CHAPTER 5

STATE SPACE ANALYSIS

5.1 Introduction

In Chapter 2 we derived the equations of motion for the tdof system shown in Figure 5.1, and showed how to solve the coupled differential equations for various transfer functions. In order to solve time domain problems using a computer, it is desirable to change the form of the equations for an n dof system with n second order differential equations to 2n first order differential equations. The first order form of equations of motion is known as **state space** form.

This chapter will develop the state space formulation for the tdof example. Once the state space formulation is completed, the subject of complex eigenvalues and eigenvectors, resulting in **complex modes** of vibration, will be covered in some detail. Once complex modes are understood, comprehending **real modes** which arise from the undamped case in the modal analysis section (Chapter 7) is simple.

Having an understanding of complex modes is especially helpful in working with experimental modal analysis. There are some very powerful experimental techniques available for testing and then visualizing the modes of vibration of structures. Frequency response data is taken at a number of selected positions on the structure and software is available to fit the data and define modes of vibration. The software identifies the resonant frequencies of the system and defines a damping value for each mode. It is then possible to create a model of the geometry of the test point locations and build a virtual model which can be animated to display the shape of motion of each mode.

The software has options which allow one to view the mode as either "real" or "complex." When the mode is viewed as "real," all the points on the structure move such that they all reach their maximum or minimum positions at the same point in time, which is consistent with our definition of "principal" or "real" modes defined in Chapter 7.

When the mode is viewed as "complex," the structure does not move such that all points reach either their minimum or maximum positions at the same point in time. Instead there appears to be a wave that moves along the structure as the different points reach their minimum or maximum positions at different times. For lightly damped mechanical structures, the assumption is often made that the modes are "real," allowing use of modal analysis methods and efficient finite element models. For structures that are not "lightly damped,"

the modal analysis method cannot be used and the state space formulation is the only practical method of solving the problem.

It is difficult to visualize complex modes without an animated structure model, but we will use a graphical method called an **Argand diagram** to explain how modes described by complex eigenvectors and complex eigenvalues combine to create physical motion of the system. We will find that if the unforced system is started from a set of initial conditions that match the complex eigenvector then only a single mode is excited. We will show how to calculate the transient response of the system for that specific initial condition case and illustrate how only a single mode is excited.

Chapter 6 will cover how to use the state space formulation to obtain both frequency and time domain results with MATLAB.

5.2 State Space Formulation

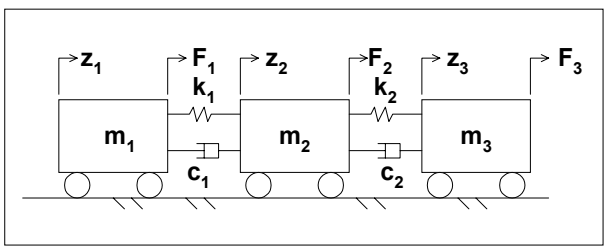

Figure 5.1: Original damped tdof system model.

Repeating the matrix equations of motion from (2.25):

$$\begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix}\begin{bmatrix} \ddot{z}_1 \\ \ddot{z}_2 \\ \ddot{z}_3 \end{bmatrix} + \begin{bmatrix} c_1 & -c_1 & 0 \\ -c_1 & (c_1+c_2) & -c_2 \\ 0 & -c_2 & c_2 \end{bmatrix}\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} + \begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & (k_1+k_2) & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix}\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} \tag{5.1}$$

Expanding the equations:

$$\begin{aligned} &m_1\ddot{z}_1 + c_1\dot{z}_1 - c_1\dot{z}_2 + k_1z_1 - k_1z_2 = F_1 \\ &m_2\ddot{z}_2 - c_1\dot{z}_1 + (c_1+c_2)\dot{z}_2 - c_2\dot{z}_3 - k_1z_1 + (k_1+k_2)z_2 - k_2z_3 = F_2 \\ &m_3\ddot{z}_3 - c_2\dot{z}_2 + c_2\dot{z}_3 - k_2z_2 + k_2z_3 = F_3 \end{aligned} \tag{5.2a,b,c}$$

The three equations above are second order differential equations which require knowledge of the initial states of position and velocity for all three degrees of freedom in order to solve for the transient response.

In the state space formulation, the three second order differential equations are converted to six first order differential equations. Following typical state space notation, we will refer to the states as "x" and the output as "y."

Start by solving (5.2) for the three equations for the highest derivatives, in this case the three second derivatives, $\ddot{z}_1,\ \ddot{z}_2,\ \ddot{z}_3$:

$$\ddot{z}_1 = (F_1 - c_1\dot{z}_1 + c_1\dot{z}_2 - k_1 z_1 + k_1 z_2)/m_1$$
$$\ddot{z}_2 = (F_2 + c_1\dot{z}_1 - (c_1 + c_2)\dot{z}_2 + c_2\dot{z}_3 + k_1 z_1 - (k_1 + k_2)z_2 + k\ z_3)/m_2$$
$$\ddot{z}_3 = (F_3 + c_2\dot{z}_2 - c_2\dot{z}_3 + k_2 z_2 - k_2 z_3)/m_3 \qquad (5.3a,b,c)$$

We now change notation, using "x" to define the six states; three positions and three velocities:

$$x_1 = z_1 \quad \text{Position of Mass 1} \qquad (5.4)$$
$$x_2 = \dot{z}_1 \quad \text{Velocity of Mass 1} \qquad (5.5)$$
$$x_3 = z_2 \quad \text{Position of Mass 2} \qquad (5.6)$$
$$x_4 = \dot{z}_2 \quad \text{Velocity of Mass 2} \qquad (5.7)$$
$$x_5 = z_3 \quad \text{Position of Mass 3} \qquad (5.8)$$
$$x_6 = \dot{z}_3 \quad \text{Velocity of Mass 3} \qquad (5.9)$$

By using this notation, we observe the relationship between the state and its first derivatives:

$$\dot{z}_1 = x_2 = \dot{x}_1 \qquad (5.10)$$
$$\dot{z}_2 = x_4 = \dot{x}_3 \qquad (5.11)$$
$$\dot{z}_3 = x_6 = \dot{x}_5 \qquad (5.12)$$

Also between the first and second derivatives:

$$\ddot{z}_1 = \dot{x}_2 \qquad (5.13)$$
$$\ddot{z}_2 = \dot{x}_4 \qquad (5.14)$$
$$\ddot{z}_3 = \dot{x}_6 \qquad (5.15)$$

Rewriting the three equations for $\ddot{z}_1$, $\ddot{z}_2$, $\ddot{z}_3$ in terms of the six states x_1 through x_6 and adding the three equations defining the position and velocity relationships:

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= (F_1 - c_1 x_2 + c_1 x_4 - k_1 x_1 + k_1 x_3)/m_1 \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= (F_2 + c_1 x_2 - (c_1 + c_2) x_4 + c_2 x_6 + k_1 x_1 - (k_1 + k_2) x_3 + k_2 x_5)/m_2 \\
\dot{x}_5 &= x_6 \\
\dot{x}_6 &= (F_3 + c_2 x_4 - c_2 x_6 + k_2 x_3 - k_2 x_5)/m_3
\end{aligned} \qquad \text{(5.16a-f)}$$

Rewriting the equations above in matrix form as:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{-k_1}{m_1} & \frac{-c_1}{m_1} & \frac{k_1}{m_1} & \frac{c_1}{m_1} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{k_1}{m_2} & \frac{c_1}{m_2} & \frac{-(k_1+k_2)}{m_2} & \frac{-(c_1+c_2)}{m_2} & \frac{k_2}{m_2} & \frac{c_2}{m_2} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{k_2}{m_3} & \frac{c_2}{m_3} & \frac{-k_2}{m_3} & \frac{-c_2}{m_3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{F_1}{m_1} \\ 0 \\ \frac{F_2}{m_2} \\ 0 \\ \frac{F_3}{m_3} \end{bmatrix} \quad (1)$$

$$\dot{\mathbf{x}} = \mathbf{A}\,\mathbf{x} + \mathbf{B}\,u \qquad \text{(5.17a,b)}$$

5.3 Definition of State Space Equations of Motion

Schematically, a SISO state space system is represented as shown in Figure 5.2. We will define the blocks in the following sections. The scalar input u(t) is fed into both the input matrix **B** and the direct transmission matrix **D**. The output of the input matrix is an nx1 vector, where "n" is the number of states. For a SISO system, the direct transmission matrix is a scalar, and its output is fed into a summing junction to be added to the output of the **C** matrix.

The output of the **B** matrix is added to the feedback term coming from the system matrix and is fed into an integrator block, where "**I**" is an nxn identify matrix. The output matrix has as many rows as outputs, a single row for a

SISO system, and has as many columns as states, n. The output y(t) is the sum of the output of the **C** and **D** matrices.

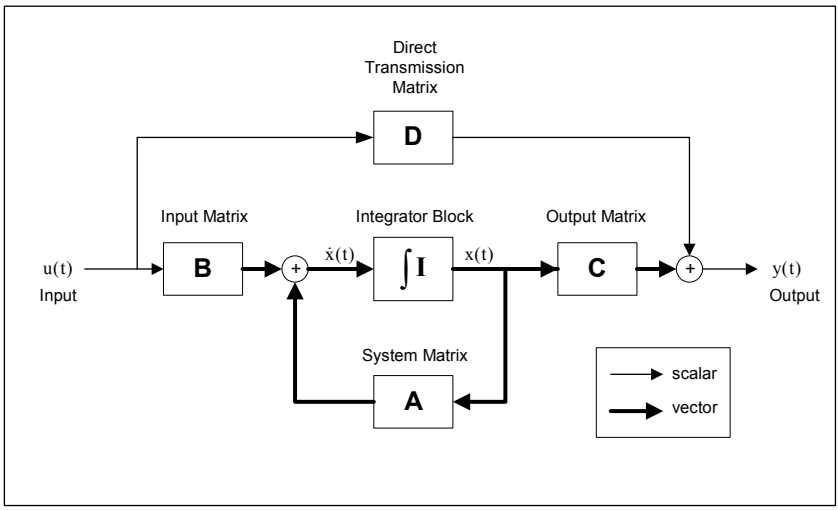

Figure 5.2: State space system block diagram.

Notation for equations of motion in state space form is:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \tag{5.18}$$

where the **A** and **B** matrices are shown in (5.17a). Matrix **A** is known as the system matrix, matrix **B** is the input matrix, and scalar u is the input. The column vector **x** is the state of the system.

5.4 Input Matrix Forms

Because "u" is a scalar, the nature of the input matrix **B** changes depending on what input is used. If the system is a Single Input (SI) system with a force either at mass 1, 2 or 3, the **B** matrix changes as follows:

$$F_1: \quad \mathbf{B} = \begin{bmatrix} 0 \\ \frac{F_1}{m_1} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad F_2: \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{F_2}{m_2} \\ 0 \\ 0 \end{bmatrix}, \quad F_3: \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{F_3}{m_3} \end{bmatrix} \qquad (5.19a,b,c)$$

If the same forcing function u (for example, a step function or sine function) is applied to several degrees of freedom simultaneously (for example, a force of magnitude F_1 to mass 1 and a force of magnitude F_3 to mass 3) the input matrix would become:

$$\mathbf{B} = \begin{bmatrix} 0 \\ \frac{F_1}{m_1} \\ 0 \\ 0 \\ 0 \\ \frac{F_3}{m_3} \end{bmatrix} \qquad (5.20)$$

For a Multi Input (MI) system, where forces are applied independent of one another to the separate masses, a multiple column input matrix is appropriate. For example, for different inputs at mass 1 and mass 2, none at mass 3, the input matrix would become:

$$\mathbf{B} = \begin{bmatrix} 0 & 0 \\ \frac{F_1}{m_1} & 0 \\ 0 & \frac{F_2}{m_2} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad (5.21)$$

5.5 Output Matrix Forms

To account for the case where the desired output is not just the states but is some linear combination of the states, an output matrix **C** is defined to relate the outputs to the states. Also, a matrix **D**, known as the direct transmission matrix, is multiplied by the input "u" to account for outputs that are related to the inputs but that bypass the states.

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathrm{u} \tag{5.22}$$

The output matrix **C** has as many rows as outputs required and as many columns as states. The direct transmission matrix **D** has the same number of columns as the input matrix **B** and as many rows as the output matrix **C**.

In our example, we are interested in all six of the states, displacements and velocities, so the matrix output equation becomes, where **C** is the identity matrix and **D** is assumed to be zero:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} (1) \tag{5.23}$$

Expanding, the matrix equations become:

$$y_1 = x_1 \qquad (= z_1) \tag{5.24}$$

$$y_2 = x_2 \qquad (= \dot{z}_1) \tag{5.25}$$

$$y_3 = x_3 \qquad (= z_2) \tag{5.26}$$

$$y_4 = x_4 \qquad (= \dot{z}_2) \tag{5.27}$$

$$y_5 = x_5 \qquad (= z_3) \tag{5.28}$$

$$y_6 = x_6 \qquad (= \dot{z}_3) \tag{5.29}$$

If we were only interested in the three displacements and not the three velocities, the output equation would be, assuming **D** is zero:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + (0)(1) \tag{5.30}$$

Expanding:

$$y_1 = x_1 \qquad (= z_1) \tag{5.31}$$

$$y_2 = x_3 \qquad (= z_2) \tag{5.32}$$

$$y_3 = x_5 \qquad (= z_3) \tag{5.33}$$

On the other hand, if the outputs are linear combinations of the states, as in a control system problem, the output equation could look like (where a, b and c are scalars), assuming **D** is zero:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & a & 0 & b & 0 \\ c & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + (0)(1) \tag{5.34}$$

Expanding:

$$y_1 = ax_3 + bx_5 \qquad (= az_2 + bz_3) \tag{5.35}$$

$$y_2 = cx_1 + x_3 \qquad (= cz_1 + z_2) \tag{5.36}$$

$$y_3 = x_1 \qquad (= z_1) \tag{5.37}$$

$$y_4 = x_4 \qquad (= \dot{z}_2) \tag{5.38}$$

If a single force is applied and a single output is desired (SISO), for example, a force applied at mass 1 and the output displacement at mass 3, assuming **D** is zero:

$$y = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + (0)(1) \tag{5.39}$$

With all the possible variations of the output equation, the state equation never changes; it is always:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \tag{5.40}$$

5.6 Complex Eigenvalues and Eigenvectors – State Space Form

The most basic analysis one can perform on a dynamic system is to solve for its eigenvalues (natural frequencies) and eigenvectors (mode shapes). In this section we will develop the most general case where there are no limitations on the presence or magnitude of the two damping terms, which could result in complex eigenvalues and eigenvectors.

Start by postulating that there is a set of initial conditions such that if the system is released with that set, the system will respond in one of its natural modes of vibration. To that end, we set the forcing function to zero and write the homogeneous state space equations of motion:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} \tag{5.41}$$

We define motion in a principal mode as:

$$\mathbf{x}_i = \mathbf{x}_{mi}\, e^{\lambda_i t} \tag{5.42}$$

Where:

λ_i is the i^{th} eigenvalue, the natural frequency of the i^{th} mode of vibration

$\mathbf{x}_i$ is the vector of states at the i^{th} frequency

$\mathbf{x}_{mi}$ is the i^{th} eigenvector, the mode shape for the i^{th} mode

For our tdof (z_1 to z_3), six state (x_1 to x_6) system, for the i^{th} eigenvalue and eigenvector, the equation would appear as:

$$\begin{bmatrix} z_{1i} \\ \dot{z}_{1i} \\ z_{2i} \\ \dot{z}_{2i} \\ z_{3i} \\ \dot{z}_{3i} \end{bmatrix} = \begin{bmatrix} x_{1i} \\ x_{2i} \\ x_{3i} \\ x_{4i} \\ x_{5i} \\ x_{6i} \end{bmatrix} = \mathbf{x}_{mi} e^{\lambda_i t} = \begin{bmatrix} x_{m1i} \\ x_{m2i} \\ x_{m3i} \\ x_{m4i} \\ x_{m5i} \\ x_{m6i} \end{bmatrix} e^{\lambda_i t} \tag{5.43}$$

Differentiating the modal displacement equation above to get the modal velocity equation:

$$\dot{\mathbf{x}}_{mi} = \frac{d}{dt}\left[\mathbf{x}_{mi} e^{\lambda t}\right] = \lambda \mathbf{x}_{mi} e^{\lambda t} \tag{5.44}$$

Substituting into the state equation and canceling the exponential terms leads to:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} \\ \lambda \mathbf{x}_{mi} e^{\lambda t} &= \mathbf{A}\mathbf{x}_{mi} e^{\lambda t} \\ \lambda \mathbf{x}_{mi} &= \mathbf{A}\mathbf{x}_{mi} \\ (\lambda \mathbf{I} - \mathbf{A})\mathbf{x}_{mi} &= 0 \end{aligned} \tag{5.45a-d}$$

Equation (5.45c) is the classic "eigenvalue problem." If $\mathbf{x}_{mi}$ is not equal to zero in (5.45d), a solution exists only if the determinant below is zero (Strang 1998):

$$\left|(\lambda \mathbf{I} - \mathbf{A})\right| = 0 \tag{5.46}$$

Taking the system matrix **A** from (5.17a) and inserting in (5.45):

$$(\lambda \mathbf{I} - \mathbf{A}) = \lambda\, \mathbf{I} - \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{-k_1}{m_1} & \frac{-c_1}{m_1} & \frac{k_1}{m_1} & \frac{c_1}{m_1} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{k_1}{m_2} & \frac{c_1}{m_2} & \frac{-(k_1+k_2)}{m_2} & \frac{-(c_1+c_2)}{m_2} & \frac{k_2}{m_2} & \frac{c_2}{m_2} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{k_2}{m_3} & \frac{c_2}{m_3} & \frac{-k_2}{m_3} & \frac{-c_2}{m_3} \end{bmatrix} \tag{5.47}$$

In Chapter 10 we will use the undamped version of (5.46) with c1 = c2 = 0 to discuss "normal" modes, where we will find that taking the determinant in closed form is practical. For the tdof damped system matrix, taking the closed form determinant is far too complicated so we will use MATLAB's "eig" function to solve the eigenvalue problem numerically, using specific values of m, c and k. We will use the MATLAB code **tdof_non_prop_damped.m** as we continue our exploration of complex modes.

5.7 MATLAB Code tdof_non_prop_damped.m: Methodology, Model Setup, Eigenvalue Calculation Listing

The sequence of development of complex modes is as follows:

1) solve original damped system equation for complex eigenvalues and eigenvectors

2) normalize the eigenvector entries to unity

3) calculate magnitude and phase angle of each of the eigenvector entries

4) use the Argand diagram to visualize the motion of a complex mode

5) calculate the percentage of critical damping (damping ratio) for each mode

6) calculate the motions of the three masses for all three modes

7) plot the real and imaginary displacements of each of the degrees of freedom separately

We have explored how to calculate the eigenvectors or mode shapes for an undamped problem using the transfer function matrix (Chapter 3). The modes for the undamped problem were real modes, meaning that the position elements of the eigenvectors were real, not complex, and we were able to plot diagrams showing the shape of the modes. For complex modes, it is not possible to draw a picture of the deformed mode shape because there are phase differences between the various degrees of freedom which prevent them from reaching their maximum/minimum points at the same point in time. This leads to the apparent "traveling wave" in an animated mode.

The first section of **tdof_non_prop_damped.m** sets up the state space equations of motion and solves the eigenvalue problem for damping values of $c_1 = 0.1, c_2 = 0.2$:

```
%       tdof_non_prop_damped.m          non-proportionally damped tdof model

        clf;

        legend off;

        subplot(1,1,1);

        clear all;

%       define the values of masses, springs, dampers

        m1 = 1;
        m2 = 1;
        m3 = 1;

        k1 = 1;
        k2 = 1;

%       define arbitrary damping values

        c1 = input('input value for c1, default 0.1, ... ');

        if  (isempty(c1))
                c1 = 0.1;
        else
        end

        c2 = input('input value for c1, default 0.2, ... ');

        if  (isempty(c2))
                c2 = 0.2;
        else
        end
```

```
%       define the system matrix, aphys, in physical coordinates

        aphys = [      0        1          0          0          0        0
                    -k1/m1   -c1/m1     k1/m1      c1/m1        0        0
                       0        0          0          1          0        0
                     k1/m2    c1/m2   -(k1+k2)/m2  -(c1+c2)/m2  k2/m2    c2/m2
                       0        0          0          0          0        1
                       0        0         k2/m3      c2/m3     -k2/m3   -c2/m3];

%       solve for the eigenvalues of the system matrix


        [xm,lambda] = eig(aphys);

%       take the diagonal elements of the generalized eigenvalue matrix lambda

        lambdad = diag(lambda);
```

The six eigenvalues, lambda values, are listed below. Since we have three degrees of freedom, there should be three sets of complex conjugate eigenvalues.

```
xm =

  Columns 1 through 4

  -0.0567 - 0.1940i  -0.0567 + 0.1940i   0.2886 - 0.4085i   0.2886 + 0.4085i
   0.3452 - 0.0535i   0.3452 + 0.0535i   0.3865 + 0.3190i   0.3865 - 0.3190i
   0.0624 + 0.4029i   0.0624 - 0.4029i  -0.0218 - 0.0123i  -0.0218 + 0.0123i
  -0.7046 + 0.0162i  -0.7046 - 0.0162i   0.0139 - 0.0209i   0.0139 + 0.0209i
  -0.0057 - 0.2089i  -0.0057 + 0.2089i  -0.2668 + 0.4208i  -0.2668 - 0.4208i
   0.3593 + 0.0373i   0.3593 - 0.0373i  -0.4004 - 0.2981i  -0.4004 + 0.2981i

  Columns 5 through 6

   0.0000 - 0.5774i   0.0000 + 0.5774i
   0.0000 + 0.0000i   0.0000 - 0.0000i
   0.0000 - 0.5774i   0.0000 + 0.5774i
   0.0000 + 0.0000i   0.0000 - 0.0000i
   0.0000 - 0.5774i   0.0000 + 0.5774i
   0.0000 + 0.0000i   0.0000 - 0.0000i


lambda =

  Columns 1 through 4

  -0.2250 + 1.7141i        0                0                0
        0          -0.2250 - 1.7141i        0                0
        0                  0          -0.0750 + 0.9991i        0
        0                  0                0          -0.0750 - 0.9991i
        0                  0                0                0
```

```
     0          0          0          0

 Columns 5 through 6

     0          0
     0          0
     0          0
     0          0
-0.0000 + 0.0000i       0
     0          -0.0000 - 0.0000i


lambdad =

 -0.2250 + 1.7141i
 -0.2250 - 1.7141i
 -0.0750 + 0.9991i
 -0.0750 - 0.9991i
 -0.0000 + 0.0000i
 -0.0000 - 0.0000i
```

Note that the two eigenvalues which correspond to each of the three modes are complex conjugates of each other, and that the real parts of the second and third mode eigenvalues are all negative.

We did not specify the form of the eigenvalues, which in the most general case can be complex, as in the second and third modes above. We will now discuss the components of complex eigenvalues. We use the term λ_{n1} to describe the first complex eigenvalue of any of the three sets of eigenvalues above. The term λ_{n2} is used to describe the second complex eigenvalue of the set, and the complex conjugacy of the two is stated as: $\lambda_{n2} = \lambda_{n1}^{*}$, where the "*" indicates a complex conjugate. The real and imaginary parts will be defined using σ_{nx} and ω_{nx}, respectively:

$$\begin{aligned} \lambda_{n1} &= \sigma_{n1} + j\omega_{n1} \\ \lambda_{n2} &= \lambda_{n1}^{*} = \sigma_{n1} - j\omega_{n1} \end{aligned} \tag{5.48}$$

See Figure 5.3 for graphical descriptions of the components of a complex eigenvalue. The figure shows two complex conjugate eigenvalues (poles) in the left half plane as "x" symbols. The real parts of the two eigenvalues are the same and are given the symbol σ, with the imaginary parts both having a distance from the origin of ω, referred to as the damped natural frequency. The radial distance from the origin to the poles is given by ω_n and is referred to as the undamped natural frequency. The angle between the imaginary axis and the line from the origin to the pole is used to define the amount of

damping of the mode, referred to as ζ, the damping ratio or percentage of critical damping. If $\sigma = 0$, $\theta = 0$ and there is no damping, therefore $\omega = \omega_n$.

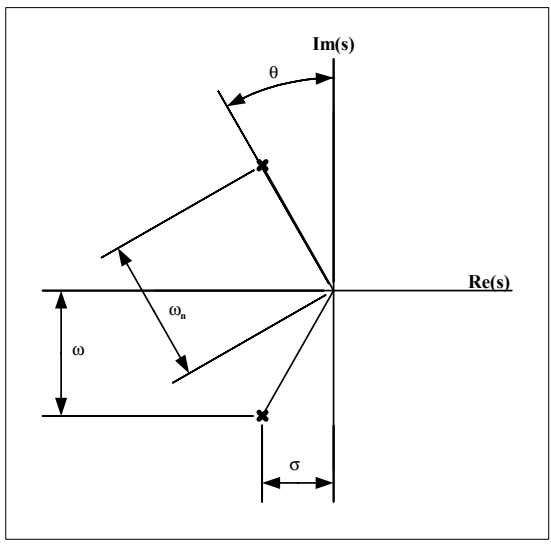

Figure 5.3: Complex eigenvalue (pole) nomenclature in complex plane.

Referring to Figure 5.3 for the definition of θ, the equation for calculating ζ for a mode from the real and imaginary components of the eigenvalue is:

$$\begin{aligned} \zeta &= \sin\theta \\ &= \sin\left(\tan^{-1}\left(\frac{\mathrm{Re}(\lambda)}{\mathrm{Im}(\lambda)}\right)\right) \\ &= \sin\left(\tan^{-1}\left(\frac{\sigma}{\omega}\right)\right) \end{aligned} \tag{5.49}$$

5.8 Eigenvectors – Normalized to Unity

The section of code below reorders the eigenvectors from low to high frequency and normalizes them. The normalization procedure is to divide each eigenvector by its position state for mass 1, the first term in each eigenvector.

```
%        now reorder the eigenvalues and eigenvectors from low to high frequency,
```

```
%       keeping track of how the eigenvalues are ordered in reorder the
%       eigenvectors to match, using indexhz

        [lambdaorder,indexhz] = sort(abs(imag(lambdad)));

        for  cnt = 1:length(lambdad)

            lambdao(cnt,1) = lambdad(indexhz(cnt));          %  reorder eigenvalues

            xmo(:,cnt) = xm(:,indexhz(cnt));          %  reorder eigenvector columns

        end

%       now normalize the eigenvectors with respect to the position of mass 1, which
%       will be set to 1.0

        for  cnt = 1:length(lambdad)

                xmon1(:,cnt) = xmo(:,cnt)/xmo(1,cnt);

        end
```

The eigenvectors, normalized such that the displacements of mass 1 are set to 1.0 are shown below as xmon1.

```
lambdao =

  -0.0000 + 0.0000i
  -0.0000 - 0.0000i
  -0.0750 + 0.9991i
  -0.0750 - 0.9991i
  -0.2250 + 1.7141i
  -0.2250 - 1.7141i


xmo =

  Columns 1 through 4

   0.0000 - 0.5774i   0.0000 + 0.5774i   0.2886 - 0.4085i   0.2886 + 0.4085i
   0.0000 + 0.0000i   0.0000 - 0.0000i   0.3865 + 0.3190i   0.3865 - 0.3190i
   0.0000 - 0.5774i   0.0000 + 0.5774i  -0.0218 - 0.0123i  -0.0218 + 0.0123i
   0.0000 + 0.0000i   0.0000 - 0.0000i   0.0139 - 0.0209i   0.0139 + 0.0209i
   0.0000 - 0.5774i   0.0000 + 0.5774i  -0.2668 + 0.4208i  -0.2668 - 0.4208i
   0.0000 + 0.0000i   0.0000 - 0.0000i  -0.4004 - 0.2981i  -0.4004 + 0.2981i

  Columns 5 through 6

  -0.0567 - 0.1940i  -0.0567 + 0.1940i
   0.3452 - 0.0535i   0.3452 + 0.0535i
   0.0624 + 0.4029i   0.0624 - 0.4029i
  -0.7046 + 0.0162i  -0.7046 - 0.0162i
  -0.0057 - 0.2089i  -0.0057 + 0.2089i
```

```
  0.3593 + 0.0373i   0.3593 - 0.0373i


xmon1 =

 Columns 1 through 4

  1.0000 - 0.0000i   1.0000 + 0.0000i   1.0000            1.0000
  0.0000 + 0.0000i   0.0000 - 0.0000i  -0.0750 + 0.9991i  -0.0750 - 0.9991i
  1.0000 - 0.0000i   1.0000 + 0.0000i  -0.0050 - 0.0498i  -0.0050 + 0.0498i
 -0.0000 + 0.0000i  -0.0000 - 0.0000i   0.0502 - 0.0013i   0.0502 + 0.0013i
  1.0000 - 0.0000i   1.0000 + 0.0000i  -0.9950 + 0.0498i  -0.9950 - 0.0498i
  0.0000 + 0.0000i   0.0000 - 0.0000i   0.0248 - 0.9978i   0.0248 + 0.9978i

 Columns 5 through 6

  1.0000 - 0.0000i   1.0000 + 0.0000i
 -0.2250 + 1.7141i  -0.2250 - 1.7141i
 -2.0001 - 0.2630i  -2.0001 + 0.2630i
  0.9009 - 3.3691i   0.9009 + 3.3691i
  1.0001 + 0.2630i   1.0001 - 0.2630i
 -0.6759 + 1.6550i  -0.6759 - 1.6550i
```

The six rows of each eigenvector are related to the six states, x_1 to x_6, where x_1, x_3, x_5 are the displacement states and x_2, x_4, x_6 are the velocity states. Each velocity row is equal to the displacement row associated with it times its eigenvector, as can be seen by repeating (5.41) and differentiating it.

$$\begin{aligned} \mathbf{x}_i &= \mathbf{x}_{mi} e^{\lambda_i t} \\ \dot{\mathbf{x}}_i &= \lambda_i (\mathbf{x}_{mi} e^{\lambda_i t}) \end{aligned} \tag{5.50}$$

The tdof model has three degrees of freedom, so we should have three modes of vibration. The first two columns of the eigenvector matrix define mode 1, the third and fourth define mode 2 and the fifth and sixth columns define mode 3. Like the two complex conjugate eigenvalues for each mode, the two eigenvector columns for each of the modes are complex conjugates of each other.

5.9 Eigenvectors – Magnitude and Phase Angle Representation

Another way of looking at the eigenvectors is to calculate the magnitude and phase angle for each entry. The code for doing this follows.

```
%       now calculate the magnitude and phase angle of each of the eigenvector
%       entries

        for  row = 1:length(lambdad)
```

```
                for  col = 1:length(lambdad)

                        xmon1mag(row,col) = abs(xmon1(row,col));

                        xmon1ang(row,col) = (180/pi)*angle(xmon1(row,col));

                end

        end

        lambdao

        xmo

        xmon1

        xmon1mag

        xmon1ang
```

The magnitude and phase angles are:

```
xmon1mag =
   1.0000   1.0000   1.0000   1.0000   1.0000   1.0000
   0.0000   0.0000   1.0019   1.0019   1.7288   1.7288
   1.0000   1.0000   0.0501   0.0501   2.0173   2.0173
   0.0000   0.0000   0.0502   0.0502   3.4875   3.4875
   1.0000   1.0000   0.9962   0.9962   1.0341   1.0341
   0.0000   0.0000   0.9981   0.9981   1.7877   1.7877


xmon1ang =
        0         0         0         0         0         0
  90.0000  -90.0000   94.2930  -94.2930   97.4782  -97.4782
   0.0000    0.0000  -95.7723   95.7723 -172.5081  172.5081
  90.0000  -90.0000   -1.4793    1.4793  -75.0299   75.0299
   0.0000    0.0000  177.1334 -177.1334   14.7356  -14.7356
  90.0000  -90.0000  -88.5736   88.5736  112.2138 -112.2138
```

We will see in Chapter 7 that undamped eigenvector oscillatory modes have phases that are multiples of 90°. For the damped complex eigenvectors the phases are slightly offset from being 90° multiples of each other.

5.10 Complex Eigenvectors Combining to Give Real Motions

Now that we have solved for the complex eigenvalues and eigenvectors, we will discuss how we can have the system respond in only a single mode of vibration by releasing the system with a particular set of initial conditions. We will answer the following question:

How does a mode that is described by complex eigenvalues and eigenvectors give "real," physically observable motions (Newland 1989)?

For the n^{th} mode, the motion in that mode is defined as the sum of the motions due to the two conjugate eigenvalues/eigenvectors for that mode, as shown in (5.51). Substituting the complex conjugate value and collecting exponential terms:

$$\begin{aligned}\mathbf{x}(t) &= e^{\lambda_{n1}t}\mathbf{x}_{n1} + e^{\lambda_{n2}t}\mathbf{x}_{n2} \\ &= e^{\lambda_{n1}t}\mathbf{x}_{n1} + e^{\lambda_{n1}^{*}t}\mathbf{x}_{n1}^{*} \\ &= e^{(\sigma_{n1}+j\omega_{n1})t}\mathbf{x}_{n1} + e^{(\sigma_{n1}-j\omega_{n1})t}\mathbf{x}_{n1}^{*} \\ &= e^{\sigma_{n1}t}\left(e^{j\omega_{n1}t}\mathbf{x}_{n1} + e^{-j\omega_{n1}t}\mathbf{x}_{n1}^{*}\right) \\ &= 2e^{\sigma_{n1}t}\,\mathrm{Re}(\mathbf{x}_{n1})\end{aligned} \tag{5.51}$$

The $e^{j\omega_{n1}t}\mathbf{x}_{n1}$ term represents a vector of magnitude $|\mathbf{x}_{n1}|$ which is rotating counter-clockwise at the rate of ω_{n1} radians/sec. The $e^{-j\omega_{n1}t}\mathbf{x}_{n1}^{*}$ term represents a vector of magnitude $|\mathbf{x}_{n1}^{*}|$ which is rotating clockwise at the rate of ω_{n1} radians/sec. This counter-rotation is the key to understanding how the sum of two complex numbers becomes real. Since the two counter-rotating eigenvector terms are complex conjugates, their imaginary portions are of opposite sign and as they rotate, the sum of the two results in only a real component as the two imaginary portions cancel each other. See the Argand diagram in the next section for a graphical representation.

The $e^{\sigma_{n1}t}$ term is an exponentially decreasing scalar which multiplies the sum of the two counter-rotating vectors. The σ_{n1} term is the real value of the eigenvalue, and for a stable mode, with the poles in the left half of the s-plane, the value is always negative. Thus, $e^{\sigma_{n1}t}$ is exponentially decreasing with a time constant of $1/\sigma_{n1}$.

For real modes, the poles are on the imaginary axis, so $\sigma_{n1} = 0$ and $e^{(0)t} = 1$. The two counter-rotating vectors are not attenuated in amplitude with time, so the motion is undamped.

If the initial conditions for the system are set at one of the eigenvectors, the system will respond in only that mode**. For systems with complex modes, initial conditions of both displacements and velocities of all the masses must be set simultaneously in order for the system to respond only in that mode.** If the initial conditions for the system are set at any other value, the

resulting motion will be composed of a superposition of the motions of several modes.

For undamped systems with normal modes, either the displacement or velocity initial conditions can be set and the system will respond only in that mode (see Chapter 7 for more details).

Equation (5.51) will be used in the MATLAB code for plotting the motion of the system for the two oscillatory modes.

5.11 Argand Diagram Introduction

Since we are dealing with complex modes where different parts of the structure reach their maximum and minimum positions at different times, we cannot plot deformed mode shape plots as we did for the undamped model in Chapter 3. The best way to visualize complex modes is by animating the mode shape, allowing one to see the different parts of the structure moving in time.

The use of an Argand or Phasor diagram is another way to visualize the motion. It plots rotating eigenvectors of position and velocity in the complex plane for each degree of freedom in the eigenvector and shows how the complex conjugate eigenvector components add to create the "real" motion.

The normalized eigenvector matrix, xmon1, is repeated below. The first two states, position and velocity of mass 1, dof z1, are highlighted in bold type for the second mode of vibration.

Figure 5.4 shows Argand diagrams for the highlighted mode and states in the eigenvector matrix below. All three plots are in the complex plane. The upper left-hand plot shows the position and velocity eigenvector components for the third column of the eigenvector matrix, where the position component is 1+0j and the velocity component is –0.075+0.999j. The position component plots from 0 to 1 on the real axis. Notice that the tip of the velocity vector is slightly to the left of the imaginary axis. The $e^{j\omega_2 t}$ term indicates that the position and velocity vectors are both rotating in the counter-clockwise direction at a speed of ω radians/sec, starting from the initial locations defined by the eigenvector components.

```
xmon1 =
  1.0000                1.0000                1.0000                1.0000
  0.0000 + 0.0000i      0.0000 - 0.0000i     -0.0750 + 0.9991i     -0.0750 - 0.9991i
  1.0000 + 0.0000i      1.0000 - 0.0000i     -0.0050 - 0.0498i     -0.0050 + 0.0498i
  0.0000 + 0.0000i      0.0000 - 0.0000i      0.0502 - 0.0013i      0.0502 + 0.0013i
  1.0000 + 0.0000i      1.0000 - 0.0000i     -0.9950 + 0.0498i     -0.9950 - 0.0498i
  0.0000 + 0.0000i      0.0000 - 0.0000i      0.0248 - 0.9978i      0.0248 + 0.9978i

  1.0000                1.0000
 -0.2250 + 1.7141i     -0.2250 - 1.7141i
 -2.0001 - 0.2630i     -2.0001 + 0.2630i
  0.9009 - 3.3691i      0.9009 + 3.3691i
  1.0001 + 0.2630i      1.0001 - 0.2630i
 -0.6759 + 1.6550i     -0.6759 - 1.6550i
```

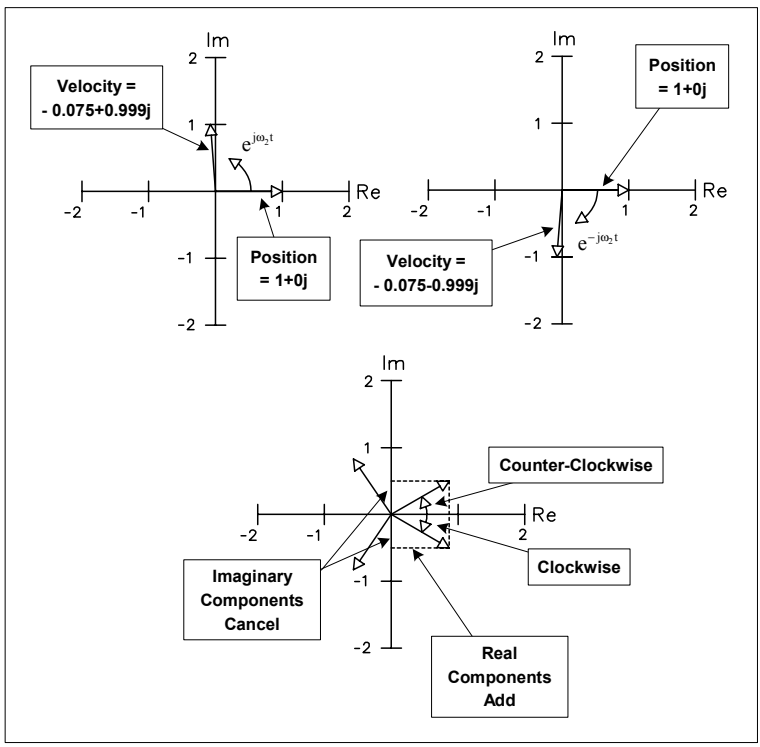

Figure 5.4: Argand diagram explanation.

The upper right-hand plot is similar to the left-hand plot except that the fourth column entries of the eigenvector matrix for the first two states are plotted and the two vectors are rotating in the clockwise direction. Note that the real components of the position and velocity components are the same as the third column, but that the imaginary components are complex conjugates of each other.

The lower plot illustrates the complex plane with both third and fourth eigenvectors shown on the same plot after rotating through the angle $\omega_2 t$. At any time "t," the two counter-rotating position vectors can be added to give the current position. At any time, the two imaginary components cancel out, leaving only the sum of the two real axis components as the "real" position. The same vector addition of the two counter-rotating velocity vectors will give the "real" velocity.

For an undamped model, the lengths of the two original eigenvector components stay the same. For the damped model, the lengths of all the vectors decrease continuously with a time constant of $1/\sigma_2$.

Looking at the Argand diagram above, which shows the "real" motion as twice the real axis component of the vector, it is clear that the motion as a function of time can also be written as:

$$\begin{aligned} \mathbf{x}(t) &= 2\, e^{\sigma_{n1} t} \left|\mathbf{x}_{n1}\right| \cos(\omega t + \phi_{ni}) \\ &= 2\, e^{\sigma_{n1} t}\ \mathrm{Re}(\mathbf{x}_{n1}) \end{aligned} \tag{5.52}$$

where the phase angle ϕ_{ni} is given by:

$$\tan(\phi_{ni}) = \mathrm{Im}(z_{ni}) / \mathrm{Re}(z_{ni}) \tag{5.53}$$

5.12 Calculating ζ, Plotting Eigenvalues in Complex Plane, Frequency Response

This section of code calculates the percentage of critical damping for each of the three modes, ζ_i using (5.49).

```
%       calculate the percentage of critical damping for each mode

        zeta1 = 0

        theta2 = atan(real(lambdao(3))/imag(lambdao(3)));
        zeta2 = abs(sin(theta2))

        theta3 = atan(real(lambdao(5))/imag(lambdao(5)));
        zeta3 = abs(sin(theta3))

        plot(lambda,'k*')
        grid on
        axis([-3 1 -2 2])
        axis('square')
        title('Damped Eigenvalues')
        xlabel('real')
```

```
ylabel('imaginary')
text(real(lambdao(3))-1,imag(lambdao(3))+0.1,['zeta = ',num2str(zeta2)])
text(real(lambdao(5))-1,imag(lambdao(5))+0.1,['zeta = ',num2str(zeta3)])
disp('execution paused to display figure, "enter" to continue'); pause
```

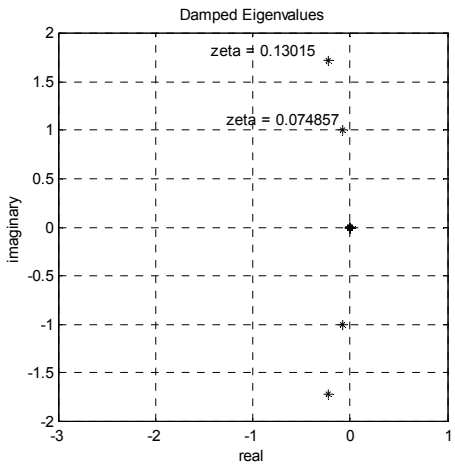

Figure 5.5: Plot of eigenvalues in complex plane for tdof model with c1 = 0.1, c2 = 0.2.

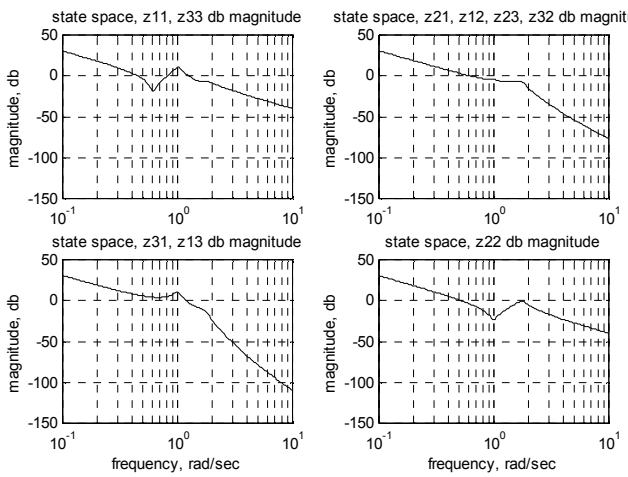

Figure 5.6: Frequency response magnitude plots.

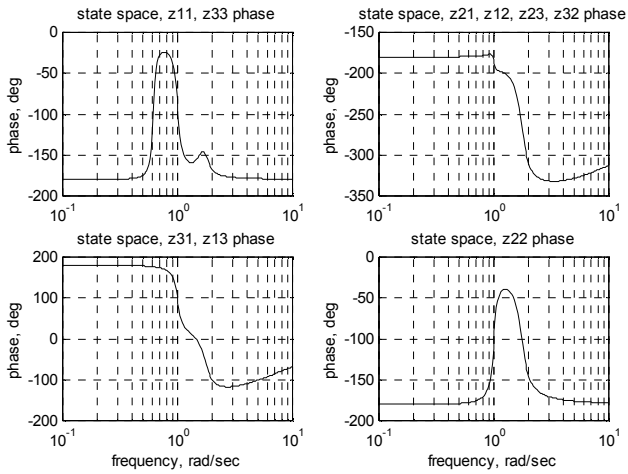

Figure 5.7: Frequency response phase plots.

The magnitude and phase frequency response plots for the system with c1 = 0.1 and c2 = 0.2 are shown above, using **tdofss.m** to plot. Note the significant attenuation of the resonances with zetas of 7.5% and 13% for modes 1 and 2, respectively. (Note: This amount of damping is very difficult to obtain in most practical structures without the use of additive damping.)

5.13 Initial Condition Responses of Individual Modes

The code below calculates the initial condition response for the oscillatory (not rigid body) second and third modes of the system when started with initial conditions defined by the appropriate eigenvector. Equation (5.51) is repeated below to show the form of the equation for **x**(t) that is used in the code.

$$\begin{aligned}\mathbf{x}(t) &= e^{\sigma_{n1}t}\left(e^{j\omega_{n1}t}\mathbf{x}_{n1} + e^{-j\omega_{n2}t}\mathbf{x}_{n2}\right)\\ &= e^{\sigma_{n1}t}\left(e^{j\omega_{n1}t}\mathbf{x}_{n1}\right) + e^{\sigma_{n1}t}\left(e^{-j\omega_{n2}t}\mathbf{x}_{n2}\right)\end{aligned} \tag{5.54}$$

The real and imaginary components of the eigenvalues are calculated to give σ and ω in the equation above. The real and imaginary displacements of each of the three masses are then calculated for both oscillatory modes for a time period of 15 seconds.

```
%       calculate the motions of the three masses for all three modes - damped case

        t = 0:.12:15;
```

```
sigma11 = real(lambdao(1));     % sigma for first eigenvalue for mode 1
omega11 = imag(lambdao(1));     % omega for first eigenvalue for mode 1

sigma12 = real(lambdao(2));     % sigma for second eigenvalue for mode 1
omega12 = imag(lambdao(2));     % omega for second eigenvalue for mode 1

sigma21 = real(lambdao(3));     % sigma for first eigenvalue for mode 2
omega21 = imag(lambdao(3));     % omega for first eigenvalue for mode 2

sigma22 = real(lambdao(4));     % sigma for second eigenvalue for mode 2
omega22 = imag(lambdao(4));     % omega for second eigenvalue for mode 2

sigma31 = real(lambdao(5));     % sigma for first eigenvalue for mode 3
omega31 = imag(lambdao(5));     % omega for first eigenvalue for mode 3

sigma32 = real(lambdao(6));     % sigma for second eigenvalue for mode 3
omega32 = imag(lambdao(6));     % omega for second eigenvalue for mode 3

%       motion of three masses for mode 1

z111r = exp(sigma11*t).*(exp(i*omega11*t)*xmon1(1,1));      % mass 1
z112r = exp(sigma12*t).*(exp(i*omega12*t)*xmon1(1,2));      % mass 1

z121r = exp(sigma11*t).*(exp(i*omega11*t)*xmon1(3,1));      % mass 2
z122r = exp(sigma12*t).*(exp(i*omega12*t)*xmon1(3,2));      % mass 2

z131r = exp(sigma11*t).*(exp(i*omega11*t)*xmon1(5,1));      % mass 3
z132r = exp(sigma12*t).*(exp(i*omega12*t)*xmon1(5,2));      % mass 3

%       motion of three masses for mode 2

z211r = exp(sigma21*t).*(exp(i*omega21*t)*xmon1(1,3));      % mass 1
z212r = exp(sigma22*t).*(exp(i*omega22*t)*xmon1(1,4));      % mass 1

z221r = exp(sigma21*t).*(exp(i*omega21*t)*xmon1(3,3));      % mass 2
z222r = exp(sigma22*t).*(exp(i*omega22*t)*xmon1(3,4));      % mass 2

z231r = exp(sigma21*t).*(exp(i*omega21*t)*xmon1(5,3));      % mass 3
z232r = exp(sigma22*t).*(exp(i*omega22*t)*xmon1(5,4));      % mass 3

%       motion of three masses for mode 3

z311r = exp(sigma31*t).*(exp(i*omega31*t)*xmon1(1,5));      % mass 1
z312r = exp(sigma32*t).*(exp(i*omega32*t)*xmon1(1,6));      % mass 1

z321r = exp(sigma31*t).*(exp(i*omega31*t)*xmon1(3,5));      % mass 2
z322r = exp(sigma32*t).*(exp(i*omega32*t)*xmon1(3,6));      % mass 2

z331r = exp(sigma31*t).*(exp(i*omega31*t)*xmon1(5,5));      % mass 3
z332r = exp(sigma32*t).*(exp(i*omega32*t)*xmon1(5,6));      % mass 3
```

5.14 Plotting Initial Condition Response, Listing

The code listing below is to plot various combinations of real and imaginary components of the displacements of the three masses when released in states which match the eigenvectors.

```
%       plot real and imaginary motions of each mass for the two complex conjugate
%       eigenvectors of mode 2

        plot(t,real(z211),'k-',t,real(z212),'k+-',t,imag(z211),'k.-',t,imag(z212),'ko-')
        title('non-prop damped real and imag for z1, mode 2')
        legend('real','real','imag','imag')
        xlabel('time, sec')
        axis([0 max(t) -1 1])
        grid on

        disp('execution paused to display figure, "enter" to continue'); pause

        plot(t,real(z221),'k-',t,real(z222),'k+-',t,imag(z221),'k.-',t,imag(z222),'ko-')
        title('non-prop damped real and imag for z2 mode 2')
        legend('real','real','imag','imag')
        xlabel('time, sec')
        axis([0 max(t) -1 1])
        grid on

        disp('execution paused to display figure, "enter" to continue'); pause

        plot(t,real(z231),'k-',t,real(z232),'k+-',t,imag(z231),'k.-',t,imag(z232),'ko-')
        title('non-prop damped real and imag for z3 mode 2')
        legend('real','real','imag','imag')
        xlabel('time, sec')
        axis([0 max(t) -1 1])
        grid on

        disp('execution paused to display figure, "enter" to continue'); pause

        plot(t,real(z211+z212),'k-',t,real(z221+z222),'k+-',t,real(z231+z232),'k.-')
        title('non-prop damped, z1, z2, z3 mode 2')
        legend('mass 1','mass 2','mass 3')
        xlabel('time, sec')
        axis([0 max(t) -2 2])
        grid on

        disp('execution paused to display figure, "enter" to continue'); pause

%       plot subplots for notes

        subplot(2,2,1)
        plot(t,real(z211),'k-',t,real(z212),'k+',t,imag(z211),'k.-',t,imag(z212),'ko-')
        title('non-prop damped real and imag for z1, mode 2')
        legend('real','real','imag','imag')
        axis([0 max(t) -1 1])
        grid on
```

```
        subplot(2,2,2)
        plot(t,real(z221),'k-',t,real(z222),'k+',t,imag(z221),'k.-',t,imag(z222),'ko-')
        title('non-prop damped real and imag for z2 mode 2')
        legend('real','real','imag','imag')
        axis([0 max(t) -1 1])
        grid on

        subplot(2,2,3)
        plot(t,real(z231),'k-',t,real(z232),'k+',t,imag(z231),'k.-',t,imag(z232),'ko-')
        title('non-prop damped real and imag for z3 mode 2')
        legend('real','real','imag','imag')
        xlabel('time, sec')
        axis([0 max(t) -1 1])
        grid on

        subplot(2,2,4)
        plot(t,real(z211+z212),'k-',t,real(z221+z222),'k+-',t,real(z231+z232),'k.-')
        title('non-prop damped, z1, z2, z3 mode 2')
        legend('mass 1','mass 2','mass 3')
        grid on
        xlabel('time, sec')
        axis([0 max(t) -2 2])

        disp('execution paused to display figure, "enter" to continue'); pause

        subplot(1,1,1)

%       plot mode 3

        plot(t,real(z311),'k-',t,real(z312),'k+-',t,imag(z311),'k.-',t,imag(z312),'ko-')
        title('non-prop damped real and imag for z1, mode 3')
        legend('real','real','imag','imag')
        xlabel('time, sec')
        axis([0 max(t) -1 1])
        grid on

        disp('execution paused to display figure, "enter" to continue'); pause

        plot(t,real(z321),'k-',t,real(z322),'k+-',t,imag(z321),'k.-',t,imag(z322),'ko-')
        title('non-prop damped real and imag for z2 mode 3')
        legend('real','real','imag','imag')
        xlabel('time, sec')
        axis([0 max(t) -2 2])
        grid on

        disp('execution paused to display figure, "enter" to continue'); pause

        plot(t,real(z331),'k-',t,real(z332),'k+-',t,imag(z331),'k.-',t,imag(z332),'ko-')
        title('non-prop damped real and imag for z3 mode 3')
        legend('real','real','imag','imag')
        xlabel('time, sec')
        axis([0 max(t) -1 1])
        grid on

        disp('execution paused to display figure, "enter" to continue'); pause

        plot(t,real(z311+z312),'k-',t,real(z321+z322),'k+-',t,real(z331+z332),'k.-')
```

```
        title('non-prop damped, z1, z2, z3 mode 3')
        legend('mass 1','mass 2','mass 3')
        xlabel('time, sec')
        axis([0 max(t) -4 4])
        grid on

        disp('execution paused to display figure, "enter" to continue'); pause

%       plot subplots for notes

        subplot(2,2,1)
        plot(t,real(z311),'k-',t,real(z312),'k+-',t,imag(z311),'k.-',t,imag(z312),'ko-')
        title('non-prop damped real and imag for z1, mode 3')
        legend('real','real','imag','imag')
        axis([0 max(t) -1 1])
        grid on

        subplot(2,2,2)
        plot(t,real(z321),'k-',t,real(z322),'k+-',t,imag(z321),'k.-',t,imag(z322),'ko-')
        title('non-prop damped real and imag for z2 mode 3')
        legend('real','real','imag','imag')
        axis([0 max(t) -2 2])
        grid on

        subplot(2,2,3)
        plot(t,real(z331),'k-',t,real(z332),'k+-',t,imag(z331),'k.-',t,imag(z332),'ko-')
        title('non-prop damped real and imag for z3 mode 3')
        legend('real','real','imag','imag')
        xlabel('time, sec')
        axis([0 max(t) -1 1])
        grid on

        subplot(2,2,4)
        plot(t,real(z311+z312),'k-',t,real(z321+z322),'k+-',t,real(z331+z332),'k.-')
        title('non-prop damped, z1, z2, z3 mode 3')
        legend('mass 1','mass 2','mass 3')
        xlabel('time, sec')
        axis([0 max(t) -4 4])
        grid on

        disp('execution paused to display figure, "enter" to continue'); pause
```

5.15 Plotted Results: Argand and Initial Condition Responses

The next four sections plot Argand and initial condition transient responses for the two oscillatory modes, illustrating the canceling of the imaginary components and the doubling of the real components.

5.15.1 Argand Diagram, Mode 2

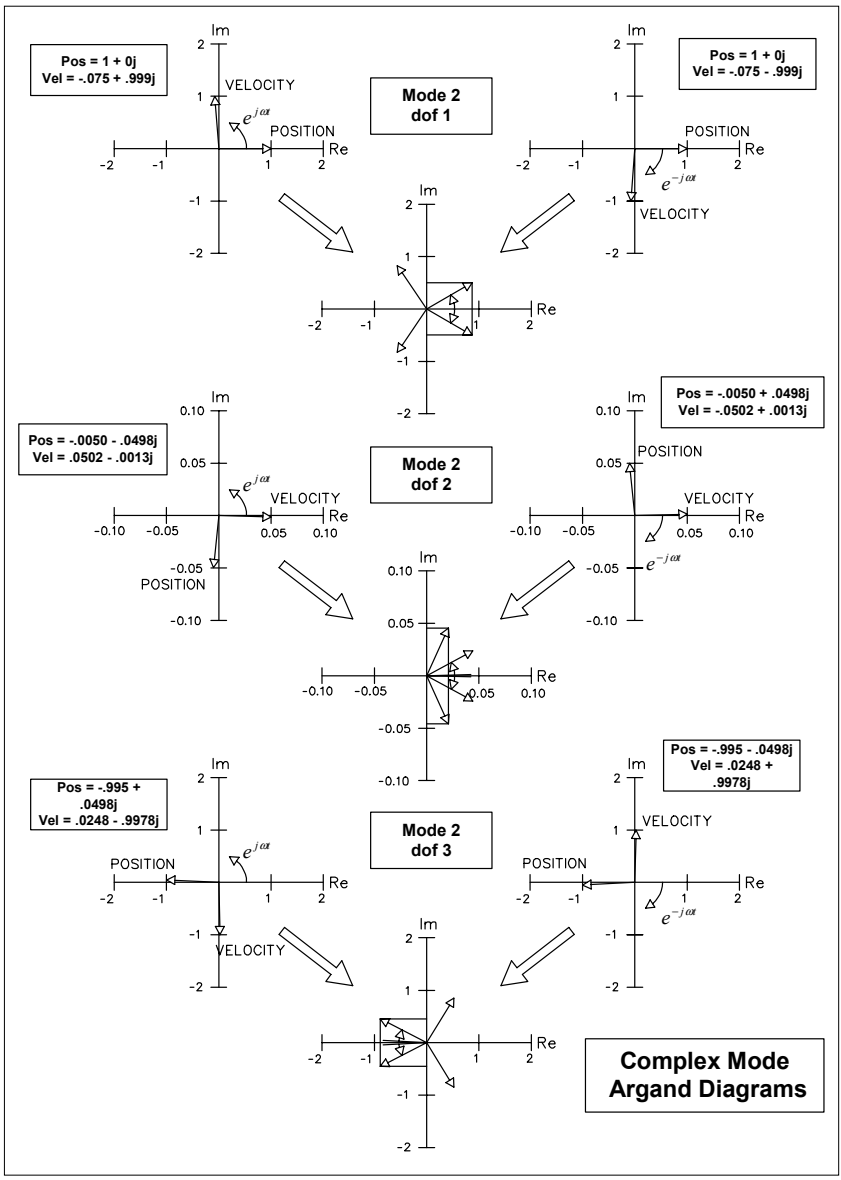

Figure 5.8 Argand diagram for three degrees of freedom for mode 2, complex damping.

5.15.2 Time Domain Responses, Mode 2

The plots below show the motions of the masses decreasing due to the damping. Once again, the imaginary components are out of phase and cancel each other, leaving only twice the real component as the final motion. Unlike the undamped case, the three masses do not reach their maximum or minimum positions at the same time. Since the damping is quite small, it is hard to see on the plots the small differences in times at which the maxima and minima are reached. Note that the unequal damping values for the two dampers make the center mass have a small motion in mode 2. We showed in Chapter 3 that for the undamped case mass 2 has no motion for mode 2.

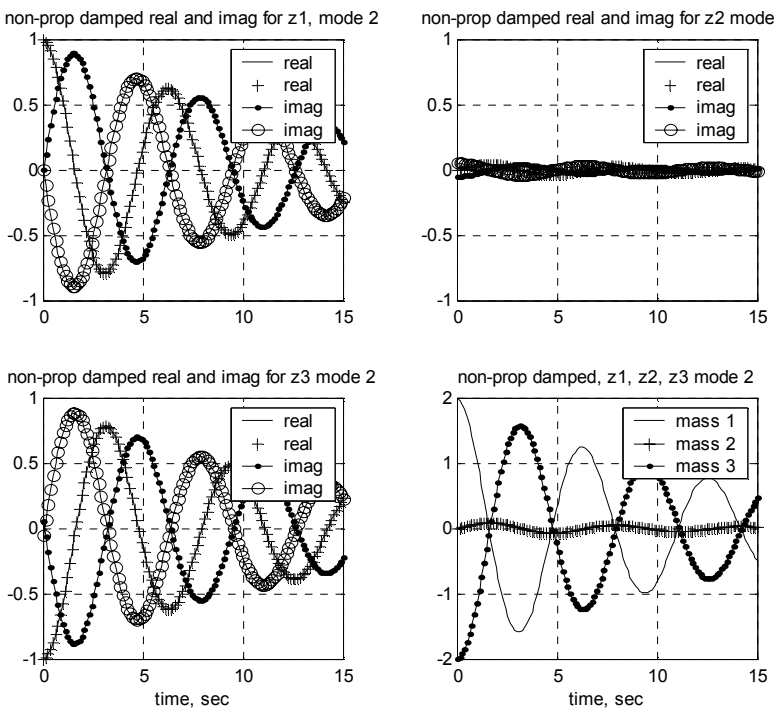

Figure 5.9: Initial condition transient response for mode 2.

5.15.3 Argand Diagram, Mode 3

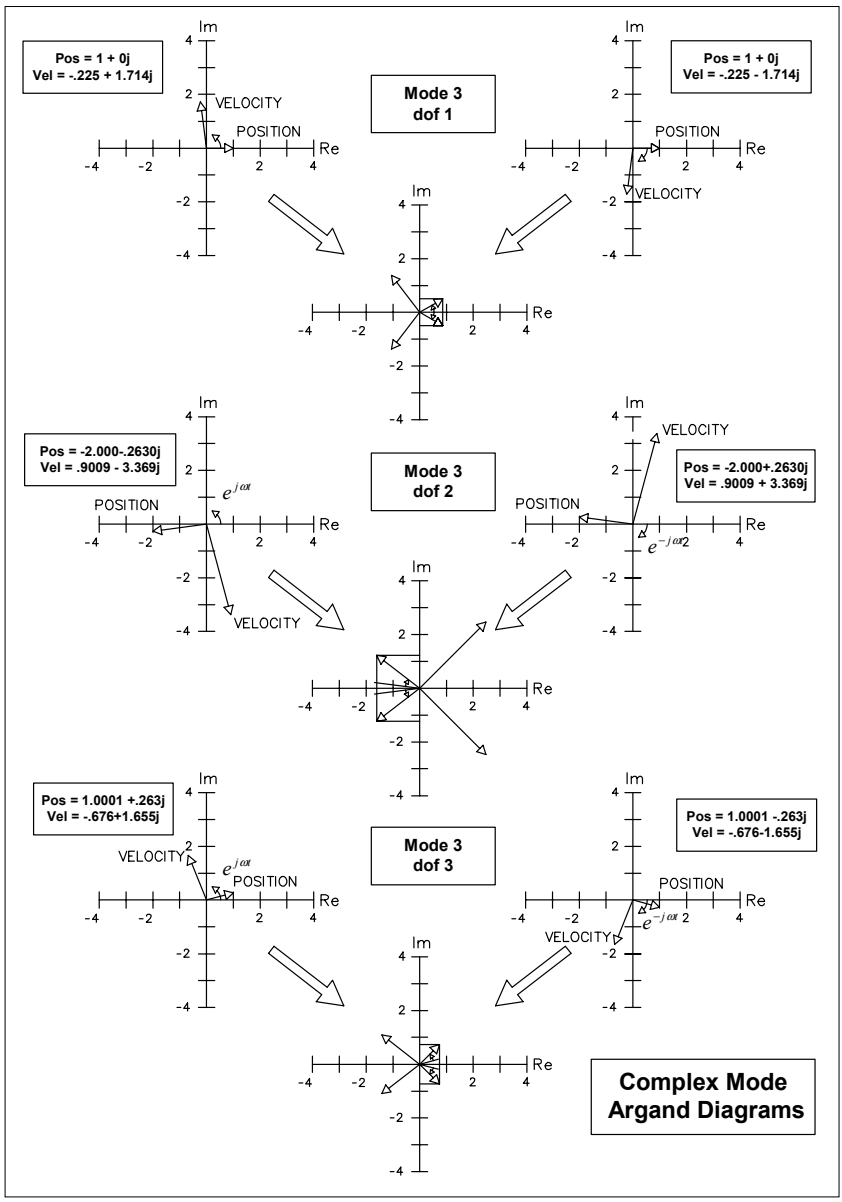

Figure 5.10: Argand diagram for three degrees of freedom for mode 3, complex damping.

5.15.4 Time Domain Responses, Mode 3

Compared to the responses for the mode 2 in Figure 5.9, the response for mode 3 damps out faster for two reasons. First, it has higher damping, 13% versus 7.5%, as shown in Figure 5.5. Secondly, even if zeta were the same for the two modes, the higher frequency of mode 3 will create higher velocities, hence higher damping from the velocity-dependent damping term.

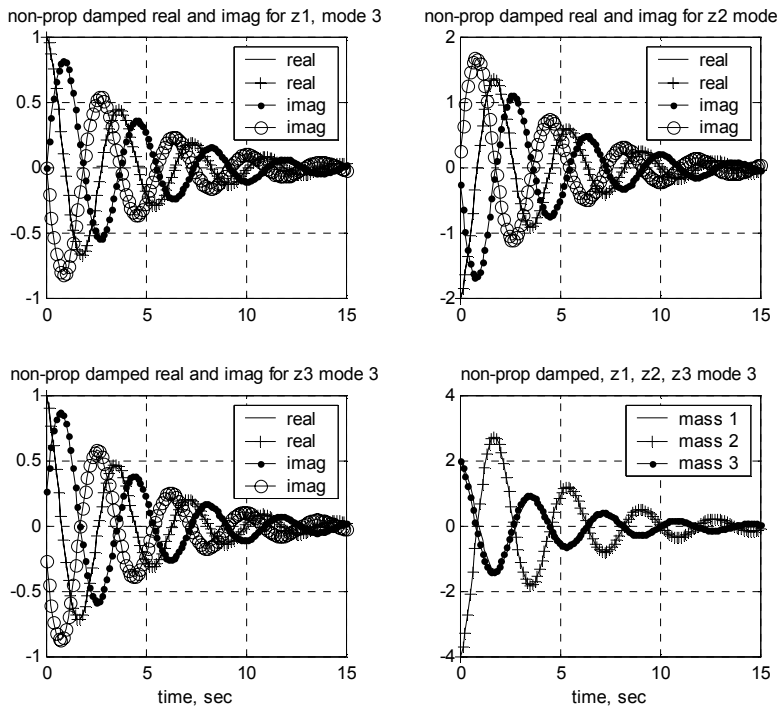

Figure 5.11: Initial condition transient response for mode 3.

Problems

Note: All the problems refer to the two dof system shown in Figure P2.2.

P5.1 Write the damped equations for the two dof system in state space form, both expanded and matrix. Show the input matrix **B** for a step force of magnitude 1 to mass 1 and magnitude –2 for mass 2. Show the output matrix **C** for the following outputs:

a) Position of masses 1 and 2

b) Position and velocity of mass 1

c) 2 times velocity of mass 1 plus 3 times the position of mass 2

P5.2 Set up the eigenvalue problem for the damped two dof problem as in (5.46).

P5.3 (MATLAB) With $m_1 = m_2 = m = 1$, $k_1 = k_2 = k = 1$, modify the code in **tdof_non_prop_damped.m** for the two dof damped model with $c_1 = c_2 = 0.1$ and:

a) list the complex eigenvalues, real and imaginary form

b) list the complex eigenvalues, magnitude and phase angle form

c) normalize the eigenvectors for unity values of the position of mass 1 and hand plot the Argand diagrams for the system

d) list the percentage of critical damping for each mode

e) plot the complex eigenvalues in the s-plane and correlate the three different descriptions in (a), (b) and (d)

P5.4 (MATLAB) Set $m_1 = m_2 = m = 1$, $k_1 = k_2 = k = 1$ and plot the initial condition responses for the system in initial conditions which match the two damped eigenvectors.

P5.5 Set $m_1 = m_2 = m = 1$, $k_1 = k_2 = k = 1$ and hand plot the Argand diagrams for modes 1 and 2.