APPENDIX 1

MATLAB AND ANSYS PROGRAMS

This appendix lists all the MATLAB and ANSYS codes used in each chapter, along with a short description of the purpose of each.

MATLAB codes have the suffix ".m" and the ANSYS codes have the suffix ".inp." Additional output files from previous runs are stored as ".grp" or other suffixes and will be used from time to time.

Coding format: All the MATLAB code available from downloading and shown in the book starts over one tab, allowing comment lines to stand out. The code also includes a lot of blank lines for readability (my apologies to tight "c" code programmers).

In most MATLAB code, critical definitions and calculations are only a few lines of code, while plotting and annotating are the bulk of the space. For this reason, some code listings in the book do not show all the plotting commands.

ANSYS eigenvalue/eigenvector results are converted to MATLAB input form using the following MATLAB extraction codes:

ext56ux.m	extracts the ANSYS UX degree of freedom
ext56uy.m	extracts the ANSYS UY degree of freedom
ext56uz.m	extracts the ANSYS UZ degree of freedom
ext56uxuy.m	extracts the ANSYS UX and UY degrees of freedom
ext56uxuz.m	extracts the ANSYS UX and UZ degrees of freedom
ext56uyuz.m	extracts the ANSYS UY and UZ degrees of freedom
ext56uxuyuz.m	extracts the ANSYS UX, UY and UZ degrees of freedom

The codes above all call a supporting MATLAB code **ext56chk.m**. All the codes should be installed in the same directory as the ANSYS output code which is to be extracted or should be installed in a directory which is in the MATLAB path. To use the extraction code, just rename the ANSYS eigenvector output file to have a ".eig" extension and open MATLAB in the same directory. MATLAB will then open a window showing all the ".eig" files in the directory. Double-click on the file to extract and MATLAB will output a file with the "ext56xx.mat" name. If several files are to be extracted in the same directory, rename the "ext56xx.mat" name to a unique name with the ".mat" extension.

The ".mat" extracted MATLAB file contains the following information:

> **evr**, the modal matrix, with rows consisting of degrees of freedom and each column representing a mode. The numbering of degrees of freedom is the same as the ANSYS listing, which is in ascending order of the selected node numbers. Where multiple directions are extracted, for instance UX and UY degrees of freedom, the degrees of freedom are listed in that order, first the UX degrees of freedom and then the UY degrees of freedom. The extracted modal matrix is of size: (total dof) x (modes).
>
> **freqvec**, a vector listing the eigenvalues (resonant frequencies), in hz values. The size of the frequency vector is (modes) x (1).
>
> **node_numbers**, a vector listing the node numbers for the extracted data, of size (dof) x (1).
>
> The extracted data can then be loaded and used to develop state space models of the system.

Chapter 2: Transfer Function Analysis

sdofxfer.m: Calculates and plots magnitude and phase for a single degree of freedom system over a range of damping values.

tdofpz3x3.m: Uses the "num/den" form of the transfer function, calculates and plots all nine pole/zero combinations for the nine different transfer functions for tdof model. It prompts for values of the two dampers, c1 and c2, where the default (hitting the "enter" key) values are set to zero to match the hand calculated values in (2.82). The "transfer function" forms of the transfer functions are then converted to "zpk - zero/pole/gain" form to enable graphical construction of frequency response in the next chapter.

tdofpz3x3_rlocus.m: Plots pole and zero values for z11 transfer function for a range of damping values.

Chapter 3: Frequency Response Analysis

tdofxfer.m: Plots tdof model poles and zeros in complex plane, user choice of damping values. Uses several different model descriptions and frequency response calculating techniques. The model is described in polynomial, transfer function and zpk forms. Magnitude and phase versus frequency are calculated using a scalar frequency "for loop," vector frequency, automatic bode plotting and bode with magnitude and frequency outputs.

Chapter 4: Zeros in SISO Mechanical Systems

ndof_numzeros.m: Calculates and plots poles/zeros and transfer functions for user selected input/output locations on a "n" dof series spring/mass model. Shows that poles of "constrained" structures to left and right of input/output degrees of freedom are the zeros of the unconstrained structure.

cantfem.inp: ANSYS code for resonant frequencies of cantilever and tip driving point transfer function. Used to identify zero locations to compare with poles of "constrained" system in cantzero.inp.

cantzero.inp: ANSYS code for resonant frequencies of cantilever with simple support at tip. Used to identify poles of "constrained" structure.

cantzero.m: Uses eigenvalues and eigenvectors from cantfem.inp and cantzero.inp to plot overlay of zeros of cantilever with poles of tip supported cantilever, showing the correspondence. Calls **cantzero_freq.m**, **cantfem_magphs.m**.

Chapter 5: State Space Analysis

tdof_non_prop_damped.m: This code is used to develop an understanding of the results of MATLAB's eigenvalue analysis and complex modes.

Chapter 6: State Space: Frequency Response, Time Domain

tdofss.m: Calculates and plots the four distinct frequency responses for the tdof model.

tdof_ss_time_ode45_slnk.m: Solves for time domain response of tdof problem using MATLAB's ODE45 solver, a Runga-Kutta method of solving differential equations, as well as, MATLAB's Simulink block-diagram simulation tool.

tdof_ss_time_slnk_plot.m: Plots results from tdof_ss_time_ode45_slnk.m.

tdofssfun.m: Function code called by tdof_ss_time_ode45_slnk.m, contains state equations.

tdofss_simulink.mdl: Simulink model called by **tdof_ss_time_ode45_slnk.m**, defines state equations.

Chapter 8: Frequency Response: Modal Form

tdof_modal_xfer.m: Calculates and plots the four distinct frequency responses and the individual modal contributions.

threedof.inp: ANSYS code that builds the undamped tdof model, calculates eigenvalues and eigenvectors, outputs the frequency listing and eigenvectors, plots the mode shapes. Calculates and plots all three transfer functions for a force applied to mass 1.

Chapter 9: Transient Response: Modal Form

tdof_modal_time.m: Plots displacements versus time in principal and physical coordinates.

Chapter 10: Modal Analysis: State Space Form

tdofss_eig.m: Solves for the eigenvalues and eigenvectors in the state space form of the tdof system.

tdof_prop_damped.m: Calculates poles and zeros of proportionally damped tdof system. Plots initial condition responses for modes 2 and 3 in physical and principal coordinate systems.

Chapter 11: Frequency Response: Modal State Space Form

tdofss_modal_xfer_modes.m: Solves for and plots frequency responses for individual modal contributions and overall responses. Has code for plotting frequency responses in different forms.

Chapter 12: Time Domain: Modal State Space Form

tdofss_modal_time_ode45.m: Plots tdof transient responses for overall and individual modal contributions. Calls the function files below, which define the state space system and individual modes.

tdofssmodalfun.m, tdofssmodal1fun.m, tdofssmodal2fun.m, tdofssmodal3fun.m: Function files called by tdofss_modal_time_ode45.m.

Chapter 14: Finite Elements: Dynamics

cant_2el_guyan.m: Solves for the eigenvalues and eigenvectors of a two-element cantilever beam.

cantbeam_guyan.m: Solves for eigenvalues and eigenvectors of a cantilever with user-defined dimensions, material properties, number of elements and number of mode shapes to plot. Guyan Reduction is an option. A 10-element beam is used as an example.

cantbeam.inp: ANSYS code solves for the eigenvalues and eigenvectors of a 10 element cantilever, the same beam as the cantbeam_guyan.m example.

Chapter 15: SISO State Space MATLAB Model from ANSYS Model

cantbeam_ss.inp: ANSYS code for cantilever beam, allows the user to change the number of elements and the eigenvalue extraction technique. The two variables "num_elem" and "eigext" can be easily changed to see their effects.

cantbeam_ss_freq.m: Compares theoretical frequencies for the first 16 modes for a cantilever beam with MATLAB finite element and ANSYS finite element results.

cantbeam_ss_modred.m: Creates a MATLAB state space model using the eigenvalue and eigenvector results from previous ANSYS runs. Modes are ranked for importance and several reduction techniques are used.

Chapter 16: Ground Acceleration MATLAB Model from ANSYS Model

cantbeam_ss_spring_shkr.inp: ANSYS model of shaker mounted cantilever with tip mass and tip spring to shaker. Outputs mode shape plot file **cantbeam16red.grp**.

cantbeam_ss_tip_con.inp: ANSYS model of shaker mounted constrained tip cantilever. Outputs mode shape file **tipcon16red.grp**.

cantbeam_shkr_modeshape.m: Plots mode shapes from ANSYS modal analysis results for any of the tip spring models, with 2, 4, 8, 10, 12, 16, 32 and 64 beam elements.

cantbeam_ss_shkr_modred.m: Creates a MATLAB state space model using the results from ANSYS model **cantbeam_ss_spring_shkr.inp**. Ranks modes, then uses several reduction techniques to define smaller model.

Chapter 17: SISO Disk Drive Actuator Model

srun.inp: ANSYS model of suspension.

arun.inp: ANSYS model of actuator/suspension system.

act8.m: MATLAB code for dc and peak gain ranking and reduction of actuator/suspension model. Output from program is used for some input to **balred.m** in Chapter 18.

Chapter 18: Balanced Reduction

balred.m: MATLAB code for balanced reduction of actuator/suspension model from **act8.m**.

Chapter 19: MIMO Two-Stage Actuator Model

arunpz.inp: ANSYS model of two-stage actuator/suspension system.

act8pz.m: MATLAB model of two-stage actuator/suspension system, balanced reduction.

Downloading

All the programs listed can be downloaded from the MathWorks FTP site at www.mathworks.com or from the author's site at www.hatchcon.com.