE (1996) Stochastic formulations of the redundancy allocation problem. In: Proc 5th Industrial Engineering Research Conf, Minneapolis, MN, pp 459–463
Cvetkovic D, Parmee IC (1998) Evolutionary design and multi-objective optimisation. In: EUFIT, Aachen, pp 397–401
Cvetkovic D, Parmee IC, Webb E (1998) Multi-objective optimisation and preliminary design. In: Parmee IC (ed) Adaptive computing in design and manufacture. Springer, Berlin Heidelberg New York, pp 255–267
DEF STAN 00-58 (2000) HAZOP studies on systems containing programmable electronics. Part 2. General application guidance. Ministry of Defence, Defence Standard 00-58, Issue 2, 19
de Gelder P (1997) Deterministic and probabilistic safety analyses. Rep AVN AIB-Vinçotte Nuclear, AVN-97/014, O/Ref 97-2635/PDG, Class XP.00.NS
DOE-NE-STD-1004-92 (1992) Root cause analysis: guidance document. DOE Guideline, US Department of Energy, Office of Nuclear Energy, Washington, DC
Doerre P (1987) Some inconsistencies in CCF data evaluation and interpretation. In: Proc National Reliability Conf
EC (1996) Safety machinery—principles for risk assessment. European Community Rep EN 1050
ECI (2001) Designing for safe and healthy construction. Int Conf Designing for Safe and Healthy Construction, June 2000, European Construction Institute (ECI), Conseil Internationale du Bâtiment (CIB W99), London
Edwards GT, Watson IA (1979) A study of common mode failures. SRD R146 UKAEA
ExSys (2000) The ExSys Knowledge Automation Expert Systems Program. ExSys Inc, Albuquerque, NM

Extend (2001) Extend performance modelling for decision support. Imagine That Inc, San Jose, CA
Farell AE, Roat SD (1994) Framework for enhancing fault diagnosis capabilities of artificial neural networks. Computers Chem Eng 18(7):613–635
Fausett L (1994) Fundamentals of neural networks. Prentice Hall, Englewood Cliffs, NJ
Fodor J, Roubens M (1994) Fuzzy preference modelling and multicriteria decision support. Kluwer, Dordrecht
Fusaro RL (1998) Feasibility of using neural network models to accelerate the testing of mechanical systems. NASA Glenn's Research & Technology Reports, NASA Lewis Research Center
Fyffe DE, Hines WW, Lee NK (1968) System reliability allocation and a computational algorithm. IEEE Trans Reliability R-17:64–69
Gertman DI, Blackman HS (1994) Human reliability & safety analysis data handbook, 1st edn. Wiley, New York
Ghare PM, Taylor RE (1969) Optimal redundancy for reliability in series system. Operations Res 17:838–847
Goldberg DE (1989) Genetic algorithms in search, optimization & machine learning. Addison-Wesley, Reading, MA
Hanks BJ (1998) An appreciation of common cause failures in reliability. Proc Inst Mech Engrs 212 Part E:31–35
Haykin S (1999) Neural networks. Prentice Hall, Englewood Cliffs, NJ
Holland J (1992) Genetic algorithms. Scientific American, pp 44–50
Hughes RP (1987) A new approach to common-cause failure. Reliability Eng System Safety 17:211–236
ICS (2003) The Pro-RAM Artificial Intelligence Based Blackboard Model for Engineering Design. ICS Industrial Consulting Services, Gold Coast City, Queensland
Ida K, Gen M, Yokota T (1994) System reliability optimisation with several failure modes by genetic algorithm. In: Proc 16th Int Conf Computers and Industrial Engineering, pp 349–352
IEC 60300-3-9 (1995) Dependability management. Part 3. Application Guide Section 9. Risk Analysis of Technological Systems. International Electrotechnical Commission (IEC), Geneva
Ilott PW, Griffiths AJ (1997) Fault diagnosis of pumping machinery using artificial neural networks. Proc Inst Mech Engrs 211 Part E:185–194
Ilott PW, Griffiths AJ, Wililarns JM (1995) Condition monitoring of pumping systems. In: Proc 8th Natl Congr Condition Monitoring and Diagnostic Engineering Management, 1, pp 369–376
INPO 84-027 (1984) An Analysis of root causes in 1983 significant event reports. Rep 84-027, Institute of Nuclear Power Operations (INPO), Atlanta, GA
INPO NUMARC (1985) A maintenance analysis of safety significant events. NUMARC Committee Pap, Maintenance Work Group, Institute of Nuclear Power Operations (INPO), Atlanta, GA
Isograph (2001) The AvSim© Availability Simulation Model. Isograph, Irvine, CA
Kepner CH, Tregoe BB (1981) The new rational manager. Princeton Research Press, Princeton, NJ
Kletz T (1999) HAZOP and HAZAN: identifying and assessing process industry hazards. Institution of Chemical Engineers (IchemE), Warwickshire
Lefebvre C, Principe J (2002) NeuroSolutions: a network simulation environment. NeuroDimension, Gainesville, FL
Lippmann RP (1987) An introduction to computing with neural nets. IEEE ASSP Mag, pp 4–22
Marshall J, Newman R (1998) Reliability enhancement methodology and modeling for electronic equipment—the REMM Project. Proc ERA Avionics, pp 4.2.1–4.2.13
Matlab (1995) Fuzzy Logic Toolbox User's Guide. MathWorks, Natick, MA
McManus JW (1991) Design and analysis tools for concurrent blackboard systems. In: 10th AIAA/IEEE Proc Digital Avionics Systems
McManus JW (1992) Design and analysis techniques for concurrent blackboard systems. PhD Thesis, Faculty of the Department of Computer Science, College of William and Mary, Williamsburg, VA

Meisl C (1988) Techniques for cost estimating in early program phases. Eng Costs Production Economics 14:95–106
Michael J, Wood W (1989) Design to cost. Wiley, New York
Mileham RA, Currie CG, Miles AW, Bradford DT (1993) A parametric approach to cost estimating at the conceptual stage of design. J Eng Design 4(2):117–125
MIL-HDBK-217F (1998) Reliability prediction of electronic equipment. Notice 2 (217F-2), Department of Defense, Washington, DC
MIL-HDBK-764 (MI) (1990) System Safety Engineering Design Guide for Army Materiel. DoD, Washington, DC
MIL-STD-882 (1962) Systems Safety Program for System and Associated Sub-System and Equipment. DoD, Washington, DC
MIL-STD-882A (1977) Systems Safety Program for System and Associated Sub-System and Equipment. DoD, Washington, DC
MIL-STD-882B (1984) Systems Safety Program for System and Associated Sub-System and Equipment. DoD, Washington, DC
MIL-STD-882C (1993) Systems Safety Program for System and Associated Sub-System and Equipment. DoD, Washington, DC
MIL-STD-882D (2000) Systems Safety Program for System and Associated Sub-System and Equipment. DoD, Washington, DC
MIL-STD-38130 (1963) Safety Engineering of Systems and Associated Sub-Systems and Equipment. DoD, Washington, DC
Misra KB, Sharma U (1991) An efficient algorithm to solve integer programming problems arising in system reliability design. IEEE Trans Reliability 40:81–91
Nakagawa Y, Miyazaki S (1981) Surrogate constraints algorithm for reliability optimization problems with two constraints. IEEE Trans Reliability R-30:175–180
NASA 1359 (1994) System engineering toolbox for design-oriented engineers. National Aeronautics and Space Administration (NASA), Huntsville, AL
NASA DHB-S-00 (1999) System safety handbook. National Aeronautics and Space Administration (NASA), Dryden Flight Research Center, Edwards, CA
NeuroDimension (2001) NeuroSolutions and NeuralExpert. NeuroDimension, Gainsville, FL
Nielsen DS, Platz O, Runge B (1975) A cause-consequence chart of a redundant protection system. IEEE Trans Reliability 24(1)
NUREG 1150 (1989) Severe accident risks: an assessment for five US nuclear power plants. US Nuclear Regulatory Commission, NRC Rep NUREG 1150
NUREG 75/014 (1975) Reactor safety study: an assessment of accident risks in US commercial nuclear power plants. US Nuclear Regulatory Commission, NRC Rep WASH-1400, NUREG 75/014, NTIS
NUREG/CF-1401 (1980) Estimates for the binomial failure rate common-cause model. US Nuclear Regulatory Commission NRC Rep WASH-1400, NUREG/CF-1401
NUREG/CR-0400 (1978) Risk Assessment Review Group Report. US Nuclear Regulatory Commission NRC Rep WASH-0400
OECD NEA (1995) Chernobyl ten years on. Nuclear Energy Institute, Source Book
Oksendal B (1985) Stochastic differential equations: an introduction with applications. Springer, Berlin Heidelberg New York
Painton L, Campbell J (1995) Genetic algorithms in optimisation of system reliability. IEEE Trans Reliability 44(2):172–178
Pattison RL, Andrews JD (1999) Genetic algorithms in optimal safety system design. Proc Inst Mech Engrs 213 Part E:187–197
PCEI (1999) Parametric estimating handbook, 2nd edn. Joint Industry/Government Parametric Cost Estimating Initiative (PCEI), Department of Defense, Washington, DC, Defense Contract Audit Agency, Special Projects Division, VA
Price CJ (1996) Effortless incremental design FMEA. In: Proc Annu Reliability and Maintainability Symp, IEEE Press, pp 43–47

Rasmussen NC (1989) Report to the Congress from the Presidential Commission on Catastrophic Nuclear Accidents. Appendix B. The Nature of Severe Nuclear Accidents. MIT Ro 24-205
Rausand M (1999) Supplement SIO3020: safety and reliability engineering event tree analysis. Pap Department of Production and Quality Engineering, Norwegian University of Science and Technology, Trondheim
Rausand M (2000) Hazard identification (HAZID). Pap Department of Production and Quality Engineering, Norwegian University of Science and Technology, Trondheim
Ridley LM, Andrews JD (1996) Application of the cause-consequence diagram method to static systems. Pap Department of Mathematical Sciences, Loughborough University, Loughborough, Leicestershire
Roy R, Bendall D, Taylor JP, Jones P, Madariaga AP, Crossland J, Hamel J, Taylor IM (1999) Identifying and capturing the qualitative cost drivers within a concurrent engineering environment. Advances in Concurrent Engineering, Technomic, Lancaster, PA, pp 39–50
Rush C, Roy R (2000) Analysis of cost estimating processes used within a concurrent engineering environment throughout a product life cycle. In: Proc 7th Int Conf Concurrent Engineering, University Lyon 1
Schmerr LW, Nugen SM, Forourachi B (1991) Planning robust design experiments using neural networks and Taguchi methods. In: Dagli C, Kumara S, Shin Y (eds) Intelligent engineering systems through artificial neural networks. ASME Press, New York, pp 829–834
Schocken S (1994) Neural networks for decision support: problems and opportunities. Decision Support Systems 11(4):393–414
Siu N (1994) Risk assessment for dynamic systems: an overview. Reliability Eng System Safety 43:43–73
Smith AE, Coit DW (1996) Reliability optimization of series-parallel systems using a genetic algorithm. IEEE Trans Reliability 45(1)
Smith AE, Mason AK (1997) Cost estimation predictive modelling: regression versus neural network. Eng Econ 42(2):137–162
Smith TC, Smith B (2000) Survival analysis and the application of proportional hazards modelling. Pap 244-26, Statistics, Data Analysis and Data Mining, Center for Deployment, DoD, US Navy, San Diego, CA
Smith AE, Tate DM (1993) Genetic optimization using a penalty function. In: Proc 5th Int Conf Genetic Algorithms, pp 499–505
Smithers T, Conkie A, Doheny J, Logan B, Millington K, Tang M (1990) Design as intelligent behaviour: an AI in design research programme. Int J Artificial Intelligence Eng 5
Stuart JR, Norvig P (1995) AI: a modern approach. Prentice Hall, Englewood Cliffs, NJ
Suri R, Shimizu M (1989) Design for analysis: a new strategy to improve the design process. Res Eng Design 1:105–120
Tang M (1997) A knowledge-based architecture for intelligent design support. Int J Knowledge Eng Rev 12:4
Thompson WA (1988) Point process models with applications to safety and reliability. Chapman and Hall, New York
Tillman FA, Hwang CL, Kuo W (1977) Determining component reliability and redundancy for optimum system reliability. IEEE Trans Reliability R-26:162–165
Vaidhyanathan R, Venkatasubramanian V (1996) Experience with an expert system for automated HAZOP analysis. Computers Chem Eng suppl 20:1589–1594
Valluru BR (1995) Neural networks and fuzzy logic. M&T Books, IDG Books Worldwide, Foster City, CA
Villemeur A (1991) Reliability, availability, maintainability and safety assessment. Wiley, Chichester, NY
Wang XY, Yang SA, Veloso E, Lu ML, McGreavy C (1995) Qualitative process modeling—a fuzzy signed directed graph method. Computers Chem Eng 19:735–740
Watson IA (1981) Review of common cause failures. NCSR R27 UKAEA
Wierda LS (1991) Linking design, process planning and cost information by feature-based modelling. Eng Design 2(1):3–19

Woodhouse J (1999) Cost/risk optimisation. European MACRO Project, Woodhouse Partnership Ltd, Newbury, Berkshire
Zarefar H, Goulding JR (1992) Neural networks in design of products: a case study. In: Kusiak A (ed) Intelligent design and manufacturing. Wiley, New York, pp 179–201

Appendix A
Design Engineer's Scope of Work

Initial Definitive Study Planning and Implementation

Fully develop and detail the scope and implementation methodology of the definitive study and submit to the owner for approval. Specific deliverables to be submitted as part of this initial phase are to include:

- Study scope of work and specific study deliverables list.
- Study resourcing plan.
- Study schedule.
- Study budget.
- Study procedures.

Feasibility Studies

Carry out a number of feasibility studies leading to specific recommendations in order to confirm and validate the optimal plant design and configuration. Studies to be undertaken will include but will not be limited to:

- Plant throughput.
- Plant location.
- Onsite production of additives.
- Availability of local supplies of materials.

The following requirements are divided into the different engineering disciplines and their relevant activities, such as process engineering, control systems engineering, mechanical engineering, civil, structural architectural and environmental engineering, and electrical engineering.

R.F. Stapelberg, *Handbook of Reliability, Availability, Maintainability and Safety in Engineering Design*, © Springer 2009

Process Engineering

Testwork Review of all testwork completed to date together with a review of the proposed future testwork program. The results of any additional testwork undertaken are also to be incorporated into the design. The contractor is also expected to participate in any additional testwork program undertaken by way of attendance during testing and logging of results to ensure timely and accurate incorporation of data from testwork into the process design.

Process design Process engineering deliverables generally issued for detail design:

- Process description and block flow diagrams.
- Process design criteria.
- PFDs for normal, start-up, shutdown & upset conditions.
- Heat and material balances for normal, start-up, shutdown and non-steady-state conditions.
- Dynamic mass-balance simulation model.
- Plant water balance (including tailings & evaporation ponds).
- Process and utility P&IDs.
- Consumption, waste and emission summary.
- Utility summary.
- Process/utility integration and optimisation study for normal operation, start-up, shutdown and upset process conditions.
- Preliminary Hazop reviews.

Plant layout

- Dimensional site plan.
- Unit plot plans.
- General arrangement plans, elevations and sections.

Piping

- Piping design criteria.
- Pipe and valve specifications.
- Line and valve lists.
- Site plan review for critical and expensive pipe routings, access arrangements and process requirements.
- Preliminary MTOs in sufficient detail for estimate purposes.

Control Systems Engineering

- Control system, operating philosophy & strategy.
- Advanced controls—where applicable.
- Applicable codes & standards.
- DCS specifications.
- Instrumentation list.

- Inline instrument data sheets.
- Control and automation plan.
- Process package plant control philosophy.
- Emergency shutdown philosophy.
- Fire and gas detection philosophy.
- Plant communications philosophy.
- CCTV & UHF radio requirements.
- Instrument air and UPS requirements.
- Standard installation details.
- Specifications for general instruments, control valves and safety systems.
- Control room layout.

Mechanical Engineering

- Mechanical design criteria.
- Full equipment list.
- Technical specifications.
- Technical data sheets.
- Reliability and maintainability analysis.
- Maintenance spares list.

Civil, Structural and Architectural Engineering

- Civil, structural and architectural design criteria.
- Coordination and integration of geotechnical investigations and topographic surveys.
- Preliminary designs for:
  - Buildings; descriptions and conceptual designs for any required buildings and structures.
  - Water supply systems and dams.
  - Standard steelwork connection details.
  - Underground drainage:
    - sanitary.
    - contaminated storm water.
- Roads and site earthworks.
- Pipe racks—loads and congestion.
- Foundations—design requirements.

Electrical Engineering

- Electrical design criteria.
- Electrical equipment list.
- Electrical load list.
- Motor list.
- Technical specifications and data sheets.
- Preliminary design of all facilities downstream of the main power transformers through to main users including all transformers, sub-stations and MCCs.
- Voltage selection for high-KW motors.
- Emergency power supply requirements.
- Plant lighting design.
- Preliminary data and communication equipment requirements.
- Optimisation study on number and size of generating units.
- Power generation control philosophy.
- Load cycle strategy for various plant operating modes.
- Load sharing study between diesel and steam turbines.
- SLDs for each unit.
- Overall SLD for total power supply system.
- GAs for electrical equipment/sub-stations.
- Standard installation drawings.
- Standard schematic and termination drawings.
- Grounding/earthing system preliminary design.
- Cable ladder route layout drawings.
- MTOs for estimate purposes.

Loss Prevention

- Fire protection, and safety equipment requirements review.
- Plant layout review—spacing of equipment.
- Emergency shutdown plan.
- Area classification (schedule and layout drawings).
- Design of fire and gas detection systems.
- Design of fire protection system.
- Spill control/containment strategy.
- Noise control.
- Ventilation.

Environmental and Permitting

Liase, interface and support the nominated environmental consultant with the evaluation and assessment of impacts as required, including:

- Ambient air quality/source.
- Waste water discharge.
- Fugitive emissions.
- Noise regulations.
- Visual impacts.
- Product transportation issues.
- Permitting/statutory requirements.

Mining

Liase, interface and support the nominated mining consultant as required on activities that will include as a minimum:

- Geotechnical investigations.
- Pit optimisations.
- Preparation of pit designs and ore reserve statements.
- Mine scheduling.
- Preparation of waste dump and haul road designs.
- Pit permeability investigations.
- Determination of materials handling properties.
- Preparation of a detailed report.

Constructability and Logistics

Constructability and logistical study addressing the following:

- Identification of delivery routes and lifting/rigging of heavy equipment.
- Site access for construction equipment.
- Scope for modularisation and offsite assembly.
- Strategy for minimising double handling of equipment and different bulk materials.
- Strategy for minimising clashes onsite.
- Plan for incorporation of locally based contractors as appropriate.

Procurement

- Develop procurement policies and procedures.
- Issue & evaluate bids for major equipment items and sub-contracts.
- Develop installed equipment costs.
- List suitable vendors for key equipment.
- Identify long-lead items.

Development of Capital and Operating Cost

The capital and operating cost estimates will be developed into a format to be agreed by the owner. The estimates will be developed to an accuracy of $\pm 10\%$.

Development of the Project Schedule

- The master schedule will be developed for the project.
- The format and level of detail to be included is to be agreed by the owner.
- The master schedule must reflect the following:
  - Fabrication/installation schedules.
  - Vendor baseline commitments.
  - Construction schedules.
  - Commissioning schedules.

Value Engineering and Risk Assessment

The contractor will ensure that during the definitive study phase, engineering effort is directed at minimising the cost of the *EPC* phase of the project without introducing unacceptable risk. As part of this requirement, a full risk assessment will be undertaken on the project to ensure that all risks have been adequately identified and quantified. Significant effort will be put into the planning of the project delivery to ensure the best approach. The constructability of the plant and such issues as onsite or offsite pre-assembly of structures and vessels will be assessed for the impact on overall cost and schedule. During engineering, discussions will be held with the owner to look at ways to optimise the design especially the full utilisation of services and utilities. Commonality of designs will be considered to reduce spares inventories, and prior studies will be reviewed and incorporated where appropriate.

Project Execution Plan

A project execution plan will be prepared that includes the following sub-plans as a minimum:

- Occupational health and safety plan.
- Contracting plan.
- Industrial relation plan.
- Procurement plan.
- Human resources plan.
- Quality assurance plan.
- Automation plan.
- Procedures for the implementation phase of the project.

General

All work during the course of the definitive study is to be completed in accordance with procedures to be developed by the contractor and approved by the owner. The contractor will make suitable office facilities available for the owner's entire project team including office accommodation and general office administration and IT support. The contractor is to provide progressive reporting on the progress of the program together with cost and schedule status.

Final Report

The contractor will be responsible for the preparation of the final study report. This is to include preparation, compilation, review & editing, and final issue. The contractor will also be responsible for incorporating the owner's contributions into the full report where relevant. The format and content of the final report will be developed by the contractor and approved by the owner.

This report will include:

- A written description of the plant and all of its sub-facilities.
- A written description of the services provided.
- A written description of the major equipment required for each area of the plant.
- All the information produced as part of the services.

Ten copies of the final report (bound) are to be made available to the owner on completion, together with a computer hard disk drive containing the complete report, all of the study deliverables and all of the information/calculations, etc. used to develop the study deliverables. All information is to be appropriately logged to ensure its rapid retrieval if required.

Appendix B
Bibliography of Selected Literature

References [ ] = handbook chapter number

Ajmone Marsan M, Balbo G, Conte G, Donatelli S, Franceschinis G (1995) Modelling with generalised stochastic Petri nets. Wiley, Chichester, NY [4]

Aslaksen E, Belcher R (1992) Systems engineering. Prentice Hall of Australia [3]

Barnett V (1973) Comparative statistical inference. Wiley, Chichester, NY [3]

Beaumont GP (1986) Probability and random variables. Ellis Horwood, New York [5]

Bellman RE, Dreyfus E (1962) Applied dynamic programming. Princeton University Press, Princeton, NJ [5]

Bing G (1996) Due diligence techniques and analysis: critical questions for business decisions. Quorum Books, Westport, CT [4]

Blanchard BS, Fabrycky WJ (1990) Systems engineering and analysis. Prentice Hall, Englewood Cliffs, NJ [3]

Blanchard BS, Verma D, Peterson EL (1995) Maintainability: a key to effective serviceability and maintenance management. Prentice Hall, Englewood Cliffs, NJ [4]

Box GEP, Hunter WG, Hunter JS (1978) Statistics for experiments. Wiley, Chichester, NY [4]

Buchanan BG, Shortliffe EH (1984) Rule-based expert systems. Addison-Wesley, Reading, MA [3]

Bulgren WG (1982) Discrete system simulation. Prentice Hall, Englewood Cliffs, NJ [4]

Bussey LE (1978) The economic analysis of industrial projects. International Series in Industrial and Systems Engineering. Prentice Hall, Englewood Cliffs, NJ [4]

Carter ADS (1986) Mechanical reliability, 2nd edn. Macmillan Press, London [3]

Carter ADS (1997) Mechanical reliability and design. Macmillan Press, London [3]

Casti J (1979) Connectivity, complexity, and catastrophe in large-scale systems. International Series on Applied Systems Analysis. Wiley, Chichester, NY [4]

Casti J (1994) Complexification. Harper Collins, New York [4]

Cheremisinoff NP (1984) Fluid flow. Gulf, Houston, TX [4]

Dhillon BS (1983) Reliability engineering in systems design and operation. Van Nostrand Reinhold, Berkshire [3, 4, 5]

Dhillon BS (1999a) Design reliability: fundamentals and applications. CRC Press, LLC 2000, NW Florida [3]

Dhillon BS (1999b) Engineering maintainability. Gulf, Houston, TX [4]

Dubois D, Prade H (1988) Possibility theory—an approach to computerized processing of uncertainty. Plenum Press, New York [3]

Dubois D, Prade H, Yager RR (1993) Readings in fuzzy sets and intelligent systems. Morgan Kaufmann, San Mateo, CA [3]

Elsayed EA (1996) Reliability engineering. Addison-Wesley Longman, Reading, MA [4]

Emshoff JR, Sisson RL (1970) Design and use of computer simulation models. Macmillan, New York [4]
Fabrycky WJ, Blanchard BS (1991) Life-cycle cost and economic analysis. Prentice Hall, Englewood Cliffs, NJ [4]
Fodor J, Roubens M (1994) Fuzzy preference modelling and multicriteria decision support. Kluwer, Amsterdam [5]
Garey MR, Johnson DS (1979) Computers and intractability: a guide to the theory of NP-completeness. WH Freeman, New York [4]
Gertman DI, Blackman HS (1994) Human reliability & safety analysis data handbook, 1st edn. Wiley, Chichester, NY [5]
Goldberg DE (1989) Genetic algorithms in search, optimization & machine learning. Addison-Wesley, Reading, MA [5]
Goldratt EM (1990) What is this thing called the Theory of Constraints? North River Press, Croton-on-Hudson, NY [4]
Grant Ireson W, Coombs CF, Moss RY (1996) Handbook of reliability engineering and management. McGraw-Hill, New York [3]
Hicks CR (1993) Fundamental concepts in the design of experiments. Oxford University Press, Oxford [4]
Hill PH (1970) The science of engineering design. Holt, Rinehart and Winson, New York [4]
Hoover SV, Perry RF (1989) Simulation: a problem-solving approach. Addison-Wesley, Reading, MA [4]
INCOSE (2002) Systems engineering. International Council on Systems Engineering, Seattle, WA, Wiley, Chichester, NY [4]
Jardine AKS (1973) Maintenance, replacement and reliability. Wiley, Chichester, NY [4]
Kececioglu D (1995) Maintainability, availability, and operational readiness engineering. Prentice Hall, Englewood Cliffs, NJ [4]
Kepner CH, Tregoe BB (1981) The new rational manager. Princeton Research Press, Princeton, NJ [5]
Kletz T (1999) HAZOP and HAZAN: identifying and assessing process industry hazards. Institution of Chemical Engineers (IchemE) Warwickshire [5]
Klir GJ, Yuan B (1995) Fuzzy sets and fuzzy logic theory and application. Prentice Hall, Englewood Cliffs, NJ [3]
Law AM, Kelton WD (1991) Simulation modelling and analysis, 2nd edn. McGraw-Hill, New York [4]
Meyer MA, Booker JM (1991) Eliciting and analyzing expert judgment: a practical guide. Academic Press, London [3]
Michael J, Wood W (1989) Design to cost. Wiley, Chichester, NY [5]
Montgomery DC (1991) Introduction to statistical quality control, 2nd edn. Wiley, Chichester, NY [4]
Moore R (1979) Methods and applications of interval analysis. SIAM, Philadelphia, PA [3]
Naylor TH, Balintfy JL, Burdick DS, Chu K (1966) Computer simulation techniques. Wiley, Chichester, NY [4]
Neuts MF (1981) Matrix geometric solutions in stochastic models. Johns Hopkins University Press, Baltimore, MD [4]
Nikolaidis E, Ghiocel DM, Singhal S (2005) Engineering design reliability handbook. CRC Press, New York [3]
O'Connor PDT (2002) Practical reliability engineering, 4th edn. Wiley, Hoboken, NJ [3]
Oksendal B (1985) Stochastic differential equations: an introduction with applications. Springer, Berlin Heidelberg New York [5]
Pahl G, Beitz W (1996) Engineering design. Springer, Berlin Heidelberg New York [3]
Payne S (1951) The art of asking questions. Princeton University Press, Princeton, NJ [3]
Pecht M (1995) Product reliability, maintainability, and supportability handbook. CRC Press, New York [4]

Peterson JL (1981) Petri net theory and the modeling of systems. Prentice Hall, Englewood Cliffs, NJ [4]
Phadke MS (1989) Quality engineering using robust design. Prentice Hall, Englewood Cliffs, NJ [4]
Roberts FS (1979) Measurement theory. Addison-Wesley, Reading, MA [3]
Ryan M, Power J (1994) Using fuzzy logic—towards intelligent systems. Prentice Hall, Englewood Cliffs, NJ [3]
Sachs NW (2006) Practical plant failure analysis. A guide to understanding machinery deterioration and improving equipment reliability. CRC Press, London [3]
Shannon RE (1975) Systems simulation: the art and science. Prentice Hall, Englewood Cliffs, NJ [4]
Simon HA (1981) The sciences of the artificial. MIT Press, Cambridge, MA [3, 4]
Smith DJ (1981) Reliability and maintainability in perspective. Macmillan Press, London [4]
Smith DJ (2005) Reliability, maintainability and risk: practical methods for engineers, 6th edn. Elsevier, Oxford [4]
Stuart JR, Norvig P (1995) Artificial intelligence: a modern approach. Prentice Hall, Englewood Cliffs, NJ [5]
Taguchi G (1993) Robust technology development: bringing quality engineering upstream. ASME Press, New York [4]
Taguchi G, Elsayed E, Hsiang T (1989) Quality engineering in production systems. McGraw-Hill, New York [4]
Thompson WA (1988) Point process models with applications to safety and reliability. Chapman and Hall, New York [5]
Tong C, Sriram D (1992) Artificial Intelligence in Engineering Design. Vol 1. Design representation and models of routine design. Vol 2. Models of innovative design, reasoning about physical systems, and reasoning about geometry. Vol 3. Knowledge acquisition, commercial systems, and integrated environments. Academic Press, San Diego, CA
Vajda S (1974) Maintenance replacement and reliability. Topics in Operational Research. University of Birmingham, Birmingham [4]
Valluru BR (1995) Neural networks and fuzzy logic. M&T Books, IDG Books Worldwide, Foster City, CA [5]
Villemeur A (1991) Reliability, availability, maintainability and safety assessment. Wiley, Chichester, NY [5]
Warfield JN (2000) A structure-based science of complexity: transforming complexity into understanding. Kluwer, Amsterdam [4]

Index

A

ABD *see* availability block diagram
abstraction rule 115
accelerated life testing 715
accessibility 305
achieved availability 303, 355, 359, 387
acquisition costs 316, 318
activation function 712
actual degree of safety 653
AFIC *see* automatic fault isolation capability
AI *see* artificial intelligence
AIB *see* artificial intelligence-based
algorithm description
  using binary decision diagrams 695
algorithm-level description 726
algorithmic complexity 457
algorithmic knowledge 26
algorithmic modelling 142
alternative performance index (API) 113
ambiguity uncertainty 216
analytic model 425
ANN *see* artificial neural network
ANS *see* artificial neural system
API *see* alternative performance index
application modelling outcome 518
applied computer modelling 22
arbitrary nesting 482
artificial intelligence (AI) 3, 25
artificial intelligence (AI) language 28
artificial intelligence (AI) modelling 13, 330, 774
artificial intelligence (AI) system 592
artificial intelligence in design 21
artificial intelligence-based (AIB) blackboard 762
artificial intelligence-based (AIB) blackboard model 24, 242, 419, 422, 727
artificial intelligence-based (AIB) blackboard system 536
artificial intelligence-based (AIB) model 241, 486, 725
artificial intelligence-based (AIB) modelling 3, 11, 21, 22, 37, 107, 139, 415, 680
artificial intelligence-based (AIB) user interface 753
artificial neural network (ANN) 20, 485, 498, 592, 702, 703
  analysis capability 721
  back propagation 711
  building blocks 704
  computation 743, 748, 778
  computational architecture 722
  learning 709
  model 744
  model architecture 722
  structure 707
  training 718
artificial neural system (ANS) 13
artificial perceptron (AP) 707
assembly of components 16
assembly reliability 58
asymptotic behaviour 194
automated continual design review 22, 24, 25, 34, 774, 777, 790
automatic diagnostic systems 393
automatic fault isolation capability (AFIC) 393
automatic test equipment (ATE) 393
availability 5, 14, 18
  analysis 12
  analytic development 415
  application modelling 486

assessment 296, 349, 351, 436
basic relationship model 297
block diagram (ABD) 465, 466, 468, 469, 476, 478
cost modelling 308
cycle 345
evaluation 385
Petri net model 453, 454
prediction 296
specific application modelling 399
theoretical overview 302

B

back-propagation (BP) algorithm 711
back-up system 46
backward analysis 540, 565
backward chaining 766, 770
barrier analysis 553
basic structure of a rule 768
Bayes theorem 221, 222, 234, 235
Bayesian estimation 14
Bayesian framework 15
Bayesian method 215, 300
Bayesian model 148
Bayesian updating 230, 233, 235
BBMS *see* blackboard management system
BDD *see* binary decision diagram
behaviour model 702
behavioural knowledge 147
Benard's approximation 201
Benard's median rank position 200
benefit-cost ratio 322
Bernoulli distribution 231
Bernoulli probability distribution 75
Bernoulli transform 633
beta distribution 229, 236
characteristics 236
beta factor model 623, 624
bill of material (BOM) 270
binary decision diagram (BDD) 567, 573, 687, 695
safety valve selection 696
binomial distribution 104, 231
binomial method 73, 75
BIT *see* built-in testing
BITE *see* built-in-test-equipment
black box 704
black box CER 592
blackboard concurrent execution 782
blackboard data object 779
blackboard management system (BBMS) 13
blackboard model 11, 25, 29, 30, 34, 107, 241, 330, 334, 415, 421, 486–488, 678, 680, 724, 725
artificial intelligence-based (AIB) 726
context 491
dynamic systems simulation 493
systems selection 489
user interface 491
blackboard system 682, 780
blackboard systems design
formalised model 778, 779
performance analysis 780
block diagram 466
Boolean disjunction operation 175
Boolean expression 643
Boolean function 710
Boolean operator 764
Boolean reduction 574
Boolean truth tables 232
bottleneck 343, 427, 473
boundary condition event tree 563
branched decision tree 765
break-even discount rate 323
broad-brush analysis 79
built-in or non-destructive testing 391
built-in-test-equipment (BITE) 391
built-in testing (BIT) 304, 360, 389, 391, 393
design 397
performance 394
system
evaluation 398

C

CAD *see* computer-aided design
calculated system unavailability 648
capability 327
capability index 330, 333
capacity 20
capital costs 4, 309
capital spares 381
cash operating costs 4
causal analysis 529, 540
causal factor analysis 553
cause-consequence analysis (CCA) 543, 565, 567, 587, 634
cause-consequence diagram (CCD) 565, 567, 642, 643
construction 570, 645
quantification 568
symbols 568
symbols and functions 569
CCA *see* cause-consequence analysis

CCD *see* cause-consequence diagram
centralised control 458
certain loss 596, 598
certainty rule 165
change analysis 553
Chapman–Kolmogorov equation 611
characteristic life 227
Chi-square distribution 15
classification problem 747
classifications of failure 540
closed mode probability 106
closed system 461
clustering problem 746
collaborative design 679
collaborative engineering design 22, 261, 416, 419, 428
collective identity 16
combination fault tree 646, 647
common cause failure (CCF) 622
  engineering causes 622
  operational causes 622
common failure mode 77, 757, 758
common mode failure (CMF) 621
common root cause analysis 553
complete functional loss 176
complex 476
complex fuzzy rule 156
complex logical test 768
complex system 458
  complicatedness 481, 483
  counteraction results 461
  increased automation 533
  interdependency 461
  safety analysis 537
complex systems theory (CST) 456
complexity logistic function 484
component failure density 670
component failure mode 137
component failure rate λ_p 86
component functional relationship 136
component level 44
component reliability 58
computational complexity 458
computer-aided design (CAD) 38, 329, 741
conceptual design 7, 45, 107, 332
conceptual design optimisation 112
conceptual design performance prediction 60
conceptual design phase 535
conceptual design reliability 60
conceptual design review 301
conceptual design safety and risk prediction 588, 678
conceptual design solution 682
conceptual effort 63
concurrent design 22
concurrent engineering design 107, 679
concurrent execution 787
condition diagnostics 262
condition inspection 365
condition measurement 365
condition monitoring 364
condition screening 365
condition worksheet 263
conditional probability 221, 564
conditional reliability 96, 670
conditional survival function 96, 672
conditions description 784
conditions failure 784
confidence level 14, 195
confidence method
  managing uncertain data 772
confidence value 763, 773
conjunction-based fuzzy rule 166
consequence analysis 529, 530, 540
consequences of failure 18, 271
constant demand rate 382
constant failure rate 74, 89, 382
constant hazard rate 67
constraint-based technique 684
constraint label 114
constraint propagation 39, 113
constraints evaluation 472
constructability 329
construction costs 64
continuous monitoring 364
continuous-time Markov chain (CTMC) 439, 443, 447
continuous-time simulation model 426
contract spares 380
control panel 30
control shell 490
control software design 534
control systems engineering 800
corrective action 299, 362
corrective maintenance action 19
corrective maintenance costs 376
corrective maintenance time 396
  lognormal distribution 359
cost
  blow-outs 9, 34
cost critical item 243
cost criticality analysis 662
cost driver 593
cost effectiveness (CE) equation 325
cost efficiency ratio 368
cost estimating
  pitfalls 65

cost estimating relationship (CER) 586, 590
development 593
multiple regression 593
cost of dependency 310, 312
cost of loss 654
cost optimisation curve 657
cost optimisation modelling 360
cost risk 655
critical design review 301
critical failure 652
critical risk 610
critical risk theory hypothesis 610
criticality analysis 135, 786
cross validation dataset 747
crossover breeding operator 693
CST *see* complex systems theory
cumulative distribution function 91
cumulative sum charting method 717
cusum charting procedure 721
cut-off probability method 622

D

damage risk 584
data point generation 72
data-directed invocation 39
database analysis tool 244
DCF *see* discounted cash flow
de-bottlenecking 662
decision logic 759
deductive analysis 543
deductive validity 168
defect maintenance 363, 369, 372
defects risk 584
delayed fatality 614
delta learning rule 710, 711
demand 20
dependability modelling 385
dependent demand maintenance spares 382
DES *see* domain expert system
design assessment 784, 790
design assistance 38
design automation (DA) 33, 38, 740
design basis event 677
design calculation check 421
design capacity 310, 335, 400
design checklist 419
design complexity 4
design cost risk analysis 586
design criteria 3, 9, 763, 784
design definition 535
design dictate 307
design effectiveness (DE) 326
design effort 63
design engineer
scope of work 799
design integrity *see also* engineering integrity, 172, 327, 370
automation 33
development and scope 12
methodology 3
uncertainty 18
design intent 577, 741
design knowledge
base 487, 681
source 487, 681
design-level FMEA 79, 757
design model
development programming 498
design optimisation 681, 689
designing for safety 617
design problem 459
definition 462
design process 29
integration with blackboard models 726
design reliability
total cost models 60
design representation 576
design review 7, 9, 21, 24, 301, 420
design space 22, 679
design specification 784
design specification FMECA 281
design synthesis 9
design to cost (DTC) 590, 591
design tool 28
design variable 31, 145
design verification 10, 142
designing for availability 18, 309
using Petri net modelling 453
designing for maintainability 19, 296, 309, 358
designing for reliability 16, 43, 69, 72, 296, 297
labelled interval calculus 123
designing for safety 20, 134, 531
cost risk models 588
critical risk theory 614
design optimisation 617
genetic algorithm 21
Markov point process 608
point process event tree analysis 627
profile modelling 738
requirements 628
detail design 11, 17, 90, 146, 332, 385
detail design model 684
detail design phase 535
detail design plant analysis 24
detail design reliability evaluation 190

detail design review 301
detail design safety and risk evaluation 627, 702
deterministic analysis 676
deterministic knowledge 775
deterministic safety analysis approach 677
deviation analysis (DA) 544
device performance index (DPI) 418
digital prototyping 742
digraph 543
discounted cash flow (DCF) 322
discrete event system (DES) 604
discrete-event simulation model 426
diseconomies of scale 344
disjunction 175
disorder independence 177
distributed control system (DCS) 242, 256, 272, 599, 616, 645
domain expert system (DES) 13, 27, 606
downtime 299, 403, 405
DPI *see* device performance index
Drenick's theorem 383
DTC *see* design to cost
durability 301
dynamic data exchange (DDE) capability 498
dynamic penalty function 692, 693
dynamic programming 689
dynamic systems simulation 492, 502
dynamic systems simulation blackboard model 487, 518
dynamic systems simulation modelling 10, 486, 736
dynaset 244, 246

E

early failure 92
economic loss 310, 312, 324
economic optimum reliability 60
economy of scale 343, 344
EDA *see* evaluation design automation
effective capacity 335
effective discount rate 322
effective maintenance 367
effectiveness 296
effectiveness measure 471
effects analysis 276
effects of failure 16
efficiency 76
efficiency measurement 337
elimination condition 117
emergency shutdown (ESD) system 560
engineered complexity 485
engineering design
  analysis
    concept of uncertainty 145
    incompleteness 173
    uncertainty 173
  analytic development of safety and risk 676
  application modelling of safety and risk 725
  artificial neural networks 715
  complexity 460
  complicatedness 480
  effort 63
    management review 64
  evaluating complexity 480
  flexibility 488
  integrity 3, 5
  intolerable risk 530
  negligible risk 531
  project management expert systems 28
  risk 529, 535
  safety 529, 537, 551
  tolerable risk 530
engineering language 6
environment risk 584
environmental protection 6
equal strength principle 111
EQUIPID 244, 246
equipment
  burn-in period 92
  failed state 404
  hazard curve 654
  maintainability 372
  operational condition 372
  potential usage 371
  survival curve 654
  useful life period 92
  wear-out phase 93
equipment age analysis 651, 670
equipment aging model 73, 77
equipment availability 371
equipment condition 361, 756–758
equipment criticality 8
equipment failure 20, 581
equipment failure mode 79, 137
equipment FMEA 79
equipment listing 246
  at assembly level 250
  at component level 250
  at system level 249
equipment maintainability 88
equipment protection 6, 652
equipment reliability 16, 371

equivalent availability (EA) 400–402, 413, 414
  change 410
equivalent maintainability measures
  downtime and outage 403
equivalent mean time to outage 405
equivalent mean time to restore 406, 407
equivalent operational time 401
ergonomics 304
error back propagation 709
error-prone automation feature 535
establishment costs 319
estimated degree of safety 653
estimating failure rate 198
estimation 502
estimator 196
  consistent 197
  unbiased 197
evaluation design automation (EDA) 33, 38
event 178, 190
event tree
  boundary condition 563
  conditional probability 560
  construction 557
  evaluation 562
  fault-tree linking 564
  quantitative assessment 560
  RBD 641
event tree analysis (ETA) 543, 554, 568, 634
evolutionary algorithm (EA) 496, 678, 685
evolutionary computing 681
evolutionary computing technique 686
evolutionary design 146, 681
execution policy 442
EXP transition 444
expected availability 408
expected maximum corrective maintenance downtime 359
expected performance 20
expected useful life 613, 615
expert judgement 214, 215, 228, 234, 728
expert system 27, 28, 728, 777
  branched decision tree 769
  framework 173
  models 217
  multiple-choice question editor 767
  rule-based 29
  rule editor 771
  rules of the knowledge base 770
  shell 29
  tool 148
  user interface 762
exponential distribution
  estimating the parameter 200
exponential failure distribution 90, 93, 198
exponential probability density function 198
ExSys© Expert System 765, 777
Extend© 486
Extend© ModL language 497
Extend© Performance Modelling 495, 511
extended FMECA 179, 190
  uncertainty 180
extended reachability graph 445
external uncertainty 428–430
extreme condition approach 428, 429, 434

F

fabrication costs 64
facts frame 760, 761
failed state 404
failure analysis 12
failure cause 138, 141
failure consequences 140, 541
  severity 666
failure cost criticality 272
failure criticality ranking 272
failure data analysis 282
failure definition and quantification (FDQ) 46
failure density 379
failure density function 670
failure detection 138
failure detection ranking 81
failure distribution 93
failure distribution function 632
failure effect probability guideline value 84
failure effects 138, 140, 541
failure elimination analysis (FEA) 47
failure hazard analysis (FHA) 135, 141
failure identification 786
failure logic diagram 733
failure mode 138, 139, 785
  critical number 82
  discriminability 179
failure mode occurrence probability 81
failure mode proportion α 86
failure modes and effects analysis (FMEA) 7, 34, 73, 78, 135, 137, 260, 262, 397, 755, 757
  advantages and disadvantages 80
  algorithmic modelling 142
  modelling uncertainty 174
  steps for performing 80
  types and benefits 79
  worksheet 85
failure modes and safety effects (FMSE) 650, 667

process criticality using residual life 674
qualitative risk-based 668
sensitivity testing 673
failure modes effects and criticality analysis (FMECA) 7, 34, 47, 80, 134, 229, 260, 650, 657, 757
analysis 774
cost criticality 663
data sources and users 84
expression of uncertainty 178
logical expression 175
modelling uncertainty 174
preventive maintenance activities 659
process and cost criticality 665
process criticality 658
uncertainty 18, 188
worksheet 85
failure occurrence likelihood 666
failure of equipment 45
failure operational consequences 651
failure pattern 227
failure physical consequences 651
failure probability (FP) 83, 93, 549, 648, 671
failure rate 228, 345
failure rate function 97
failure replacement 379
false alarm rate (FAR) 621, 626
FAP *see* fuzzy artificial perceptron
FAR *see* fatal accident rate, *see* false alarm rate
fatal accident rate (FAR) 560
fault graph 543
fault tree 735
diagram 731, 734
dormant failure 620
linking 563
probability evaluation 550
quantification 573
RBD transformation 640
select event 694
transformation 640, 641
fault-tree analysis (FTA) 34, 73, 86, 236, 541, 542, 552, 565, 568, 587, 616, 634, 687, 694
logic and event symbols 546
safety and risk assessment 90
safety systems design 615
steps 88
FBC *see* feature-based costing
FDQ *see* failure definition and quantification
FEA *see* failure elimination analysis
feasibility study 799
feature panel 30
feature-based costing (FBC) 591
feed-forward ANN 718
feed-forward network 705, 706
final detail design 7
firing policy 442
firing time 441, 454
first cost curve 61
first cost estimate 62
fitness value 698, 700, 701
flow capacity 474, 475
FMEA *see* failure modes and effects analysis
FMECA *see* failure modes effects and criticality analysis
FMSE *see* failure modes and safety effects
formal elicitation 214
forward analysis 540
forward chaining 771
frame name 761
frame slot 761
frame-based knowledge 38
FTA *see* fault-tree analysis
full outage 409, 413
function
complete loss 71
definition 71
partial loss 71
function approximation problem 746
functional analysis 464
functional block diagram (FBD) 135, 136, 138, 466
functional effectiveness 337, 423
functional event tree 556
functional failure 17, 70, 71, 134, 139, 141, 257, 362, 378
physical consequences 652
safety operational consequences 652
functional FMEA 78
functional knowledge 147
functional performance 17, 71
functional performance limit 70, 72
functional relationship 135
functional specialisation 789
functional systems breakdown structure (FSBS) 135, 136
functions analysis 256, 784
functions description 785
fuzzification 144
fuzziness of probability 148
fuzzy ANN modelling 720
fuzzy artificial perceptron (FAP) 714
fuzzy Euler integration 144
fuzzy fact 158
fuzzy implication 164
fuzzy inference 153
fuzzy interval 144

fuzzy judgment 224, 230, 239
reliability evaluation 225
fuzzy knowledge 147, 157
fuzzy logic 158, 161, 216, 217, 773
fuzzy logic expert system 775
fuzzy membership function 163, 773
fuzzy neural rule-based system 713
fuzzy pre-processing technique 721
fuzzy preference 679
fuzzy reasoning 158, 165
fuzzy rule 153, 154
fuzzy set 18, 52, 147, 149, 151, 159, 216, 217, 220, 476, 714
intersection 714
theory 148, 150, 218
fuzzy simulation 144
fuzzy system 240

G

gamma distribution 15, 228, 287
general algorithm (GA)
methodology 701
parameter 701
general law of addition 50
generalised modus ponens (GMP) 164, 167
genetic algorithm (GA) 20, 411, 590, 678, 686, 687, 690, 696, 748, 750
implementation 697
natural selection 697
optimal safety system design 687
genetic operator 698
geometry panel 30
global contribution 157
goodness-of-fit results 284
goodness-of-fit test 283, 502
gradient descent technique 710
gradual rule 165
graphical user interface (GUI) 742

H

HazAn *see* hazards analysis
hazard and operability study (HazOp) 599
hazard consequences 540
hazard identification (HAZID) 537, 538, 547, 582
qualitative modelling 605
hazard rate 613
hazard rate curve 92, 227
hazard rate function 90, 91
hazard severity 539
hazard-contributing factor 558
hazardous operations (HazOp) 7, 34, 544, 545, 575, 604
hazardous operations (HazOp) assessment 784
hazards analysis (HazAn) 7, 34, 529, 530, 537, 541, 582, 587
hazards criticality analysis 263, 264
condition spreadsheet 264
costs spreadsheet 268
costs worksheet 268
criticality worksheet 265
logistics spreadsheet 270
logistics worksheet 269
strategy worksheet 266
hazards definition 535, 576
HAZID *see* hazard identification
HAZOP *see* hazardous operability studies
HazOp *see* hazardous operations
secondary keyword 601
HAZOP study 577
consequences 581
process parameter 578
safeguard 581
HazOp study
methodology 601
primary keyword 600
secondary keyword 600
health risk 584
health status and monitoring (HSM) 304
hedge 151
heuristic knowledge 27, 29
hierarchical frame 762
high-integrity protection system (HIPS) 619, 625, 638, 687, 690
cause-consequence diagram 649
component functions 644
control valve 270
higher-order uncertainty 172
HIPS *see* high-integrity protection system
holding ability 334
Holland's fixed-length coding 687
house event 619, 621
human error 581
human error analysis 534
human factor 533
human factor analysis 535
human–machine interaction 534
human performance evaluation 553
hypothesis testing 501, 502, 673

I

IIT *see* information integration technology
implication-based fuzzy rule 165

incidence matrix 477
incompleteness 15
independent demand maintenance spares 382
indeterminate rate of return 325
inductive analysis 543
industry perception 34
information integrated technology (IIT) 624
information integration technology (IIT) 18, 214, 346, 348
inherent availability 303, 344, 346, 387
  exponential function 345
inhibitor arc 441
initial failure rate estimate 586
initial operational test and evaluation (IOT&E) 399
initiating event 556
installation costs 64
instantiation parameter 494, 738
integrated information technology (IIT) 630
integrity engineering design 3
integrity prediction 420
intelligent computer automated methodology 12
intelligent design system 37
intensity function 610, 613
interaction and feedback loops 458
interaction model taxonomy 493
interchangeability 305
interference theory 65
internal rate of return (IRR) 322–324
internal uncertainty 428, 430
inter-process communication (IPC) 498
interval matrix 130
inventory control 380
IPAT SO3 cooler 275
IRR *see* internal rate of return
item criticality number 84

J

job safety instruction (JSI) 603
judgment bias 222
jump connection back propagation 722

K

k-out-of-*m* unit network 104
Kaplan–Meier estimator 202
Kaplan–Meier survival curve
  rotating equipment 655
kinetic energy 342
knowledge base 766
knowledge-based decision process 624
knowledge-based expert system 11, 22, 25, 26, 34, 37, 107, 330, 334, 415, 419, 486, 678, 717, 752, 754
  testing and validating 771
knowledge engineer 27, 682
knowledge engineering 26, 703
knowledge-level specification 726
knowledge source 11, 30, 488–490, 768, 776, 779, 780
  connectivity analysis 778
  interdependence 778, 782, 790
  serialisation 778, 781, 790
  specialisation 778, 781, 787
  specialisation value 780
knowledge training 742
Kohonen self-organising map 724
Kolmogorov backward equation 611
Kolmogorov differential equations 610, 613
Kolmogorov forward equation 611
Kolmogorov's theorem 703
Kolmogorov–Smirnov (K–S) test 283

L

labelled interval 130
labelled interval calculus (LIC) 17, 112, 113, 123
labelled interval inference 115
Laplace transform 75, 89, 354
Latin hypercube sampling technique 429
law of multiplication 48
laws of probability 52
LCC *see* life-cycle costs
Lebesgue logic 220
level of diversity 617
level of redundancy 52, 617
LIC *see* labelled interval calculus
  inference rules 124
life-cycle analysis 314, 315
life-cycle costs (LCC) 309, 314, 316
  present value calculations 321
  trade-off measurement 325
life risk 584
likelihood function 222, 223
limit of capability 416
limit theory 383
linguistic variable 150, 159
  translation rule 160
logic diagram 733
logical flow initiation 503
logical flow storage 504
loss in production 310
loss-less transformation 714
loss of function 139, 403

loss risk 584
lower limit interval 128
lower tolerance limit (LL) 507, 509, 512, 517

M

maintainability 5, 14, 19, 298
  analysis 12, 299, 304, 306
  analytic development 415
  application modelling 486
  assessment 349, 356, 436
  checklist 422
  cost indices 392
  cost modelling 308
  design review 19, 301
  evaluation 385, 391
  evaluation indices 391
  function 347
  measures 358
  modelling 300
  score 306
  specific application modelling 399
  theoretical overview 302
maintenance
  assessment 358
  basic principles 361
  cost optimisation modelling 375
  modelling 356
  practice 67
  ratio (MR) 392
  spares
    dependent demand 381
    independent demand 381
  strategy 360, 367, 368, 372, 377, 657
management oversight and risk tree (MORT) analysis 553
manpower costs 376
manufacturability 328
mapping 160
marking 438
  tangible state 444
  vanishing state 444
marking-dependent arc multiplicity 441
Markov chain 610, 613
Markov modelling 73, 349, 350, 543
Markov point process 608
Markov regenerative process (MRGP) 452
Markov reward model 451
Markovian stochastic Petri net (MSPN)
  definition 443
  measures 449
mass-flow balance 340, 341
mass-flow rate 339
mathematical model 10, 338, 350
  preventive maintenance physical checks 365
  preventive maintenance replacement costs 377
  preventive maintenance replacement shuts 366
  spares requirement 382
maximum dependable capacity (MDC) 401, 406, 412, 471
maximum likelihood 14, 223
maximum likelihood estimation (MLE) 193, 194, 203, 348
  parameter estimation 193
maximum likelihood ratio test 224
maximum-likelihood technique 76
maximum limit interval 124
maximum process capacity 412
maximum safety margin 17
maximum time to repair (MaxTTR) 304, 391
MDT *see* mean downtime
mean downtime (MDT) 18, 389, 403
mean expected loss risk (MEL-risk) 595, 597
mean residual life (MRL) 672
mean squared error (MSE) 750
mean time between failures (MTBF) 18, 211, 478, 662, 671
mean time between maintenance actions (MTBMA) 392
mean time for maintenance 357
mean time to fail (MTTF) 94, 97, 379, 672
mean time to repair (MTTR) 18, 300, 304, 391, 403, 406, 478
measure of performance 370
measure of probability 652
median rank 201
membership function 151, 217, 218, 223, 225, 240
  probability measures 219
memory policy 442
military standard technique 82
minimal cut set (MCS) 548
minimal network 748
minimum limit interval 125
MLE *see* maximum likelihood estimation
  normal distribution 195
MLP *see* multi-layer perceptron
model
  component 518
  configuration 494, 738
  functional behaviour 500
  scripting 498

structure uncertainty 428
validation 500, 501
verification 500, 501
modelling result, evaluation 271, 776
modular architecture 494
interface connection 494
object connection 494
modus ponens 163
modus tollens 163
moment matching method 435
Monte Carlo (MC) simulation 15, 230, 232, 286, 300, 302, 416, 432, 433, 731, 733, 735
MTBF *see* mean time between failures
MTTF *see* mean time to fail
MTTR *see* mean time to repair
multi-layer perceptron (MLP) 706
weight matrix 706
multi-layered network 703
multi-state Markov model 351, 353
multiple expert system 762
multiple logical flow 737
mutation operator 693

N

net present value (NPV) 322
network complexity 749
network diagram 731, 732, 734
neural expert program 725, 743
neural network 411, 678
iterative prediction 747
NeuralExpert© program 744, 750
non-destructive test (NDT) 365, 391
non-Markovian marking process 452
non-Markovian stochastic Petri net
definition 451
non-Markovian system 352
non-recurring costs 63
normalised mean squared error (NMSE) 751
NPV *see* net present value
nuclear power plant 77
numerical analysis 142

O

OA *see* optimisation algorithm
object-oriented programming (OOP) 21, 486
encapsulation 727
inheritance 727
simulation model 21, 23, 541
occupational safety and health (OSH) 532
occurrence probability 84
off-system maintainability indices 392
OOP *see* object-oriented programming
open mode probability 106
open system 461
operability analysis 587
operating costs 309
operating environment 67
operational availability 303, 355, 387, 400
time-line model 389, 390
operational condition 423
operational failure rate λ_o 86
operational integrity 370, 386
operational modelling 385
operational risk analysis 586
operational time 401
operator control panel (OCP) 550
OPI *see* overall performance index
optimisation algorithm (OA) 10, 415, 680
Petri net (PN)-based 514
optimisation capability 496
optimisation module 681
order of magnitude 143
OSH *see* occupational safety and health
outage 403, 405
measurement 408
output conversion function 504
output performance results 505, 511, 514
output set overlap 780
overall performance index (OPI) 113, 131, 133

P

parallel configuration 50
parallel network 103, 105
parallel reliability block diagram 467
parameter performance index (PPI) 130, 132, 417, 418
parameter profile index (PPI) 113
parameter profile matrix 108, 112, 338, 417, 421
parametric cost estimating (PCE) 592
parametric estimating (PE) 590
Pareto principle 243, 667, 680
partial functional loss 176
partial loss of system function 409
partial outage 409, 413, 415
partial redundancy 617
partial state matrix 413
PDS *see* procedural diagnostic system
PEM *see* process equipment model
holding tank 739
penalty formula 698
penalty function 699

people risk 584
percent error 752
performance 16, 35, 43, 70
performance and reliability evaluation with diverse information combination and tracking (PREDICT) 214
performance assessment 783, 790
performance distribution
  statistical approach 435
performance measure 31
performance specification 783
performance variable 31
periodic monitoring 364
personal protection 6, 652
perspective 22
Petri net (PN) 19, 436, 437, 745
  definition 439
  graphical representation 440
  model
    numerical computations 453
    steady-state solution 454
  reachability graph 445
  theory 437
  transition 451
Petri net-based optimisation algorithm 740, 744
Petri nets and performance models (PNPM) 437
PFD *see* process flow diagram
PHA *see* preliminary hazard analysis
phenomena event tree 556
physical design factor 307
pipe and instruments diagram (P&ID) 45, 264, 303, 575, 605
plant analysis 773
point of reference (POR) 580
point process 608
  intensity function 609
point process analysis 587
point process consequence analysis 630
point process event tree analysis 627
Poisson demand 384
Poisson distribution 15, 67, 231, 383, 560, 561
Poisson process 94, 300, 630
POR *see* point of reference
possibilistic knowledge 775
possibilistic logic
  generalised modus ponens 178
possibility distribution 151
possibility rule 166
possibility theory 16, 18, 169, 216, 220, 347
  deviation from fuzzy logic 170
  engineering design analysis 172
post-design testing and training 742
potential energy 342
potential failure 141, 362
potential risk 676
PPI *see* parameter profile index, *see* parameter performance index
predictable behaviour 458
prediction problem 746
predictive maintenance 364
preliminary 73
preliminary design 135
  safety and risk assessment 607, 687
preliminary design phase 535
preliminary design process analysis 24
preliminary hazard analysis (PHA) 539
preliminary hazards identification (PHI) 607
preventive action 362
preventive maintenance 344, 363, 369, 436, 455
preventive maintenance policy 355
preventive maintenance program 358
preventive maintenance strategy 378
preventive replacement modelling 378
probabilistic analysis 676
probabilistic knowledge 775
probabilistic reasoning 171
probabilistic risk analysis (PRA) 635
probabilistic safety evaluation (PSE) 627, 628
probability density function 91, 93, 193, 199, 345
probability distribution 14
probability distribution definition 675
probability function 225
probability generating function 633
probability law 52
probability of failure 20, 210
probability of failure consequence β 86
probability of survival 210
probability plotting 200
probability qualifier 666
probability theory 216, 347
probable loss 596, 598
problem analysis 501
procedural diagnostic system (PDS) 13
process analysis 13, 21, 23
process block diagram 479
process capability 328, 331, 386, 423
process capability model 330
process capacity 334
  measuring 335
process critical item 243
process criticality 8
process definition 31, 783

process description 783
process design 800
process design blackboard section 786
  functional independence 791
  functional specialisation 791
process design criteria 8
process design specifications 510, 514
process effectiveness 337, 471
process engineering 800
process equipment model (PEM) 10, 241, 439, 486, 503, 504, 510, 513, 713, 725, 737
  logical flow 495
  logical flow storage 504
  model component 503
process failure consequences 8
process flow block diagram 464, 466, 468
process flow diagram (PFD) 8, 45, 250, 251, 264, 303, 605, 736, 737, 754
  sector 1 503
  sector 2 509
  sector 3 513
process flow rate 339
process hazard identification (PHI) 599
process industry 4
process level 44
process-level FMEA 79
process operational risk modelling 594
process parameter 578, 580
process reliability 8
process risk 584
process simulation model 488, 493
process stability 333
process utilisation 338
process view 332, 333
processing element (PE) 704, 749
procurement costs 64
product assurance 6, 21
product risk 584
product yield 336
productive capability
  efficiency measurement 337
productivity 337
productivity ratio 368
profitability index 322
programmable logic controller (PLC) 273, 274, 599, 616
project cost estimation 62
project execution plan 805
propagation rule 121
proportional hazards (PH) model 191, 193
  non-parametric model formulation 191
  parametric model formulation 192
  reliability function 193
propositional logic 161
PSE *see* probabilistic safety evaluation

Q

Q-matrix 612
qualitative analysis 12, 16
qualitative assessment scale 666
qualitative cost estimating 592
qualitative criticality analysis 667
qualitative FMECA 178, 189
qualitative parameter estimation 194
qualitative simulation 143
quantitative analysis 12
quantitative maintainability analysis 19
quantitative review 420
queuing theory 300

R

RA *see* risk analysis
RAM assessment 783
RAMS analysis 3, 6, 10
RAMS analysis list 251, 258
RAMS analysis model 21, 23, 241, 242, 486, 725
RAMS program 373
  principles 374
RAMS study 657
random failure 77, 94
random failure occurrence 613
random failure test 285
rapid risk ranking (RRR) 539
rated capacity 335, 400
Rayleigh distribution 204, 208
RBD *see* reliability block diagram
RCA *see* root cause analysis
reachability analysis 606
  checking safety 607
reachability graph 445, 452, 542
reachable markings
  distribution of the tokens 447
reactor safety study 630
receiving ability 334
recovery costs 320
recovery time 390
recurrent back-propagation 722, 723
recurrent network 704
recurring costs 63, 64
reduced efficiency 399, 400
reduced reachability graph 445, 447
redundancy 15, 56
redundancy allocation problem 689, 691
  objective function 692

redundancy condition 118
relative lost time cost 311, 312
relative value of dependency 311
reliability 5, 14, 35, 43
reliability analysis 12, 46, 654, 676
reliability application modelling 241
reliability assessment 44, 45, 69, 72, 86, 106, 133, 174, 560
reliability Bayesian evaluation 233
reliability block diagram (RBD) 466, 634, 635
  parallel configuration 467
reliability checklist 422
reliability-critical item 134
Reliability Enhancement Methodology and Modelling (REMM) project 551
reliability evaluation 44, 45, 69, 90, 106
  fuzzy logic 217
  fuzzy set 217
  three-state device networks 105
  two-state device networks 102
reliability function 91
reliability index 691
reliability initial calculation 230
reliability modelling 65
reliability of a component 47
reliability of a system 47
reliability prediction 44, 45, 68, 106, 110
reliability system-level 226
reliability theory 670
reliability uncertainty 239
reliable life 96
remote terminal unit (RTU) 274
renewal theory 383
repair action 19, 299
repair rate 88
replacement costs 309
replacement policy 379
replacement-power costs 309
reproduction probability 700
requirements analysis 464
residual life 96, 672
residual life evaluation 651, 670
residual risk 676
reuse 23
Reynolds number 341
risk
  actual severity 653
  estimated severity 653
  verification 536
risk analysis (RA) 47, 546, 582
  decision criteria 662
risk assessment 536, 804
risk assessment scale 585, 667
risk-based maintenance 655, 661
risk cost analysis 593
risk cost curve 61
risk cost estimation 60
risk equation 594
risk estimation 536, 582, 583
risk evaluation 785
risk hypothesis 594
risk identification 785
risk measure 595
risk of failure 20
risk priority number (RPN) 582
risk priority number (RPN) technique 80
robust design (RD) 329, 416, 419, 428, 429, 434, 436
root cause analysis (RCA) 47, 542, 551, 552, 587
  common cause failures 621
  safety 551
routine maintenance 363, 369, 372
RRR *see* rapid risk ranking
rule editor 767
rule-based expert system 759
  multiple-choice question editor 764

S

safety 6
  actual degree 584
  estimated degree 583
safety analysis 534, 537, 565
safety consequences 559
safety criticality 530
safety criticality analysis 650, 651, 654, 661
safety criticality rank 586
safety engineering 532
safety function 557, 558
safety intent specification 531
safety margin 20, 31, 67, 71, 72, 108, 416
safety protection system 616
safety risk 655
safety system 89, 688
safety systems design, cause-consequence analysis 634
safety systems, assessment with FTA 619
satisficing 23
SBS *see* systems breakdown structure
SCADA system 274
scale parameter 227
schematic design 7, 11, 73, 682, 729
schematic design review 301
scripting 498
SEA *see* systems engineering analysis
sector 1, simulation output 508

sector 3, simulation output 520
select event 694
selected equipment specifications 254
sensitivity testing 673
series configuration 50
series formula of reliability 54
series network 102, 106
series reliability 48
serviceability 300
set label 114
set-point control 273
shell 28, 38
sigmoid function 709
simplex 476, 477
simulation 230
simulation analysis 12
simulation model 384, 416, 423, 425, 427
  output 499
  sector 1 506, 508
  sector 2 509, 512
  sector 3 513, 515, 520
single failure mode 177
sizing design capacity 343
software deviation analysis (SDA) 544
solution encoding 691
spares requirements planning (SRP) 380
specification costs 319
specifications worksheet 260
square symmetric matrix 618
SRP *see* spares requirements planning
standard back propagation 722
standard deviation 211
standard work instruction (SWI) 603
standby redundant system 105
state matrix 412, 413
state probability 448
statistical approach 428, 429
statistical model 702
statistical technique 14
steady-state availability 351
stochastic optimisation technique 690
stochastic Petri net (SPN) 438, 441
stochastic point process 630
stochastic reward net (SRN) 451
stochastic system 384
stress/strength interference diagram 66
string fitness 698
sufficiency 76
sum squared error (SSE) 719
super-projects 4, 9
supervised learning 716
supervised learning paradigm 722
supervised training 717
supervisory control and data acquisition (SCADA) 273
supplementary variable 352
supportability 301
sustaining costs 316, 318
synthetic fault insertion 399
system analysis 12, 23
  with GAs and fault trees 694
system availability 30, 449, 455
system boundary 463
system breakdown structure (SBS) 8, 47, 61, 69, 72, 88, 134, 135, 138, 243, 246, 397, 607, 627, 728, 729, 762
system complexity 457, 480
system component 464
system composition 494
system configuration 463
system definition 784
system dependency 310
system design blackboard section 786
  specification 788
system effectiveness (SE) 325, 327, 388
system engineering 456, 459
  complexity 460
system engineering analysis (SEA) 69, 411, 456, 457, 460, 462
system event tree 556
system failure 353, 562, 632
  quantification 571
system failure effect 8
system hazard analysis 534
system hierarchical modelling 541
system hierarchy 70, 78
system integrity 478
system-level FMEA 79
system-level reliability 226
system life-cycle analysis 315, 551
system modelling option 729
system objective 463
system operability 342
system output
  deviation 432
  nominal value 432
system performance 134, 145, 328, 342
  prognosis 44
system performance analysis 416, 423, 424
system performance index (SPI) 111, 130, 132
system performance measures 108
system performance model 425
system performance sensitivity 703
system procedures blackboard section 786
system reliability 16, 46, 134, 449, 637
  effect of redundancy 55

system safety 533
system simulation option 739
system state space 74
system success 562
system transition diagram 74
system unreliability 46, 623

T

T-conorm function 715
t-norm operator 168
Taguchi's methodology 329
Taguchi's orthogonal arrays technique 429
Taguchi's robust design 429, 434, 704
target engineering design project 21
tautology 162
Taylor series 618, 621
technical specification document 253
test equipment 305
test point 305
testability 301, 305
theory of constraints (TOC) 343
three-parameter beta distribution function 237
three-parameter Weibull distribution 209
three-parameter Weibull fit 285
threshold logic unit (TLU) 708
threshold of chaos 457
throughput capacity 595, 597
  hazard-free 598
time before failure (TBF) 286
TOC *see* theory of constraints
total energy balance 341, 342
total loss of system function 409
total preventive maintenance 355
total system cost
  objective function 699
trade-off matrix technique 478
traditional cost estimating 588
transition priority 441
translation rule 121
truth table 162
truth value 149
two-state Markov model 349, 353

U

unavailability 301, 408
unavailability profile graph 735
uncertainty 15, 146, 153, 216
uncertainty analysis 428
  extreme condition approach 430
  statistical approach 432
uncontrolled process
  quantitative representation 606
universal approximation 703
universe of discourse 150
unreliability 46, 54, 301
  consequences 51
unsupervised learning 716
unsupervised network 722
unsupervised neural network 724
updating process 235
upper limit interval 127
upper tolerance limit (UL) 507, 509, 512, 517
useful life expectancy 613
  survival function 614
utilisation costs 320
utilisation factor 450
utilisation rate 388

V

value engineering 804
value of the system 326
vertex 476, 477
virtual prototyping 492, 736
volumetric energy 342
volumetric flow rate 340
voting redundancy 621

W

Ward back propagation 723
WBS *see* work breakdown structure
Weibull analysis 735
Weibull cumulative failure probability graph 737
Weibull density function 99
Weibull distribution 15, 192, 285, 485, 672
  function 100
  standard deviation 212
  statistical properties 98
Weibull distribution model
  expansion 204
  qualitative analysis 212
  quantitative analysis 212
Weibull equation 231
Weibull failure distribution 90, 97
Weibull failure rate function 206
Weibull graph 210
Weibull graph chart 101
Weibull hazard rate function 101, 227
Weibull life distribution 191

Weibull probability density 199, 227
Weibull probability distribution 219
Weibull reliability function 205
Weibull scale parameter 208
Weibull shape parameter 99
Weibull unreliability function 205
work breakdown structure (WBS) 63, 317

Y

Young's modulus 745, 746

Z

Zadeh's possibility measures 147