CHAPTER 2

TRANSFER FUNCTION ANALYSIS

2.1 Introduction

The purpose of this chapter is to illustrate how to derive equations of motion for Multi Degree of Freedom (mdof) systems and how to solve for their transfer functions.

The chapter starts by developing equations of motion for a specific **three degree of freedom** damped system (indicated throughout the book by the acronym "**tdof**"). A systematic method of creating "global" mass, damping and stiffness matrices is borrowed from the stiffness method of matrix structural analysis. The tdof model will be used for the various analysis techniques through most of the book, providing a common thread that links the pieces into a whole.

Two additional examples are used to illustrate the method for building matrix equations of motion. The first is a lumped mass **six degree of freedom** (**6dof**) system for which the stiffness matrix is developed. The second is a simplified rotary actuator system from a disk drive, for which the complete undamped equations of motion are developed.

Following the equations of motion sections, the chapter continues with a review of the transfer function and frequency response analyses of a **single degree of freedom** (**sdof**) damped example. After developing the closed form solution of the equations, MATLAB code is used to calculate and plot magnitude and phase versus frequency for a range of damping values.

The tdof model is then reintroduced and Laplace transforms are used to develop its transfer functions. In order to facilitate hand calculations of poles and zeros, damping is set to zero. The characteristic equation, poles and zeros are then defined and calculated in closed form. MATLAB code is used to plot the pole/zero locations for the nine transfer functions using MATLAB's "pzmap" command.

MATLAB is used to calculate and plot poles and zeros for values of damping greater than zero and we will see that additional real values zeros start appearing as damping is increased from zero. The significance of the real axis zeros is discussed.

2.2 Deriving Matrix Equations of Motion

2.2.1 Three Degree of Freedom (tdof) System, Identifying Components and Degrees of Freedom

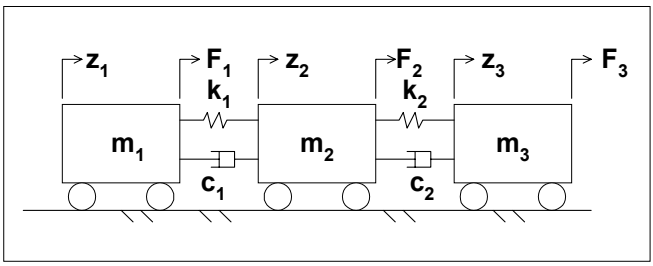


Figure 2.1: tdof system schematic.

The first step in analyzing a mechanical system is to sketch the system, showing the degrees of freedom, the masses, stiffnesses and damping present, and showing applied forces. The tdof system to be followed throughout the book, shown in Figure 2.1, consists of three masses, numbered 1 to 3, two springs between the masses and two dampers also between the masses. The model is purposely not connected to ground to allow a "rigid body" degree of freedom, meaning that at "low" frequencies the set of three masses can all move in one direction or the other as a single rigid body, with no relative motion between them.

The number of **degrees of freedom** (**dof**) for a model is the number of geometrically independent coordinates required to specify the configuration for the model. For consistency, the notation "z" will be used for degrees of freedom, saving "x" and "y" for state space representations later in the book. For the system shown in Figure 2.1 where each mass can move only along the z axis, a single degree of freedom for each mass is sufficient, hence the degrees of freedom z_1, z_2 and z_3.

2.2.2 Defining the Stiffness, Damping and Mass Matrices

The equations of motion will be derived in matrix form using a method derived from the stiffness method of structural analysis, as follows:

> **Stiffness Matrix: Apply a unit displacement** to each dof, one at a time. Constrain the dof's not displaced and **define the stiffness dependent constraint force** required for all dof's to hold the system in the constrained position.

The row elements of each column of the stiffness matrix are then defined by the constraints associated with each dof that are required to hold the system in the constrained position.

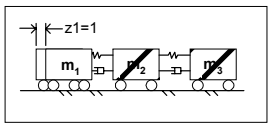
Damping Matrix: Apply a unit velocity to each dof, one at a time. Constrain the dof's not moving and **define the velocity-dependent constraint force** required to keep the system in that state.

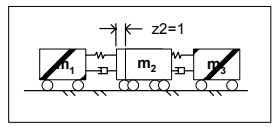
The row elements of each column of the damping matrix are then defined by the constraints associated with each dof that are required to keep the system in that state – with one dof moving with constant velocity and all the other dof's not moving.

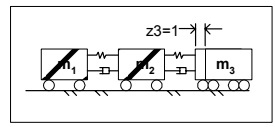
Mass Matrix: Apply a unit acceleration to each dof, one at a time. Constrain the dof's not being accelerated and **define the acceleration-dependent constraint forces** required.

The row elements of each column of the mass matrix are then defined by the constraints associated with keeping one dof accelerating at a constant rate and the other dof's stationary. Since in this model the only forces transmitted between the masses are proportional to displacement (the springs) and velocity (viscous damping), no forces are transmitted between masses due to one of the masses accelerating. This leads to a diagonal mass matrix in cases where the origin of the coordinate systems are taken through the center of mass of the bodies and the coordinate axes are aligned with the principal moments of inertia of the body.

Table 2.1 shows how the three matrices are filled out. To fill out column 1 of the mass, damping and stiffness matrices, mass 1 is given a unit acceleration, velocity and displacement, respectively. Then the constraining forces required to keep the system in that state are defined for each dof, where row 1 is for dof 1, row 2 is for dof 2 and row 3 is for dof 3.







Column 1	Column 2	Column 3
$\text{UNIT}\begin{Bmatrix}\text{accel}\\ \text{vel}\\ \text{disp}\end{Bmatrix}\text{dof1}$	$\text{Unit}\begin{Bmatrix}\text{accel}\\ \text{vel}\\ \text{disp}\end{Bmatrix}\text{dof2}$	$\text{Unit}\begin{Bmatrix}\text{accel}\\ \text{vel}\\ \text{disp}\end{Bmatrix}\text{dof3}$
$\begin{bmatrix} m_1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{matrix} 0 \\ m_2 \\ 0 \end{matrix}$	$\begin{matrix} 0 \\ 0 \\ m_3 \end{matrix}\begin{matrix}\text{dof1}\\ \text{dof2}\\ \text{dof3}\end{matrix}$
$\begin{bmatrix} c_1 \\ -c_1 \\ 0 \end{bmatrix}$	$\begin{matrix} -c_1 \\ c_1 + c_2 \\ -c_2 \end{matrix}$	$\begin{matrix} 0 \\ -c_2 \\ c_2 \end{matrix}\begin{matrix}\text{dof1}\\ \text{dof2}\\ \text{dof3}\end{matrix}$
$\begin{bmatrix} k_1 \\ -k_1 \\ 0 \end{bmatrix}$	$\begin{matrix} -k_1 \\ k_1 + k_2 \\ -k_2 \end{matrix}$	$\begin{matrix} 0 \\ -k_2 \\ k_2 \end{matrix}\begin{matrix}\text{dof1}\\ \text{dof2}\\ \text{dof3}\end{matrix}$

Table 2.1: m, c, k columns and associated dof displacements. The cross-hatched masses in the figures above each column are constrained and non-cross-hatched mass is moved a unit displacement.

The general matrix form for a tdof system is shown below, where the "ij" subscripts in m_{ij}, c_{ij}, k_{ij} are defined as follows: "i" is the row number and "j" is the column number.

$$\begin{matrix} & \text{j=1} & \text{j=2} & \text{j=3} \\ \text{i=1} \\ \text{i=2} \\ \text{i=3} \end{matrix}\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}\begin{bmatrix} \ddot{z}_1 \\ \ddot{z}_2 \\ \ddot{z}_3 \end{bmatrix} + \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} + \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix}\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} \qquad (2.1)$$

Mass Damping Stiffness

Expanding the matrix equations of motion by multiplying across and down:

$$m_{11}\ddot{z}_1 + m_{12}\ddot{z}_2 + m_{13}\ddot{z}_3 + c_{11}\dot{z}_1 + c_{12}\dot{z}_2 + c_{13}\dot{z}_3 + k_{11}z_1 + k_{12}z_2 + k_{13}z_3 = F_1 \quad (2.2)$$

$$m_{21}\ddot{z}_1 + m_{22}\ddot{z}_2 + m_{23}\ddot{z}_3 + c_{21}\dot{z}_1 + c_{22}\dot{z}_2 + c_{23}\dot{z}_3 + k_{21}z_1 + k_{22}z_2 + k_{23}z_3 = F_2 \quad (2.3)$$

$$m_{31}\ddot{z}_1 + m_{32}\ddot{z}_2 + m_{33}\ddot{z}_3 + c_{31}\dot{z}_1 + c_{32}\dot{z}_2 + c_{33}\dot{z}_3 + k_{31}z_1 + k_{32}z_2 + k_{33}z_3 = F_3 \quad (2.4)$$

The matrix equations of motion for our tdof problem, from Table 2.1, is:

$$\begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix}\begin{bmatrix} \ddot{z}_1 \\ \ddot{z}_2 \\ \ddot{z}_3 \end{bmatrix} + \begin{bmatrix} c_1 & -c_1 & 0 \\ -c_1 & (c_1+c_2) & -c_2 \\ 0 & -c_2 & c_2 \end{bmatrix}\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} + \begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & (k_1+k_2) & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix}\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} \quad (2.5)$$

Expanding:

$$\begin{aligned} m_1\ddot{z}_1 + c_1\dot{z}_1 - c_1\dot{z}_2 + k_1z_1 - k_1z_2 &= F_1 \\ m_2\ddot{z}_2 - c_1\dot{z}_1 + (c_1+c_2)\dot{z}_2 - c_2\dot{z}_3 - k_1z_1 + (k_1+k_2)z_2 - k_2z_3 &= F_2 \\ m_3\ddot{z}_3 - c_2\dot{z}_2 + c_2\dot{z}_3 - k_2z_2 + k_2z_3 &= F_3 \end{aligned} \quad (2.6a,b,c)$$

2.2.3 Checks on Equations of Motion for Linear Mechanical Systems

Two quick checks which should always be carried out for linear mechanical systems are the following:

1) All diagonal terms must be positive.

2) The mass, damping and stiffness matrices must be symmetrical. For example $k_{ij} = k_{ji}$ for the stiffness matrix.

2.2.4 Six Degree of Freedom (6dof) Model – Stiffness Matrix

The stiffness matrix development for a more complicated model than the tdof model used so far is shown below. The figure below shows a 6dof system with a rigid body mode and no damping.

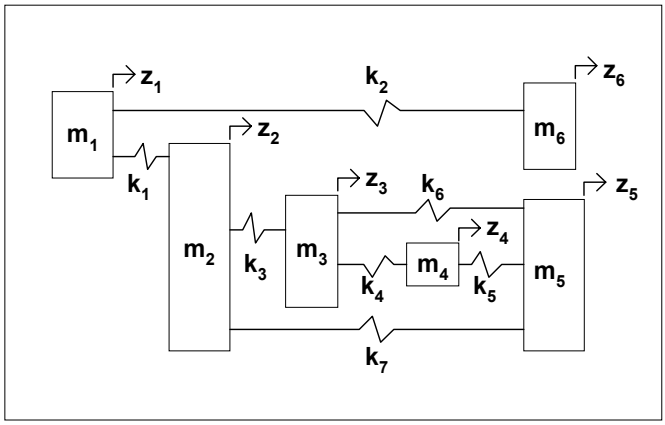


Figure 2.2: 6dof model schematic.

Moving each dof a unit displacement and then writing down the reaction forces to constrain that configuration for each of the column elements, the stiffness matrix for this example can be written by inspection as shown in Table 2.2. Note that the symmetry and positive diagonal checks are satisfied.

$$\begin{bmatrix} (k_1+k_2) & -k_1 & 0 & 0 & 0 & -k_2 \\ -k_1 & (k_1+k_3+k_7) & -k_3 & 0 & -k_7 & 0 \\ 0 & -k_3 & (k_3+k_4+k_6) & -k_4 & -k_6 & 0 \\ 0 & 0 & -k_4 & (k_4+k_5) & -k_5 & 0 \\ 0 & -k_7 & -k_6 & -k_5 & (k_5+k_6+k_7) & 0 \\ -k_2 & 0 & 0 & 0 & 0 & k_2 \end{bmatrix}$$

Table 2.2: Stiffness matrix terms for 6dof system.

2.2.5 Rotary Actuator Model – Stiffness and Mass Matrices

The technique is also applicable to systems with rotations combined with translations, as long as rotations are kept small. The system shown below represents a simplified rotary actuator from a disk drive that pivots about its mass center, has force applied at the left-hand end (representing the rotary voice coil motor) and has a "recording head" m_2 at the right-hand end. The "head" is connected to the end of the actuator with a spring and the pivot bearing is connected to ground through the radial stiffness of its bearing.

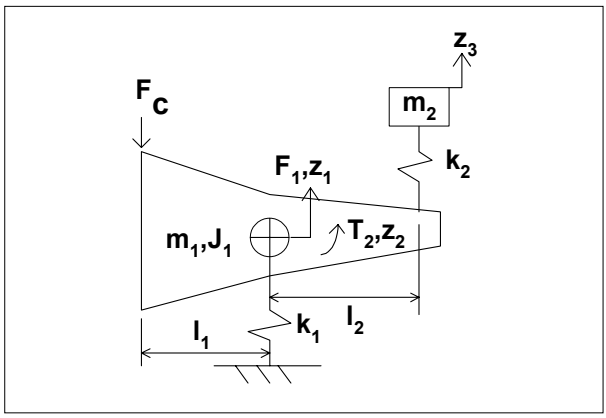


Figure 2.3: Rotary actuator schematic.

Starting off by defining the degrees of freedom, stiffnesses, mass and inertia terms:

dof:

z_1	translation of actuator
z_2	rotation of actuator
z_3	translation of head

Stiffnesses:

k_1	actuator bearing radial stiffness
k_2	"suspension" stiffness

Inertias:

m_1, J_1	actuator mass, inertia
m_2	"head" mass

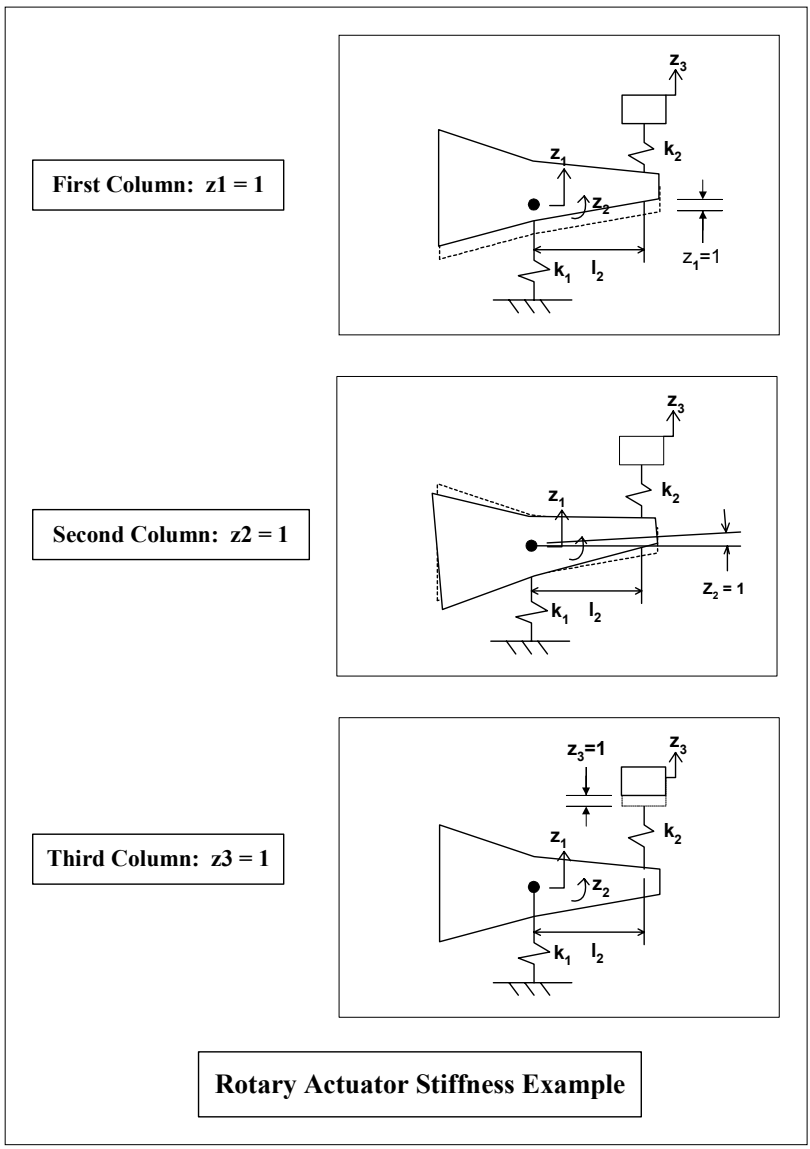


Figure 2.4: Unit displacements to define mass and stiffeness matrices.

See Figure 2.4 to define the entries of each column of (2.7), the forces/moments required to constrain the respective dof in the configuration shown.

$$\begin{bmatrix} m_1 & 0 & 0 \\ 0 & J_1 & 0 \\ 0 & 0 & m_2 \end{bmatrix} \begin{bmatrix} \ddot{z}_1 \\ \ddot{z}_2 \\ \ddot{z}_3 \end{bmatrix} + \begin{bmatrix} (k_1 + k_2) & l_2 k_2 & -k_2 \\ l_2 k_2 & l_2^2 k_2 & -l_2 k_2 \\ -k_2 & -l_2 k_2 & k_2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} F_1 \\ T_2 \\ 0 \end{bmatrix} = \begin{bmatrix} -F_c \\ F_c l_1 \\ 0 \end{bmatrix} \tag{2.7}$$

$$F_1 = -F_c \tag{2.8}$$

$$T_2 = F_c l_1 \tag{2.9}$$

2.3 Single Degree of Freedom (sdof) System Transfer Function and Frequency Response

2.3.1 sdof System Definition, Equations of Motion

The sdof system to be analyzed is shown below. The system consists of a mass, m, connected to ground by a spring of stiffness k and a damper with viscous damping coefficient c. Since the mass can only move in the z direction, a single degree of freedom is sufficient to define the system configuration. Force F is applied to the mass.

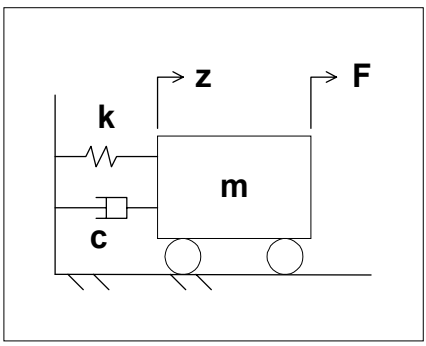


Figure 2.5: Single degree of freedom system.

The equation of motion for this system is given by:

$$m\ddot{z} + c\dot{z} + kz = F \tag{2.10}$$

2.3.2 Transfer Function

Taking the Laplace transform of a general second order differential equation (DE) with initial conditions is:

Second Order DE: $\mathcal{L}\{\ddot{z}(t)\} = s^2 z(s) - sz(0) - \dot{z}(0)$, (2.11)

where $z(0)$ and $\dot{z}(0)$ are position and velocity initial conditions, respectively, and $z(s)$ is the Laplace transform of $z(t)$. See Appendix 2 for more on Laplace transforms.

Because we are taking a transfer function, representing the steady state response of the system to a sinusoidal input, initial conditions are set to zero, leaving

$$\mathcal{L}\{\ddot{z}(t)\} = s^2 z(s) \tag{2.12}$$

The Laplace transform of the sdof equation of motion (2.10), where $F(s)$ represents the Laplace transform of F, is:

$$ms^2 z(s) + csz(s) + kz(s) = F(s) \tag{2.13}$$

Solving for the transfer function:

$$\frac{z(s)}{F(s)} = \frac{1}{ms^2 + cs + k} = \frac{1/m}{s^2 + \frac{c}{m}s + \frac{k}{m}} \tag{2.14}$$

We can simplify the equation above by applying the following definitions:

1) $\omega_n^2 = \frac{k}{m}$, where ω_n is the undamped natural frequency, rad/sec

2) $c_{cr} = 2\sqrt{km}$, where c_{cr} is the "critical" damping value

3) ζ is the amount of proportional damping, typically stated as a percentage of critical damping

4) $2\zeta\omega_n$ is the multiplier of the velocity term, $\dot{z}$, developed below:

$$\begin{aligned}\frac{c}{m} &= 2\zeta\omega_n \\ &= 2\frac{c}{c_{cr}}\sqrt{\frac{k}{m}} \\ &= \frac{2c}{2\sqrt{km}}\frac{\sqrt{k}}{\sqrt{m}} \\ &= \frac{c}{m}\end{aligned} \tag{2.15}$$

Rewriting, using the above substitutions:

$$\frac{z(s)}{F(s)} = \frac{1/m}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{2.16}$$

2.3.3 Frequency Response

Substituting " $j\omega$ " for "s" to calculate the frequency response, where "j" is the imaginary operator:

$$\begin{aligned}\frac{z(j\omega)}{F(j\omega)} &= \frac{1/m}{(j\omega)^2 + 2\zeta\omega_n(j\omega) + \omega_n^2} \\ &= \frac{1/m}{-\omega^2 + 2\zeta\omega\omega_n j + \omega_n^2} \\ &= \frac{1/(m\omega^2)}{-1 + \frac{2\zeta\omega_n j}{\omega} + \frac{\omega_n^2}{\omega^2}} \\ &= \frac{1/(m\omega^2)}{\left(\frac{\omega_n^2}{\omega^2} - 1\right) + \frac{2\zeta\omega_n j}{\omega}} \\ &= \frac{1/(m\omega^2)}{\left[\left(\frac{\omega_n}{\omega}\right)^2 - 1\right] + j2\zeta\left(\frac{\omega_n}{\omega}\right)}\end{aligned} \tag{2.17a,b,c,d,e}$$

The frequency response equation above shows how the ratio (z/F) varies as a function of frequency, ω. The ratio is a complex number that has some interesting properties at different values of the ratio (ω_n / ω).

At low frequencies relative to the resonant frequency, $\omega_n^2 >> \omega\omega_n >> \omega^2$, and the transfer function is given by:

$$\begin{aligned}\frac{z(j\omega)}{F(j\omega)} &= \frac{1/m}{-\omega^2 + 2\zeta\omega\omega_n j + \omega_n^2} \\ &\cong \frac{1/m}{\omega_n^2} = \frac{1}{m\omega_n^2} = \frac{1}{m\left(\frac{k}{m}\right)} = \frac{1}{k}\end{aligned} \tag{2.18}$$

Since the frequency response value at any frequency is a complex number, we can take the magnitude and phase.

$$\begin{aligned}\left|\frac{z(j\omega)}{F(j\omega)}\right| &= \frac{1}{k} \\ \angle\frac{z(j\omega)}{F(j\omega)} &= 0\end{aligned} \tag{2.19a,b}$$

Thus, the gain at low frequencies is a constant, (1/k) or the inverse of the stiffness. Phase is 0° because the sign is positive.

At high frequencies, $\omega^2 >> \omega\omega_n >> \omega_n^2$, the transfer function is given by:

$$\begin{aligned}\frac{z(j\omega)}{F(j\omega)} &= \frac{1/m}{-\omega^2 + 2\zeta\omega\omega_n j + \omega_n^2} \\ &\cong \frac{1/m}{-\omega^2} = \frac{-1}{m\omega^2}\end{aligned} \tag{2.20}$$

Once again, taking the magnitude and phase:

$$\begin{aligned}\left|\frac{z(j\omega)}{F(j\omega)}\right| &= \left|\frac{-1}{m\omega^2}\right| = \frac{1}{m\omega^2} \\ \angle\frac{z(j\omega)}{F(j\omega)} &= -180^\circ\end{aligned} \tag{2.21a,b}$$

At high frequencies, the gain is given by $1/(m\omega^2)$ and the phase is -180° because the sign is negative.

At resonance, $\omega = \omega_n$, the transfer function is given by:

$$\frac{z(j\omega)}{F(j\omega)} = \frac{1/m}{-\omega^2 + 2\zeta\omega\omega_n j + \omega_n^2} \tag{2.22}$$
$$= \frac{1/m}{2\zeta\omega\omega_n j} = \frac{1/m}{2\zeta\omega_n^2 j} = \frac{1}{2\zeta\omega_n^2 mj} = \frac{1}{\frac{2\zeta kmj}{m}} = \frac{1}{2\zeta kj} = \frac{1/k}{2\zeta j} = \frac{-j/k}{2\zeta}$$

Taking magnitude and phase at resonance:

$$\left|\frac{z(j\omega)}{F(j\omega)}\right| = \left|\frac{-j/k}{2\zeta}\right| = \frac{1/k}{2\zeta}$$

$$\angle\frac{z(j\omega)}{F(j\omega)} = -90^\circ \tag{2.23a,b}$$

The magnitude at resonance is seen to be the gain at low frequency, 1/k, divided by 2ζ. Since ζ is typically a small number, for example 1% of critical damping or 0.01, the magnitude at resonance is seen to be amplified. At resonance the phase angle is -90°.

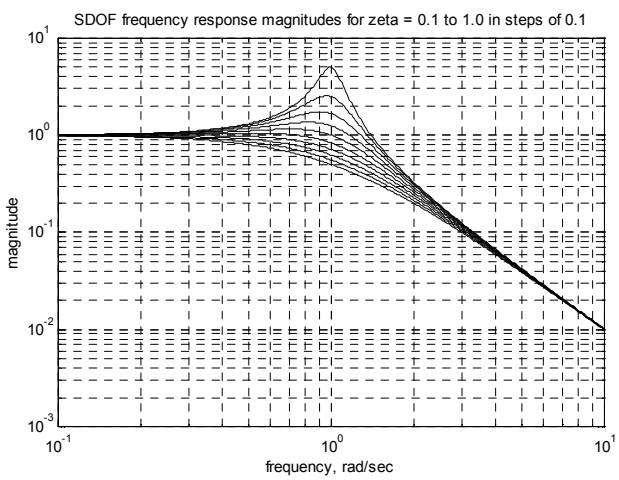


Figure 2.6: sdof magnitude versus frequency for different damping ratios.

The MATLAB code **sdofxfer.m**, listed in the next section, is used to plot the frequency responses from (2.17) for a range of damping values for m = k = 1.0, shown in Figures 2.6 and 2.7. These m and k values give a ω_n value of 1.0 rad/sec.

Since ω_n is 1.0 rad/sec, the resonant peak in Figure 2.6 should occur at that frequency. The low frequency magnitude was shown above to be equal to 1/k = 1.0. The curves for all the damping values approach 1.0 ($10^0 = 1.0$) at low frequencies. At high frequencies the magnitude is given by $1/\left(m\omega^2\right)$, and since m = 1, we should have magnitude of $1/\omega^2$. Checking the plot above, at a frequency of 10 rad/sec, the magnitude should be 1/100 or 0.01.

Note that the slope of the low frequency asymptote is zero, meaning it is not changing with frequency. However, the slope of the high frequency asymptote is "-2," meaning that for every decade increase in frequency the magnitude at high frequency decreases by two orders of magnitude by virtue of the ω^2 term in the denominator. The "-2" slope on a log magnitude versus log frequency plot comes from the following:

$$\log\left|\text{high frequency}\right| \propto \log\left(\frac{1}{\omega^2}\right) = \log\left(\omega^{-2}\right) = -2\log\left(\omega\right) \qquad (2.24)$$

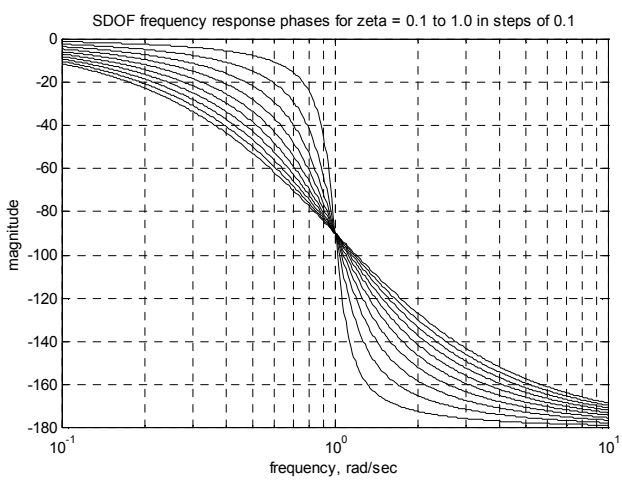


Figure 2.7: sdof phase versus frequency for different damping ratios.

From Figure 2.7, note that at resonance ($\omega_n = 1.0\,\text{rad/sec}$) the phase for all values of damping is -90°. At low frequencies phase is approaching 0° and at high frequencies it is approaching -180°.

2.3.4 MATLAB Code sdofxfer.m Description

The code uses the transfer function form shown in (2.14) to calculate the complex quantity "xfer," where $s = j\omega$, using a vector of defined ω values. Magnitude and phase of the complex value of the transfer function are then plotted versus frequency.

2.3.5 MATLAB Code sdofxfer.m Listing

```
%       sdofxfer.m plotting frequency responses of sdof model for different damping values

        clf;

        clear all;

%       assign values for mass, percentage of critical damping, and stiffnesses
%       zeta is a vector of damping values from 10% to 100% in steps of 10%

        m = 1;
        zeta = 0.1:0.1:1;                       %   0.1 = 10% of critical
        k = 1;

        wn = sqrt(k/m);

%       Define a vector of frequencies to use, radians/sec.  The logspace command uses
%       the log10 value as limits, i.e. -1 is 10^-1 = 0.1 rad/sec, and 1 is
%       10^1 = 10 rad/sec.  The 400 defines 400 frequency points.

        w = logspace(-1,1,400);

%       pre-calculate the radians to degree conversion

        rad2deg = 180/pi;

%       define s as the imaginary operator times the radian frequency vector

        s = j*w;

%       define a for loop to cycle through all the damping values for calculating
%       magnitude and phase

        for  cnt = 1:length(zeta)

%       define the frequency response to be evaluated

                xfer(cnt,:) = (1/m) ./ (s.^2 + 2*zeta(cnt)*wn*s + wn^2);
```

```
%       calculate the magnitude and phase of each frequency response

                mag(cnt,:) = abs(xfer(cnt,:));

                phs(cnt,:) = angle(xfer(cnt,:))*rad2deg;

        end

%       define a for loop to cycle through all the damping values for plotting magnitude

        for  cnt = 1:length(zeta)

        loglog(w,mag(cnt,:),'k-')
        title('SDOF frequency response magnitudes for zeta = 0.1 to 1.0 in steps of 0.1')
        xlabel('frequency, rad/sec')
        ylabel('magnitude')
        grid

        hold on

        end

        hold off

        grid on

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

%       define a for loop to cycle through all the damping values for plotting phase

        for  cnt = 1:length(zeta)

        semilogx(w,phs(cnt,:),'k-')
        title('SDOF frequency response phases for zeta = 0.1 to 1.0 in steps of 0.1')
        xlabel('frequency, rad/sec')
        ylabel('magnitude')
        grid

        hold on

        end

        hold off

        grid on

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

2.4 tdof Laplace Transform, Transfer Functions, Characteristic Equation, Poles, Zeros

We now return to the original tdof model as shown in Figure 2.1. In order to define transfer functions and understand poles and zeros of the system, we need to transform from the time domain to the frequency domain. We do this by taking Laplace transforms of the equations of motion.

2.4.1 Laplace Transforms with Zero Initial Conditions

Repeating (2.5) for the tdof system:

$$\begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix}\begin{bmatrix} \ddot{z}_1 \\ \ddot{z}_2 \\ \ddot{z}_3 \end{bmatrix}+\begin{bmatrix} c_1 & -c_1 & 0 \\ -c_1 & (c_1+c_2) & -c_2 \\ 0 & -c_2 & c_2 \end{bmatrix}\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} + \begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & (k_1+k_2) & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix}\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}=\begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} \tag{2.25}$$

Taking Laplace transforms assuming initial conditions of zero, where z_1, z_2, z_3 now represent the Laplace transforms of the original z_1, z_2, z_3:

$$\begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix}\begin{bmatrix} s^2 z_1 \\ s^2 z_2 \\ s^2 z_3 \end{bmatrix}+\begin{bmatrix} c_1 & -c_1 & 0 \\ -c_1 & (c_1+c_2) & -c_2 \\ 0 & -c_2 & c_2 \end{bmatrix}\begin{bmatrix} sz_1 \\ sz_2 \\ sz_3 \end{bmatrix} + \begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & (k_1+k_2) & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix}\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}=\begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} \tag{2.26}$$

Rearranging:

$$\begin{bmatrix} (m_1 s^2+c_1 s+k_1) & (-c_1 s-k_1) & 0 \\ (-c_1 s-k_1) & (m_2 s^2+c_1 s+c_2 s+k_1+k_2) & (-c_2 s-k_2) \\ 0 & (-c_2 s-k_2) & (m_3 s^2+c_2 s+k_2) \end{bmatrix}\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}=\begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} \tag{2.27}$$

2.4.2 Solving for Transfer Functions

In this section we solve for the nine possible transfer functions for all combinations of degrees of freedom where force is applied and where displacements are taken. Solving for the transfer functions for greater than a 2dof system is a task not to be taken lightly – symbolic algebra programs such as Mathematica, Maple or the MATLAB Symbolic Toolbox should be used.

$\frac{z_1}{F_1}$	$\frac{z_1}{F_2}$	$\frac{z_1}{F_3}$
$\frac{z_2}{F_1}$	$\frac{z_2}{F_2}$	$\frac{z_2}{F_3}$
$\frac{z_3}{F_1}$	$\frac{z_3}{F_2}$	$\frac{z_3}{F_3}$

Table 2.3: Nine possible transfer functions for tdof system.

The results below were obtained by use of a symbolic algebra program.

$$\frac{z_1}{F_1} = \left\{ \begin{array}{l} s^4\left(m_2m_3\right) + s^3\left(m_3c_1 + m_3c_2 + m_2c_2\right) \\ +s^2\left(c_1c_2 + m_2k_2 + m_3k_1 + m_3k_2\right) + s\left(c_1k_2 + c_2k_1\right) + k_1k_2 \end{array} \right\} / \text{Den} \quad (2.28)$$

$$\frac{z_1}{F_2} = \left\{ s^3\left(m_3c_1\right) + s^2\left(c_1c_2 + m_3k_1\right) + s\left(c_1k_2 + k_1c_2\right) + k_1k_2 \right\} / \text{Den} \quad (2.29)$$

$$\frac{z_1}{F_3} = \left\{ s^2\left(c_1c_2\right) + s\left(c_1k_2 + c_2k_1\right) + k_1k_2 \right\} / \text{Den} \quad (2.30)$$

$$\frac{z_2}{F_1} = \left\{ s^3\left(m_3c_1\right) + s^2\left(c_1c_2 + m_3k_1\right) + s\left(c_1k_2 + c_2k_1\right) + k_1k_2 \right\} / \text{Den} \quad (2.31)$$

$$\frac{z_2}{F_2} = \left\{ \begin{array}{l} s^4\left(m_1m_3\right) + s^3\left(m_1c_2 + m_3c_1\right) \\ +s^2\left(m_1k_2 + c_1c_2 + m_3k_1\right) \\ +s\left(c_1k_2 + c_2k_1\right) + k_1k_2 \end{array} \right\} / \text{Den} \quad (2.32)$$

$$\frac{z_2}{F_3} = \left\{ s^3\left(m_1c_2\right) + s^3\left(m_1k_2 + c_1c_2\right) + s\left(c_1k_2 + c_2k_1\right) + k_1k_2 \right\} / \text{Den} \quad (2.33)$$

$$\frac{z_3}{F_1} = \left\{ s^2 \left(c_1 c_2 \right) + s\left(c_1 k_2 + c_2 k_1 \right) + k_1 k_2 \right\} / \text{Den} \tag{2.34}$$

$$\frac{z_3}{F_2} = \left\{ s^3 \left(m_1 c_2 \right) + s^2 \left(m_1 k_2 + c_1 c_2 \right) + s\left(c_1 k_2 + c_2 k_1 \right) + k_1 k_2 \right\} / \text{Den} \tag{2.35}$$

$$\frac{z_3}{F_3} = \left\{ \begin{array}{l} s^4 \left(m_1 m_2 \right) + s^3 \left(m_1 c_2 + m_1 c_1 + m_2 c_1 \right) \\ + s^2 \left(m_2 k_1 + m_1 k_1 + m_1 k_2 + c_1 c_2 \right) \\ + s\left(c_2 k_1 + c_1 k_2 \right) + \left(k_1 k_2 \right) \end{array} \right\} / \text{Den} \tag{2.36}$$

Where Den is:

$$\text{Den} = s^2 \left\{ \begin{array}{l} s^4 \left(m_1 m_2 m_3 \right) + s^3 \left(m_2 m_3 c_1 + m_1 m_3 c_1 + m_1 m_2 c_2 + m_1 m_3 c_2 \right) \\ + s^2 \left(m_1 m_3 k_1 + m_1 m_3 k_2 + m_1 m_2 k_2 + m_2 c_1 c_2 + m_3 c_1 c_2 + m_1 c_1 c_2 \right. \\ \qquad \left. + k_1 m_2 m_3 \right) \\ + s\left(m_3 c_1 k_2 + m_2 c_2 k_1 + m_1 c_2 k_1 + m_1 c_1 k_2 + m_3 c_2 k_1 + m_2 c_1 k_2 \right) \\ + \left(m_1 k_1 k_2 + m_2 k_1 k_2 + m_3 k_1 k_2 \right) \end{array} \right\} \tag{2.37}$$

Note that all the transfer functions have the same denominator, Den, called the **characteristic equation**.

To simplify the system for hand calculations, take:

$$\begin{array}{l} m_1 = m_2 = m_3 = m \\ c_1 = c_2 = c \\ k_1 = k_2 = k \end{array} \tag{2.38}$$

$$z_{11} = \frac{z_1}{F_1} = \left(m^2 s^4 + 3mcs^3 + \left(c^2 + 3mk \right) s^2 + 2cks + k^2 \right) / \text{Den1} \tag{2.39}$$

$$z_{12} = \frac{z_1}{F_2} = \left(mcs^3 + \left(c^2 + mk \right) s^2 + 2cks + k^2 \right) / \text{Den1} \tag{2.40}$$

$$z_{13} = \frac{z_1}{F_3} = \left(c^2 s^2 + 2cks + k^2 \right) / \text{Den1} \tag{2.41}$$

$$z_{21} = \frac{z_2}{F_1} = \left(mcs^3 + \left(c^2 + mk\right)s^2 + \left(2ck\right)s + k^2\right) / \text{Den1} \tag{2.42}$$

$$z_{22} = \frac{z_2}{F_2} = \left(m^2s^4 + 2mcs^3 + \left(2mk + c^2\right)s^2 + 2cks + k^2\right) / \text{Den1} \tag{2.43}$$

$$z_{23} = \frac{z_2}{F_3} = \left(mcs^3 + \left(c^2 + mk\right)s^2 + 2cks + k^2\right) / \text{Den1} \tag{2.44}$$

$$z_{31} = \frac{z_3}{F_1} = \left(c^2s^2 + 2cks + k^2\right) / \text{Den1} \tag{2.45}$$

$$z_{32} = \frac{z_3}{F_2} = \left(mcs^3 + \left(c^2 + mk\right)s^2 + 2cks + k^2\right) / \text{Den1} \tag{2.46}$$

$$z_{33} = \frac{z_3}{F_3} = \left(m^2s^4 + 3mcs^3 + \left(c^2 + 3mk\right)s^2 + 2cks + k^2\right) / \text{Den1} \tag{2.47}$$

Where:

$$\text{Den1} = \left\{m^3s^4 + 4m^2cs^3 + \left(4m^2k + 3mc^2\right)s^2 + 6mcks + 3mk^2\right\}s^2 \tag{2.48}$$

To enable hand calculations of roots, simplify another level by making damping equal to zero:

$$\frac{z_1}{F_1} = \left(m^2s^4 + 3mks^2 + k^2\right) / \text{Den2} \tag{2.49}$$

$$\frac{z_1}{F_2} = \left(mks^2 + k^2\right) / \text{Den2} \tag{2.50}$$

$$\frac{z_1}{F_3} = k^2 / \text{Den2} \tag{2.51}$$

$$\frac{z_2}{F_1} = \left(mks^2 + k^2\right) / \text{Den2} \tag{2.52}$$

$$\frac{z_2}{F_2} = \left(m^2 s^4 + 2mks^2 + k^2\right) / \text{Den2} \tag{2.53}$$

$$\frac{z_2}{F_3} = \left(mks^2 + k^2\right) / \text{Den2} \tag{2.54}$$

$$\frac{z_3}{F_1} = k^2 / \text{Den2} \tag{2.55}$$

$$\frac{z_3}{F_2} = \left(mks^2 + k^2\right) / \text{Den2} \tag{2.56}$$

$$\frac{z_3}{F_3} = \left(m^2 s^4 + 3mks^2 + k^2\right) / \text{Den2} \tag{2.57}$$

$$\text{Den2} = s^2\left(m^3 s^4 + 4m^2 ks^2 + 3mk^2\right) \tag{2.58}$$

2.4.3 Transfer Function Matrix for Undamped Model

A more convenient method of arranging and keeping track of the various transfer functions is to use a matrix form for the transfer function, called the **transfer function matrix**:

$$\begin{bmatrix} z_{11} & z_{12} & z_{13} \\ z_{21} & z_{22} & z_{23} \\ z_{31} & z_{32} & z_{33} \end{bmatrix} \tag{2.59}$$

Where:

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} z_{11} & z_{12} & z_{13} \\ z_{21} & z_{22} & z_{23} \\ z_{31} & z_{32} & z_{33} \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} \tag{2.60}$$

The transfer function matrix can then be written for the undamped case as follows, where each term of the numerator matrix is divided by the common denominator:

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} =$$

$$\frac{\begin{bmatrix} (m^2s^4 + 3mks^2 + k^2) & (mks^2 + k^2) & k^2 \\ (mks^2 + k^2) & (m^2s^4 + 2mks^2 + k^2) & (mks^2 + k^2) \\ k^2 & (mks^2 + k^2) & (m^2s^4 + 3mks^2 + k^2) \end{bmatrix}}{s^2\left(m^3s^4 + 4m^2ks^2 + 3mk^2\right)} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} \tag{2.61}$$

2.4.4 Four Distinct Transfer Functions

We will be dealing with only Single Input Single Output (SISO) systems until Chapter 19, when a Multi Input Multi Output (MIMO) system is examined. This means that we will be applying only a single force to the system at any time, F_1, F_2 or F_3 ,and will only be taking the displacement of a single degree of freedom, z_1, z_2 or z_3 .

Because there are three inputs and three outputs, there are nine possible SISO transfer functions to investigate. However, because of the symmetry of the system ($z_{ij} = z_{ji}$) there are only four distinct transfer functions. Expanding the denominator into factors and simplifying:

$$\frac{z_1}{F_1} = \frac{m^2s^4 + 3mks^2 + k^2}{s^2\left(m^3s^4 + 4m^2ks^2 + 3mk^2\right)} \tag{2.62}$$

$$\frac{z_2}{F_1} = \frac{(mks^2 + k^2)}{s^2\left(m^3s^4 + 4m^2ks^2 + 3mk^2\right)}$$

$$= \frac{k(ms^2 + k)}{s^2(ms^2 + k)(m^2s^2 + 3km)}$$

$$= \frac{k}{s^2(m^2s^2 + 3km)} \quad \text{(note cancelling of pole/zero)} \tag{2.63}$$

$$\frac{z_3}{F_1} = \frac{k^2}{s^2\left(m^3s^4 + 4m^2ks^2 + 3mk^2\right)} \tag{2.64}$$

$$\frac{z_2}{F_2} = \frac{m^2 s^4 + 2mks^2 + k^2}{s^2\left(m^3 s^4 + 4m^2 ks^2 + 3mk^2\right)} \tag{2.65}$$

2.4.5 Poles

The **poles**, **eigenvalues**, or **resonant frequencies**, are the roots of the characteristic equation. Poles show the frequencies where the system will amplify inputs, and are a basic characteristic of the system. The poles are not a function of which transfer function is used since all the transfer functions for a given system have the same characteristic equation, as shown by the common denominator of (2.61).

The poles for a system depend only on the distribution of mass, stiffness, and damping throughout the system, not on where the forces are applied or where displacements are measured.

Setting the characteristic equation equal to zero and solving for the roots (poles):

$$s^2\left(m^3 s^4 + 4m^2 ks^2 + 3mk^2\right) = 0 \tag{2.66}$$

$$s^2 = 0 \text{ is a double root at the origin } s_{1,2} = 0 \tag{2.67}$$

Now taking the term in parentheses and setting equal to zero:

$$\left(m^3\right)s^4 + \left(4m^2 k\right)s^2 + \left(3mk^2\right) = 0 \tag{2.68}$$

Solving as a quadratic in s^2:

$$s^2 = \frac{-4m^2 k \pm \left(16m^4 k^2 - 12m^4 k^2\right)^{\frac{1}{2}}}{2m^3}$$

$$= \frac{-4m^2 k \pm \left(4m^4 k^2\right)^{\frac{1}{2}}}{2m^3}$$

$$= \frac{-4m^2 k \pm 2m^2 k}{2m^3} = \frac{-2m^2 k}{m^3}$$

$$= \frac{-2k}{2m}, \frac{-6k}{2m}$$

$$= \frac{-k}{m}, \frac{-3k}{m} \tag{2.69}$$

$$s_{3,4} = \pm j\sqrt{\frac{k}{m}} = \pm j1 \tag{2.70}$$

$$s_{5,6} = \pm j\sqrt{\frac{3k}{m}} = \pm j1.732 \tag{2.71}$$

Because there is no damping, the poles all fall on the s-plane imaginary axis.

2.4.6 Zeros

The **zeros** of each SISO transfer function are defined by the roots of its numerator. Zeros show the frequencies where the system will attenuate inputs. Unlike the poles, which are a characteristic of the system and are the same for every transfer function, zeros can be different for every transfer function and some transfer functions may have no zeros. Chapter 4 will discuss one physical interpretation of zeros, showing how to calculate the number of zeros for various transfer functions for a series-connected lumped mass system.

Calculate the z_1 / F_1 zeros:

$$m^2 s^4 + 3mks^2 + k^2 = 0 \tag{2.72}$$

$$s^2 = \frac{-3mk \pm \left(9m^2k^2 - 4m^2k^2\right)^{\frac{1}{2}}}{2m^2}$$

$$= \frac{-3mk \pm \sqrt{5}mk}{2m^2} = \frac{-3k \pm \sqrt{5}k}{2m}$$

$$= \left(\frac{k}{m}\right)\left(\frac{-3 \pm \sqrt{5}}{2}\right) = \left(\frac{k}{m}\right)(-0.3820), \ \left(\frac{k}{m}\right)(-2.618) \tag{2.73}$$

Taking the square root of the two values above gives two pair of complex conjugate roots:

$$s_{1,2} = \pm j0.618\sqrt{\frac{k}{m}} = \pm j\ 0.618 \tag{2.74}$$

$$s_{3,4} = \pm j1.618\sqrt{\frac{k}{m}} = \pm j\ 1\ .618 \tag{2.75}$$

Calculate the z_2 / F_1 zeros:

$$mks^2 + k^2 = 0 \tag{2.76}$$

$$s^2 = \frac{-k^2}{mk} = \frac{-k}{m} \tag{2.77}$$

$$s_{1,2} = \pm j\sqrt{\frac{k}{m}} = \pm j \tag{2.78}$$

Calculate the z_3 / F_1 zeros:

$$k^2 = 0 \quad \text{there are no zeros.} \tag{2.79}$$

Calculate the z_2 / F_2 zeros:

$$m^2 s^4 + 2mks^2 + k^2 = 0 \tag{2.80}$$

$$s^2 = \frac{-2mk \pm \left(4m^2k^2 - 4m^2k^2\right)}{2m^2}$$

$$= \frac{-2mk}{2m^2} = \frac{-k}{m} \pm 0 \tag{2.81}$$

$$s_{1,2} = \pm j\sqrt{\frac{k}{m}} = \pm j \tag{2.82}$$

$$s_{3,4} = \pm j \tag{2.83}$$

As with the poles, since there is no damping in the system, all the zeros are also on the imaginary axis.

2.4.7 Summarizing Poles and Zeros, Matrix Format

$$\frac{\begin{bmatrix} (\pm 0.62, \pm 1.62) & \pm j & \text{none} \\ \pm j & (\pm j, \pm j) & \pm j \\ \text{none} & \pm j & (\pm 0.62, \pm 1.62) \end{bmatrix}}{(\pm 0\,j)(\pm 1, \pm 1.732)j} \tag{2.84}$$

The 3x3 matrix of zero values for the 3x3 transfer function matrix is in the numerator of (2.82) and the pole values are in the denominator.

2.5 MATLAB Code tdofpz3x3.m – Plot Poles and Zeros

2.5.1 Code Description

The program listing below uses the "num/den" form of the transfer function and calculates and plots all nine pole/zero combinations for the nine different transfer functions. It prompts for values of the two dampers, c1 and c2, where the default values (hitting the "enter" key) 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.

The values of the poles and zeros as well as the "zpk" forms of the transfer functions are listed in the MATLAB command window.

Note that in most MATLAB code, the critical definitions and calculations take only a few commands while plotting and annotating the plots take the bulk of the space.

2.5.2 Code Listing

```
%       tdofpz3x3.m          plotting poles/zeros of tdof model, all 9 plots

        clf;

        clear all;

%       using MATLAB's pzmap function with the "tf" form using num/den
%       to define the numerator and denominator terms of the different
%       transfer functionx

%       assign values for masses, damping, and stiffnesses

        m1 = 1;
        m2 = 1;
```

```
m3 = 1;
k1 = 1;
k2 = 1;

%       prompt for c1 and c2 values, set to zero to match closed form solution

c1 = input('enter value for damper c1, default is zero, ...  ');

if  isempty(c1)
        c1 = 0;
end

c2 = input('enter value for damper c2, default is zero, ...  ');

if  isempty(c2)
        c2 = 0;
end

%       define row vectors of numerator and denominator coefficients

den = [(m1*m2*m3) (m2*m3*c1 + m1*m3*c1 + m1*m2*c2 + m1*m3*c2) ...
      (m1*m3*k1 + m1*m3*k2 + m1*m2*k2 + m2*c1*c2 + m3*c1*c2 + ...
       m1*c1*c2 + k1*m2*m3) ...
            (m3*c1*k2 + m2*c2*k1 + m1*c2*k1 + m1*c1*k2 + …
             m3*c2*k1 + m2*c1*k2) ...
            (m1*k1*k2 + m2*k1*k2 + m3*k1*k2) 0 0];

z11num = [(m2*m3) (m3*c1 + m3*c2 + m2*c2) (c1*c2 + m2*k2 +…
            m3*k1 + m3*k2)   (c1*k2 + c2*k1) (k1*k2)];

z21num = [(m3*c1) (c1*c2 + m3*k1) (c1*k2 + c2*k1) (k1*k2)];

z31num = [(c1*c2) (c1*k2 + c2*k1) (k1*k2)];

z22num = [(m1*m3) (m1*c2 + m3*c1) (m1*k2 + c1*c2 + m3*k1) ...
                        (c1*k2 + c2*k1) (k1*k2)];

%       use the "tf" function to convert to define "transfer function" systems

sysz11 = tf(z11num,den)

sysz21 = tf(z21num,den)

sysz31 = tf(z31num,den)

sysz22 = tf(z22num,den)

%       use the "zpk" function to convert from transfer function to zero/pole/gain form

zpkz11 = zpk(sysz11)

zpkz21 = zpk(sysz21)

zpkz31 = zpk(sysz31)
```

```
        zpkz22 = zpk(sysz22)

%       use the "pzmap" function to map the poles and zeros of each transfer function

        [p11,z11] = pzmap(sysz11);

        [p21,z21] = pzmap(sysz21);

        [p31,z31] = pzmap(sysz31);

        [p22,z22] = pzmap(sysz22);

        p11

        z11

        z21

        z31

        z22

%       plot z11 for later use

        subplot(1,1,1)
        plot(real(p11),imag(p11),'k*')
        hold on
        plot(real(z11),imag(z11),'ko')
        title('Poles and Zeros of z11')
        ylabel('Imag')
        axis([-2 2 -2 2])
        axis('square')
        grid
        hold off

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

%       plot all 9 plots on a 3x3 grid

        subplot(3,3,1)
        plot(real(p11),imag(p11),'k*')
        hold on
        plot(real(z11),imag(z11),'ko')
        title('Poles and Zeros of z11')
        ylabel('Imag')
        axis([-2 2 -2 2])
        axis('square')
        grid
        hold off

        subplot(3,3,2)
        plot(real(p21),imag(p21),'k*')
        hold on
        plot(real(z21),imag(z21),'ko')
        title('Poles and Zeros of z12')
```

```
ylabel('Imag')
axis([-2 2 -2 2])
axis('square')
grid
hold off

subplot(3,3,3)
plot(real(p31),imag(p31),'k*')
hold on
plot(real(z31),imag(z31),'ko')
title('Poles and Zeros of z13')
ylabel('Imag')
axis([-2 2 -2 2])
axis('square')
grid
hold off

subplot(3,3,4)
plot(real(p21),imag(p21),'k*')
hold on
plot(real(z21),imag(z21),'ko')
title('Poles and Zeros of z21')
ylabel('Imag')
axis([-2 2 -2 2])
axis('square')
grid
hold off

subplot(3,3,5)
plot(real(p22),imag(p22),'k*')
hold on
plot(real(z22),imag(z22),'ko')
title('Poles and Zeros of z22')
ylabel('Imag')
axis([-2 2 -2 2])
axis('square')
grid
hold off

subplot(3,3,6)
plot(real(p21),imag(p21),'k*')
hold on
plot(real(z21),imag(z21),'ko')
title('Poles and Zeros of z23')
ylabel('Imag')
axis([-2 2 -2 2])
axis('square')
grid
hold off

subplot(3,3,7)
plot(real(p31),imag(p31),'k*')
hold on
plot(real(z31),imag(z31),'ko')
title('Poles and Zeros of z31')
```

```
xlabel('Real')
ylabel('Imag')
axis([-2 2 -2 2])
axis('square')
grid
hold off

subplot(3,3,8)
plot(real(p21),imag(p21),'k*')
hold on
plot(real(z21),imag(z21),'ko')
title('Poles and Zeros of z32')
xlabel('Real')
ylabel('Imag')
axis([-2 2 -2 2])
axis('square')
grid
hold off

subplot(3,3,9)
plot(real(p11),imag(p11),'k*')
hold on
plot(real(z11),imag(z11),'ko')
title('Poles and Zeros of z33')
xlabel('Real')
ylabel('Imag')
axis([-2 2 -2 2])
axis('square')
grid
hold off

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

%       check for real axis values to set plot scale

z11_realmax = max(abs(real(z11)));
z21_realmax = max(abs(real(z21)));
z31_realmax = max(abs(real(z31)));
z22_realmax = max(abs(real(z22)));

maxplot = max([z11_realmax z21_realmax z31_realmax z22_realmax]);

if  maxplot > 2

        maxplot = ceil(maxplot);

else

        maxplot = 2.0;

end

z11_realmax = max(abs(real(z11)));
subplot(1,1,1)
```

```
plot(real(p11),imag(p11),'k*')
hold on
plot(real(z11),imag(z11),'ko')
title('Poles and Zeros of z11, z33')
ylabel('Imag')
axis([-maxplot maxplot -maxplot maxplot])
axis('square')
grid
hold off

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

plot(real(p21),imag(p21),'k*')
hold on
plot(real(z21),imag(z21),'ko')
title('Poles and Zeros of z21, z12, z23, z32')
ylabel('Imag')
axis([-maxplot maxplot -maxplot maxplot])
axis('square')
grid
hold off

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

plot(real(p31),imag(p31),'k*')
hold on
plot(real(z31),imag(z31),'ko')
title('Poles and Zeros of z31, z13')
xlabel('Real')
ylabel('Imag')
axis([-maxplot maxplot -maxplot maxplot])
axis('square')
grid
hold off

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

plot(real(p22),imag(p22),'k*')
hold on
plot(real(z22),imag(z22),'ko')
title('Poles and Zeros of z22')
ylabel('Imag')
axis([-maxplot maxplot -maxplot maxplot])
axis('square')
grid
hold off
```

2.5.3 Code Output – Pole/Zero Plots in Complex Plane

2.5.3.1 Undamped Model – Pole/Zero Plots

The pole/zero plot and pole/zero calculated values for c1 = c2 = 0 are shown below. Poles are plotted as asterisks and zeros as circles.

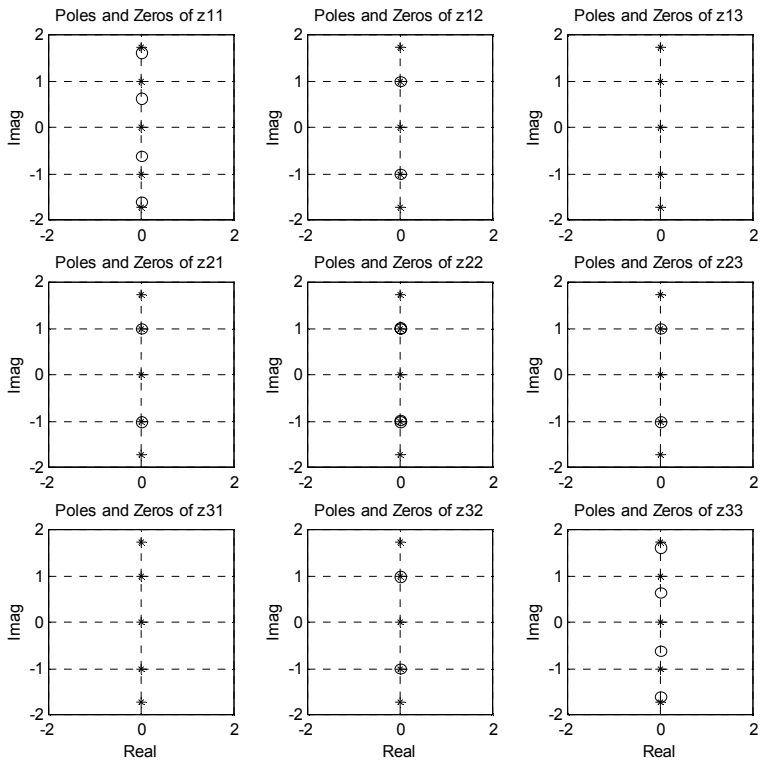


Figure 2.8: Pole/zero plots for nine transfer functions. Poles are indicated by asterisks and zeros by circles.

The first thing to notice about the pole/zero plots is that they all have the same poles. The rigid body mode (resonant frequency = 0 hz) is evident by the pair of zeros at the origin, $\pm 0\mathrm{j}$. The zeros of each particular transfer function are seen to be dependent upon which transfer function is taken. Note that with zero damping, all the poles and zeros are on the imaginary axis, indicating that the real portions of their complex values are zero and that there is no damping.

In the next chapter we will discuss frequency responses of transfer functions and will link the pole/zero locations in the complex plane to amplification/attenuation regions of the frequency response plots.

The poles and zeros from the MATLAB output are listed below:

```
poles =

      0
      0
      0 + 1.7321i
      0 - 1.7321i
      0 + 1.0000i
      0 - 1.0000i

zeros_z11 =

      0 + 1.6180i
      0 - 1.6180i
      0 + 0.6180i
      0 - 0.6180i

zeros_z21 =

      0 + 1.0000i
      0 - 1.0000i

zeros_z31 =

   Empty matrix: 0-by-1

zeros_z22 =

  -0.0000 + 1.0000i
  -0.0000 - 1.0000i
   0.0000 + 1.0000i
   0.0000 - 1.0000i
```

Table 2.3: Poles and zeros of tdof transfer functions, undamped.

Repeating the matrix listing of pole/zero locations from previous analysis:

$$\frac{\begin{bmatrix} (\pm 0.62, \pm 1.62) & \pm j & \text{none} \\ \pm j & (\pm j, \pm j) & \pm j \\ \text{none} & \pm j & (\pm 0.62, \pm 1.62) \end{bmatrix}}{(\pm 0j)(\pm 1, \pm 1.732)j} \tag{2.85}$$

Note that MATLAB calculates an "Empty matrix 0 by 1" for the zeros of z31, which matches our calculations which show "none." Also note that several of the plots, z12, z21, z22, z23 and z32, have zeros and poles overlaying each other, where the pole cancels the effect of the zero. We will discuss this cancellation further in the next chapter.

2.5.3.2 Damped Model – Pole/Zero Plots

If damping is not set to zero for c1 and/or c2, the poles (with the exception of the two poles at the origin) and zeros will move from the imaginary axis to the left hand side of the complex plane, with the real parts of the poles and zeros having negative values. The pole/zero plot and MATLAB output listing below are for values of c1 = c2 = 0.1, arbitrarily chosen to illustrate the "damped" case.

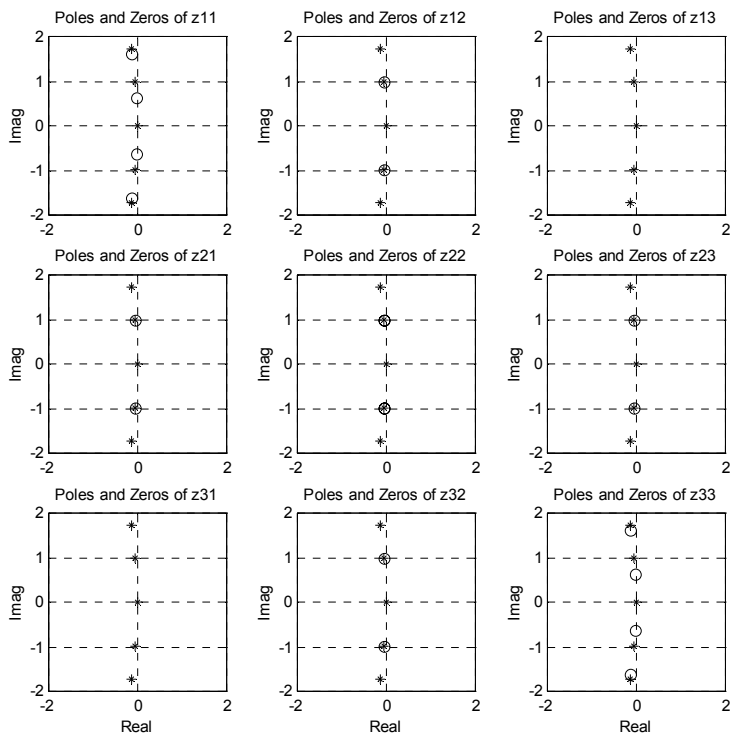


Figure 2.9: Pole/zero plots for nine transfer functions for c1 = c2 = 0.1. Poles are indicated by asterisks and zeros by circles. Negative real axis zeros not shown because of plot scaling.

The limited scale for the nine plots above do not show the real axis zeros, see the figures below for the entire plot. The only poles/zeros that are on the imaginary axis are the two poles at zero, the rigid body mode – which will be described in detail in Chapter 3.

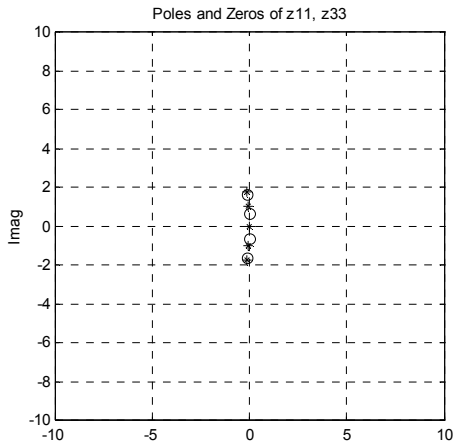


Figure 2.10: Expanded scale pole/zero plots for z11, z33 transfer functions – no real axis zeros.

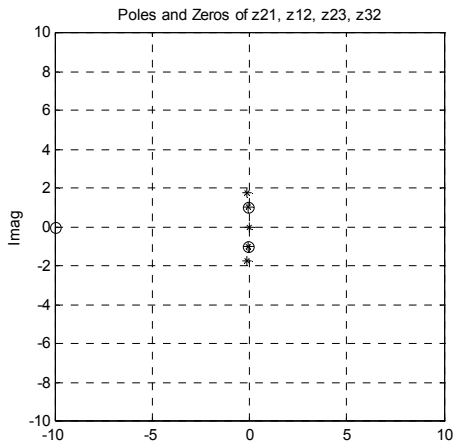


Figure 2.11: Expanded scale pole/zero plots for z21, z12, z23 and z32 transfer functions – one real axis zero at -10.

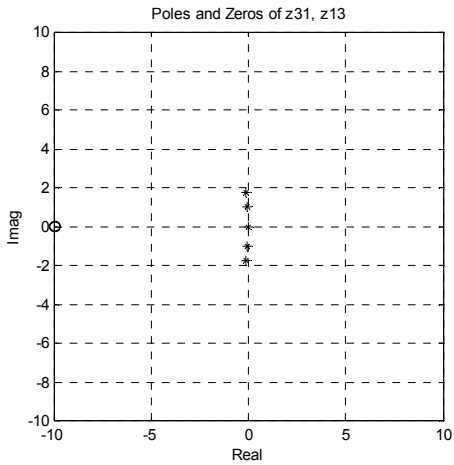


Figure 2.12: Expanded scale pole/zero plots for z31 and z13 transfer functions – two real axis zeros at -10.

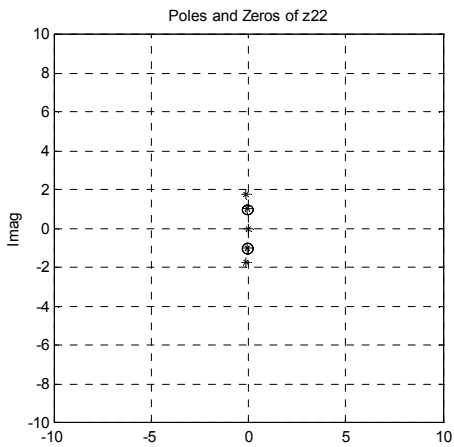


Figure 2.13: Expanded scale pole/zero plots for z31 and z13 transfer functions – no real axis zeros.

The MATLAB calculated values for the poles and zeros for the damped case are below:

```
p11 =

      0
      0
  -0.1500 + 1.7255i
  -0.1500 - 1.7255i
  -0.0500 + 0.9987i
  -0.0500 - 0.9987i

z11 =

  -0.1309 + 1.6127i
  -0.1309 - 1.6127i
  -0.0191 + 0.6177i
  -0.0191 - 0.6177i

z21 =

 -10.0000
  -0.0500 + 0.9987i
  -0.0500 - 0.9987i

z31 =

 -10.0000 + 0.0000i
 -10.0000 - 0.0000i

z22 =

  -0.0500 + 0.9987i
  -0.0500 - 0.9987i
  -0.0500 + 0.9987i
-0.0500 - 0.9987i
```

Table 2.4: Poles and zeros of tdof transfer functions, damped.

Several observations can be made about the poles and zeros above. First, all of the poles with the exception of the two rigid body poles p11 = 0 are to the left of the imaginary axis, indicating that the system now has damping. Note that there are several new zeros. The z21 transfer function now has a real zero at –10.0 in addition to the two complex zeros. The z31 transfer function has two zeros now at –10, whereas for the no damping case it had no zeros. These extra zeros do not show up on Figure 2.9 because of plot axis scaling but with the real axis expanded in Figures 2.10 to 2.13 they appear. The reason for these "additional" zeros can be seen if we look at the z21 and z31 transfer functions, repeated from (2.31) and (2.34):

$$\frac{z_2}{F_1} = \left\{ s^3 \left(m_3 c_1 \right) + s^2 \left(c_1 c_2 + m_3 k_1 \right) + s\left(c_1 k_2 + c_2 k_1 \right) + k_1 k_2 \right\} / \text{Den} \qquad (2.86)$$

$$\frac{z_3}{F_1} = \left\{ s^2 \left(c_1 c_2 \right) + s\left(c_1 k_2 + c_2 k_1 \right) + k_1 k_2 \right\} / \text{Den} \qquad (2.87)$$

With values for c1 and c2 not equal to zero, the z21 transfer function is third degree, meaning that it should have three roots. With damping equal to zero, only two complex zeros are calculated by MATLAB and by hand. The third root is located at $-\infty$. As damping values for c1 and c2 are increased the root at $-\infty$ moves to the right, towards the origin.

The z31 transfer function has no zeros with zero damping, but is second degree and with infinitely small damping values has two roots at $-\infty$. As the values of c1 and c2 increase, the two zeros at $-\infty$ start moving toward the origin.

2.5.3.3 Root Locus, tdofpz3x3_rlocus.m

In the last two sections we have discussed pole/zero plots for undamped and damped models. For the damped model we chose values of 0.1 for c1 and c2. It would be nice to have a systematic method to display poles and zeros for a range of damping values. There is a MATLAB Control Toolbox function "rlocus" which plots the root locus for an open-loop SISO system. We could use this function if the damping values could be broken out of the system and be treated as a feedback gain. Unfortunately for our tdof system this is not possible, but we can still plot a locus by using a for-loop.

The code listed below, **tdofpz3x3_rlocus.m**, is taken from the initial section of **tdofpz3x3.m**. A for-loop cycles through a vector of damping values, calculating and plotting the poles and zeroes for each damping value.

```
        echo off
%       tdofpz3x3_rlocus.m      plotting locus of poles/zeros of z11 for tdof
%        model for range of damping values.

        clf;

        clear all;

%       assign values for masses, damping, and stiffnesses

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

```
k1 = 1;
k2 = 1;

%       define vector of damping values for c1 and c2

cvec = [0 .2 .4 .6 .8 1.0 1.1 1.05 1.1 1.15 1.16];

for cnt = 1:length(cvec)

c1 = cvec(cnt);

c2 = cvec(cnt);

%       define row vectors of numerator and denominator coefficients

den = [(m1*m2*m3) (m2*m3*c1 + m1*m3*c1 + m1*m2*c2 + m1*m3*c2) ...
      (m1*m3*k1 + m1*m3*k2 + m1*m2*k2 + m2*c1*c2 + m3*c1*c2 + ...
       m1*c1*c2 + k1*m2*m3) ...
                (m3*c1*k2 + m2*c2*k1 + m1*c2*k1 + m1*c1*k2 + …
                 m3*c2*k1 + m2*c1*k2) ...
                (m1*k1*k2 + m2*k1*k2 + m3*k1*k2) 0 0];

z11num = [(m2*m3) (m3*c1 + m3*c2 + m2*c2) …
         (c1*c2 + m2*k2 + m3*k1 + m3*k2) .(c1*k2 + c2*k1) (k1*k2)];

z21num = [(m3*c1) (c1*c2 + m3*k1) (c1*k2 + c2*k1) (k1*k2)];

z31num = [(c1*c2) (c1*k2 + c2*k1) (k1*k2)];

z22num = [(m1*m3) (m1*c2 + m3*c1) (m1*k2 + c1*c2 + m3*k1) ...
                                (c1*k2 + c2*k1) (k1*k2)];

%       use the "tf" function to convert to define "transfer function" systems

sysz11 = tf(z11num,den);

sysz21 = tf(z21num,den);

sysz31 = tf(z31num,den);

sysz22 = tf(z22num,den);

%       use the "pzmap" function to map the poles and zeros of each transfer function

[p11,z11] = pzmap(sysz11);

[p21,z21] = pzmap(sysz21);

[p31,z31] = pzmap(sysz31);

[p22,z22] = pzmap(sysz22);

%       plot poles and zeros of z11

subplot(1,1,1)
```

```
plot(real(p11),imag(p11),'k*')
hold on
plot(real(z11),imag(z11),'ko')
title('Poles and Zeros of z11 for range of damping values c1 and c2')
xlabel('Real')
ylabel('Imag')
axis([-3 1 -2 2])
axis('square')
grid on

end

hold off
```

The root locus plot below is for the following values of damping:

cvec = [0 .2 .4 .6 .8 1.0 1.1 1.05 1.1 1.15 1.16];

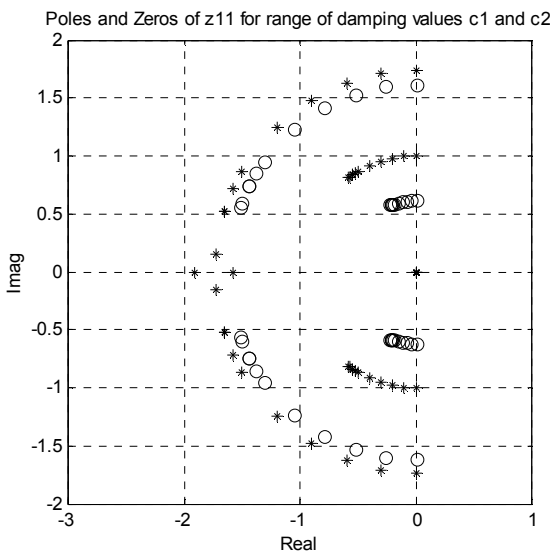


Figure 2.14: Pole zero plot for z11 transfer function.

The plot starts out with damping values of zero for c1 and c2. The poles and zeros for zero damping are located on the imaginary axis. The poles are located at 0, 0, $\pm 1j$, $\pm 1.732j$. The zeros are located at $\pm 0.62j$ and $\pm 1.62j$. As damping is increased from zero, the poles and zeros (except the two poles at the origin) start moving to the left, away from the imaginary axis. The poles and zeros move at different rates as damping is increased. The poles at $\pm 1j$

and zeros at $\pm 0.62j$ move to the left less than the poles at $\pm 1.732j$ and the zeros at $\pm 1.62j$. In fact, the two poles at $\pm 1.732j$ move so much that at damping values of 1.16 the poles intercept the real axis and split. One moves to the left and the other to the right along the real axis.

Plotting pole and zero locations as a function of system parameters was introduced in 1949 (Evans 1949), as the Evans root locus technique. The hand plotting originally used has been largely replaced with computer plotting techniques as shown above or by using the "rlocus" function. However, because the ability to hand sketch root loci is such a powerful tool, it is still taught in beginning control theory courses (Franklin 1994).

2.5.3.4 Undamped and Damped Model – tf and zpk Forms

This section is included to start familiarizing the reader with the various forms of transfer functions available with MATLAB and to prepare for issues in the next chapter.

Table 2.6 shows the transfer function form of the four distinct transfer functions for the tdof model for the undamped (c1 = c2 = 0) and damped (c1 = c2 = 0.1) cases run earlier. The numerator and denominator are both arranged in polynomial form. Table 2.7 shows the zpk form, where the numerator and denominator are both arranged as products of the zeros and poles with a gain term multiplying the numerator.

Note that the denominators of all the undamped transfer functions are the same, as are the denominators of all the damped transfer functions. However, the numerators are all different because of the different number of poles and zeros for each transfer function. For instance the z31 undamped transfer function has no zeros, only a gain term of 1.0, while the z11 undamped transfer function has two sets of complex zeros.

In going from the undamped to damped case, we showed that extra zeros appeared in the z21 and z31 transfer functions. It is easier to see where the extra zeros originate using the zpk form than using the tf form. Comparing the undamped and damped numerators of the z31 zpk transfer function form shows the extra $(s+10)^2$ term, from which the two real axis zeros arise. We will use the zpk form of the transfer functions in the next chapter to calculate frequency response at a specific frequency.

z11 Undamped Transfer function:	z11 Damped Transfer function:
$\frac{s^4 + 3 s^2 + 1}{s^6 + 4 s^4 + 3 s^2}$	$\frac{s^4 + 0.3 s^3 + 3.01 s^2 + 0.2 s + 1}{s^6 + 0.4 s^5 + 4.03 s^4 + 0.6 s^3 + 3 s^2}$
z21 Undamped Transfer function:	z21 Damped Transfer function:
$\frac{s^2 + 1}{s^6 + 4 s^4 + 3 s^2}$	$\frac{0.1 s^3 + 1.01 s^2 + 0.2 s + 1}{s^6 + 0.4 s^5 + 4.03 s^4 + 0.6 s^3 + 3 s^2}$
z31 Undamped Transfer function:	z31 Damped Transfer function:
$\frac{1}{s^6 + 4 s^4 + 3 s^2}$	$\frac{0.01 s^2 + 0.2 s + 1}{s^6 + 0.4 s^5 + 4.03 s^4 + 0.6 s^3 + 3 s^2}$
z22 Undamped Transfer function:	z22 Damped Transfer function:
$\frac{s^4 + 2 s^2 + 1}{s^6 + 4 s^4 + 3 s^2}$	$\frac{s^4 + 0.2 s^3 + 2.01 s^2 + 0.2 s + 1}{s^6 + 0.4 s^5 + 4.03 s^4 + 0.6 s^3 + 3 s^2}$

Table 2.5: Transfer function (tf) form of undamped and damped tdof transfer functions.

z11 Undamped Zero/pole/gain:	z11 Damped Zero/pole/gain:
$\frac{(s^2 + 0.382)(s^2 + 2.618)}{s^2 (s^2 + 1)(s^2 + 3)}$	$\frac{(s^2 + 0.0382s + 0.382)(s^2 + 0.2618s + 2.618)}{s^2 (s^2 + 0.1s + 1)(s^2 + 0.3s + 3)}$
z21 Undamped Zero/pole/gain:	z21 Damped Zero/pole/gain:
$\frac{(s^2 + 1)}{s^2 (s^2 + 1)(s^2 + 3)}$	$\frac{0.1 (s+10)(s^2 + 0.1s + 1)}{s^2 (s^2 + 0.1s + 1)(s^2 + 0.3s + 3)}$
z31 Undamped Zero/pole/gain:	z31 Damped Zero/pole/gain:
$\frac{1}{s^2 (s^2 + 1)(s^2 + 3)}$	$\frac{0.01 (s+10)^2}{s^2 (s^2 + 0.1s + 1)(s^2 + 0.3s + 3)}$
z22 Undamped Zero/pole/gain:	z22 Damped Zero/pole/gain:
$\frac{(s^2 + 1)^2}{s^2 (s^2 + 1)(s^2 + 3)}$	$\frac{(s^2 + 0.1s + 1)^2}{s^2 (s^2 + 0.1s + 1)(s^2 + 0.3s + 3)}$

Table 2.6: Zero/Pole/Gain (zpk) for undamped and damped tdof transfer functions.

Problems

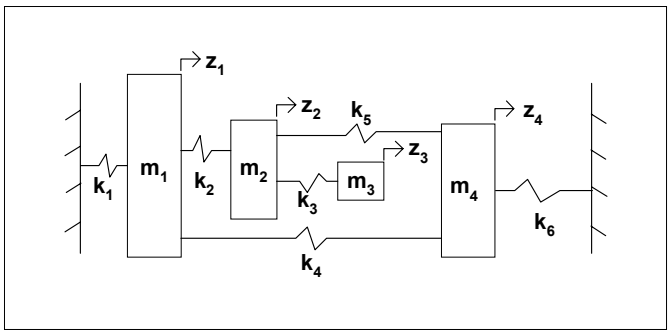


Figure P2.1: four dof system.

P2.1 Derive the global stiffness and mass matrices for the four dof system in Figure P2.1.

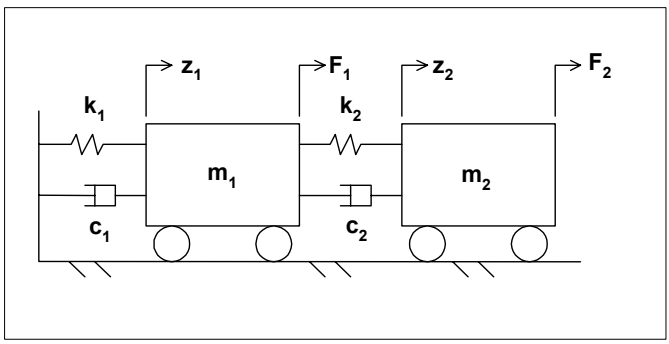


Figure P2.2: two dof problem.

P2.2 Derive the equations of motion in matrix form for the two dof model in Figure P2.2. Check for signs of diagonal terms and symmetry of off-diagonal terms.

P2.3 Solve for the four transfer functions for the two dof problem and define the 2x2 transfer function matrix. Are the denominators of all four transfer functions the same? How many unique transfer functions are there for this problem?

P2.4 Set $m_1 = m_2 = m = 1$, $k_1 = k_2 = k = 1$ and $c_1 = c_2 = 0$ and solve for the eigenvalues for the system. Solve for the zeros of the system and use the form

shown in (2.84) to summarize the poles and zeros. Hand sketch the poles and zeros in the s-plane.

P2.5 (MATLAB) Set $m_1 = m_2 = m = 1$, $k_1 = k_2 = k = 1$. Modify the **tdofpz3x3.m** file to plot the poles and zeros of the undamped two dof system. Identify the poles and zeros in the MATLAB output listing and compare with the hand-calculated values.

P2.6 (MATLAB) Set $m_1 = m_2 = m = 1$, $k_1 = k_2 = k = 1$, add damping values of $c_1 = c_2 = 0.1$ and plot the poles and zeros in the s-plane. List the poles and zeros from MATLAB and correlate the listed values with the plots. Are there any real axis zeros? How do the real axis zero(s) change with different values of c_1 and c_2, where $c_1 = c_2$.