

## CHAPTER 3

### FREQUENCY RESPONSE ANALYSIS

#### 3.1 Introduction

In Chapter 2 we calculated the transfer functions and identified the poles and zeros for the undamped system, which are repeated as (3.1) and (3.2) below, respectively. The next step in understanding the system is to plot the frequency domain behavior of each transfer function. Frequency domain behavior means identifying the magnitude and phase characteristics of each transfer function, showing how they change as the frequency of the forcing function is varied over a frequency range. Each transfer function is evaluated in the frequency domain by evaluating it at  $s = j\omega$ , where  $\omega$  is the frequency of the forcing function, radians/sec.

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \frac{\begin{bmatrix} (m^2s^4 + 3mks^2 + k^2) & (mks^2 + k^2) & k^2 \\ (mks^2 + k^2) & (m^2s^4 + 2mks^2 + k^2) & (mks^2 + k^2) \\ k^2 & (mks^2 + k^2) & (m^2s^4 + 3mks^2 + k^2) \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix}}{s^2(m^3s^4 + 4m^2ks^2 + 3mk^2)} \quad (3.1)$$

$$\frac{\begin{bmatrix} (\pm 0.62, \pm 1.62) & \pm j & \text{none} \\ \pm j & (\pm j, \pm j) & \pm j \\ \text{none} & \pm j & (\pm 0.62, \pm 1.62) \end{bmatrix}}{(\pm 0j)(\pm 1, \pm 1.732)j} \quad (3.2)$$

Instead of going directly into MATLAB to calculate and plot the frequency responses, we will first sketch them by hand, using information about the low and high frequency asymptotes and the locations of the poles and zeros. We will discuss how to find the gain and phase of a transfer function at a given frequency graphically using the locations of the poles and zeros in the complex plane and then use MATLAB to plot. Finally, mode shapes are defined, then calculated using transfer function information and plotted.

### 3.2 Low and High Frequency Asymptotic Behavior

It is always good to check either a system's rigid body or spring-like low frequency nature by hand. For this t dof system at very low frequencies there are no spring connections to ground so the system moves as a rigid body, no matter where the force is applied, to  $F_1$ ,  $F_2$ , or  $F_3$ .

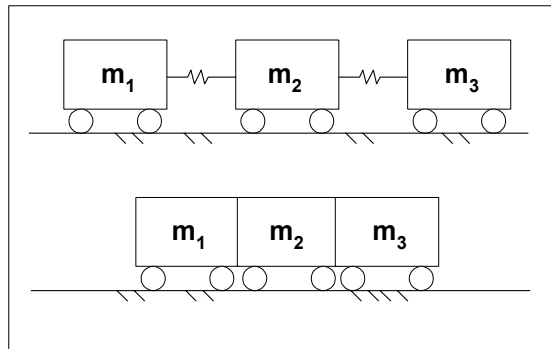


Figure 3.1: Rigid body mode of vibration.

The rigid body equation of motion (where  $z$  is the motion of all three masses together) is:

$$(3m)\ddot{z} = F \quad (3.3a,b)$$
$$\frac{z}{F} = \frac{1}{3ms^2}$$

Now we can solve for the frequency domain behavior of the system by substituting  $j\omega$  for  $s$ .

At a radian frequency of 0.1 rad/sec, a frequency taken to be an order of magnitude less than the lowest resonant frequency of 1 rad/sec, the transfer function is:

$$\frac{z}{F} = \frac{1}{3m(j\omega)^2} = \frac{1}{3m[j(0.1)]^2}$$
$$= \frac{-1}{3m(.01)} = \frac{-100}{3m} = \frac{-33.3}{m} = -33.3 \quad (3.4)$$

Converting from vector (real/imaginary) form to magnitude/phase (polar) form and using the definition of db as follows:

$$db = 20 * \log_{10}(z/F) \quad (3.5)$$

$$\left| \frac{z}{F} \right| = 33.3, \text{ or } 30.45 \text{ db} \quad (3.6a,b)$$

$$\angle \frac{z}{F} = -180^\circ$$

These results show that at a frequency of 0.1 rad/sec, the magnitude of the motion of the masses is 33.3\*F and the motion is  $-180^\circ$  out of phase with the force input.

We will now look at each individual transfer function, checking asymptotic behavior at both low and high frequencies. To do this, the four transfer functions are divided by the mass terms to give coefficients that are proportional to  $\omega_n^2 = k/m$ :

Starting with the  $z_1/F_1$  transfer function:

$$\frac{z_1}{F_1} = \frac{m^2s^4 + 3mks^2 + k^2}{s^2(m^3s^4 + 4m^2ks^2 + 3mk^2)} \quad (3.7)$$

Dividing numerator and denominator by  $m^3$  allows redefining the equation in terms of  $\omega_n$ :

$$\frac{z_1}{F_1} \Bigg|_{\div m^3} = \frac{\frac{1}{m} \left( s^4 + \frac{3ks^2}{m} + \frac{k^2}{m^2} \right)}{s^2 \left( s^4 + \frac{4ks^2}{m} + \frac{3k^2}{m^2} \right)} = \frac{s^4 + 3\omega_n^2s^2 + \omega_n^4}{ms^2 (s^4 + 4\omega_n^2s^2 + 3\omega_n^4)} \quad (3.8)$$

Substituting  $s = j\omega$  and looking at low and high frequency behaviors:

$$\left. \frac{z_1}{F_1} \right| = \frac{(\omega^4 + 3\omega_n^2(-\omega^2) + \omega_n^4)}{m(-\omega^2)(\omega^4 - 4\omega_n^2\omega^2 + 3\omega_n^4)} = \frac{\omega_n^4}{-m\omega^2(3\omega_n^4)} = \frac{-1}{3m\omega^2} \quad (3.9)$$

$\omega \ll \omega_n$

At low frequencies, the rigid body motion of  $z_1$  is falling off at a  $(-1/\omega^2)$  rate, and with a gain of  $(1/3m)$ . A rate of  $(-1/\omega^2)$  means that every decade of frequency shift, the amplitude drops by a factor of 100. Since a factor of 100 is  $-40\text{db}$ , we should see the low frequency amplitude change  $40\text{db/decade}$ .

$$\left. \frac{z_1}{F_1} \right| = \frac{-\omega^4}{m\omega^2(\omega^4)} = \frac{-1}{m\omega^2}$$

$\omega \gg \omega_n$

At high frequencies, the rigid body motion of  $z_1$  is again falling at a  $(-1/\omega^2)$  rate, but the gain is only  $(1/m)$  instead of  $(1/3m)$ . This is because at high frequencies  $z_1$  moves more as a result of  $F_1$ ; the other two masses do not want to move, as will be seen from the high frequency asymptotes of the  $z_2/F_1$  and  $z_3/F_1$  transfer functions.

Checking  $\frac{z_2}{F_1}$  :

$$\left. \frac{z_2}{F_1} \right| = \frac{\left( \frac{mks^2}{m^3} + \frac{k^2}{m^3} \right)}{s^2 \left( s^4 + \frac{4ks^2}{m} + \frac{3k^2}{m^2} \right)} = \frac{(\omega_n^2 s^2 + \omega_n^4)}{s^2 m (s^4 + 4\omega_n^2 s^2 + 3\omega_n^4)} \quad (3.11)$$

$\div m^3$

$$\left. \frac{z_2}{F_1} \right| = \frac{-\omega_n^2 \omega^2 + \omega_n^4}{-m\omega^2(\omega^4 - 4\omega_n^2\omega^2 + 3\omega_n^4)}$$

$\omega \ll \omega_n$

$$= \frac{\omega_n^4}{-m\omega^2(3\omega_n^4)} = \frac{-1}{3m\omega^2} \quad (3.12)$$

$$\left. \frac{z_2}{F_1} \right| = \frac{-\omega^2 \omega_n^2}{-m\omega^2(\omega_n^4 - 4\omega_n^2\omega^2 + 3\omega_n^4)} = \frac{-\omega^2}{-m\omega^2} \left( \frac{\omega_n^2}{\omega^4} \right) = \frac{\omega_n^2}{m\omega^4} = \frac{k}{\omega^4 m^2} \quad (3.13)$$

$$\omega \gg \omega_n$$

At low frequencies,  $z_2$  looks exactly like  $z_1$ . But at high frequencies,  $z_2$  is dropping off at a  $(1/\omega^4)$  rate, or 80db/decade, with a gain of  $(k/m^2)$ .

Checking  $\frac{z_3}{F_1}$  now:

$$\left. \frac{z_3}{F_1} \right| = \frac{\frac{k^2}{m^3}}{s^2 \left( s^4 + \frac{4ks^2}{m} + \frac{3k^2}{m^2} \right)} = \frac{\omega_n^4}{ms^2(s^4 + 4\omega_n^2 s^2 + 3\omega_n^4)} \quad (3.14)$$

$$\div m^3$$

$$\left. \frac{z_3}{F_1} \right| = \frac{\omega_n^4}{-m\omega^2(\omega^4 - 4\omega_n^2\omega^2 + 3\omega_n^4)} = \frac{\omega_n^4}{-m\omega^2(3\omega_n^4)} = \frac{-1}{3m\omega^2} \quad (3.15)$$

$$\omega \ll \omega_n$$

$$\left. \frac{z_3}{F_1} \right| = \frac{\omega_n^4}{-m\omega^2(\omega^4)} = \frac{\omega_n^4}{-m\omega^6} = \left( \frac{-k^2}{m^3} \right) \frac{1}{\omega^6} \quad (3.16)$$

$$\omega \gg \omega_n$$

At low frequencies,  $z_3$  looks exactly like  $z_1$  and  $z_2$ , but at high frequencies  $z_3$  is dropping at a  $(1/\omega^6)$  rate, or 120db/decade, with a gain of  $(-k^2/m^3)$ .

Checking  $\frac{z_2}{F_2}$  :

$$\left. \frac{z_2}{F_2} \right| = \frac{\left( \frac{m^2 s^4}{m^3} + \frac{2mks^2}{m^3} + \frac{k^2}{m^3} \right)}{s^2 \left( s^4 + \frac{4m^2 ks^2}{m^3} + \frac{3mk^2}{m^3} \right)} = \frac{(s^4 + 2\omega_n^2 s^2 + \omega_n^4)}{ms^2 (s^4 + 4\omega_n^2 s^2 + 3\omega_n^4)} \quad (3.17)$$

$\div m^3$

$$\left. \frac{z_2}{F_2} \right| = \frac{(\omega^4 + 2\omega_n^2(-\omega^2) + \omega_n^4)}{-m\omega^2(\omega^4 - 4\omega_n^2\omega^2 + 3\omega_n^4)} = \frac{\omega_n^4}{-m\omega^2(3\omega_n^4)} = \frac{-1}{3m\omega^2} \quad (3.18)$$

$\omega \ll \omega_n$

$$\left. \frac{z_2}{F_2} \right| = \frac{\omega^4}{-m\omega^2(\omega^4)} = \frac{-1}{m\omega^2} \quad (3.19)$$

$\omega \gg \omega_n$

At low frequencies,  $z_2/F_2$  looks exactly like  $z_1/F_1$ ,  $z_2/F_1$ , and  $z_3/F_1$ . But at high frequencies  $z_2/F_2$  is dropping at a  $(-1/\omega^2)$  rate and has a higher gain of  $(1/m)$  instead of  $(1/3m)$ . Thus, the low and high frequency asymptotes look exactly like  $z_1/F_1$ .

Summarizing the low and high frequency asymptotes, and solving for the gains and phases at  $\omega = 0.1$  rad/sec and  $\omega = 10$  rad/sec.

$$\left. \frac{z_1}{F_1} \right| = \frac{-1}{3m\omega^2} = \frac{-1}{3m(0.1)^2} = \frac{-1}{3(.01)} = \frac{-100}{3} = -33 = 30.46 \text{ db}, 180^\circ \quad (3.20)$$

$\omega = 0.1 \frac{\text{rad}}{\text{sec}}$

$$\left. \frac{z_1}{F_1} \right| = \frac{-1}{m\omega^2} = \frac{-1}{(10)^2} = \frac{-1}{100} = -.01 = -40 \text{ db}, 180^\circ \quad (3.21)$$

$\omega = 10 \frac{\text{rad}}{\text{sec}}$

$$\left. \frac{z_2}{F_1} \right| = \frac{-1}{3m\omega^2} = 30.46 \text{ db}, 180^\circ \quad (3.22)$$

$$\omega = 0.1$$

$$\left. \frac{z_2}{F_1} \right| = \frac{k}{m^2\omega^4} = \frac{1}{(10)^4} = 0.0001 = -80 \text{ db}, 0^\circ \quad (3.23)$$

$$\omega = 10$$

$$\left. \frac{z_3}{F_1} \right| = \frac{-1}{3m\omega^2} = 30.46 \text{ db}, 180^\circ \quad (3.24)$$

$$\omega = 0.1$$

$$\left. \frac{z_3}{F_1} \right| = \frac{-k^2}{m^3\omega^6} = \frac{-1}{1e^6} = -1e^{-6} = -120 \text{ db}, 180^\circ \quad (3.25)$$

$$\omega = 10$$

$$\left. \frac{z_2}{F_2} \right| = \frac{-1}{3m\omega^2} = 30.46 \text{ db}, 180^\circ \quad (3.26)$$

$$\omega = 0.1$$

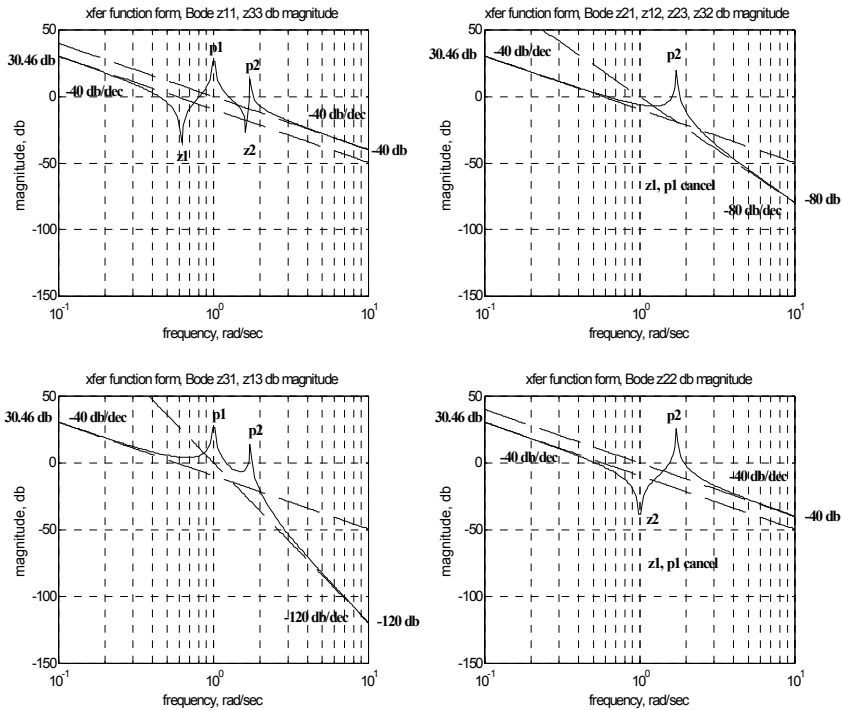
$$\left. \frac{z_2}{F_2} \right| = \frac{-1}{m\omega^2} = \frac{-1}{(10)^2} = -.01 = -40 \text{ db}, 180^\circ \quad (3.27)$$

$$\omega = 10$$

### 3.3 Hand Sketching Frequency Responses

Knowing the pole and zero locations and the asymptotes, the complete frequency response can be sketched by hand, as shown in [Figure 3.2](#). We will not worry about the exact magnitudes at the poles and zeros, but will use the hand sketch to get an idea of the overall shape and characteristics of the frequency response. Start by drawing the low and high frequency asymptotes, straight lines with appropriate magnitudes and slopes starting at the 0.1 and 10 rad/sec frequencies. Next, locate the poles and zeros at some distance above and below the asymptote line at the appropriate frequency and start “connecting the dots.” Start at the low frequency asymptote and follow it to the first zero or pole encountered. Keep plotting, moving to the next higher frequency pole or zero until all the poles/zeros are passed and move onto the high frequency asymptote. Note that for z21 the pole and zero at 1 rad/sec cancel as do one of the zeros and the pole for z22. Note that z31 has no zeros,

only poles. Compare these plots to the MATLAB generated plots in [Figure 3.5](#). Chapter 4 will give a physical interpretation of the zeros.



**Figure 3.2:** Hand sketch of frequency responses using asymptotes and pole/zero locations.

### 3.4 Interpreting Frequency Response Graphically in Complex Plane

There are many ways to plot frequency responses using MATLAB, as shown in the MATLAB code `tdofxfer.m` in the next section. One method of visualizing graphically what happens in calculating a frequency response is shown below.

In Chapter 2 we defined the four unique transfer functions in both “transfer function” and “zpk” forms. We will use the zpk form to graphically compute the frequency response.

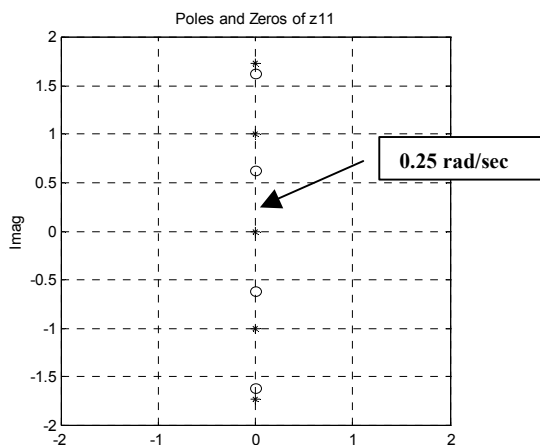
Start by defining a specific frequency for which to calculate the magnitude and phase. Then locate that frequency on the positive imaginary axis.

The gain and phase of the numerator term of a transfer function is the vector product of distances from all the zeros to the frequency of interest times the dc gain. Consider an undamped model, where all the poles and zeros lie on the imaginary axis. If the frequency happens to lie on a zero, that distance is zero, which multiplies all the other zero distances, resulting in a frequency response magnitude of zero. For a damped model the distance will not be zero, as the zeros are to the left of the imaginary axis, but the distance will be small, giving a small multiplier at that frequency and attenuating the response.

The gain and phase of the denominator term is the product of distances from all the poles to the frequency of interest. For an undamped model, if the frequency happens to lie on a pole, that distance is zero, which multiplies all the other pole distances. When the numerator is divided by the zero denominator value, the response goes to  $\infty$ . For a damped model the distance will not be zero as the poles are to the left of the imaginary axis; the distance will be small, however, giving a small multiplier at that frequency and amplifying the response.

Once the numerator and denominator are known, a vector division will give the transfer function.

The pole/zero plot, pole/zero values and zpk form for the z11 transfer function are shown below. We will calculate the frequency response for 0.25 rad/sec, where the frequency is indicated in [Figure 3.3](#).



**Figure 3.3: Interpreting the frequency response graphically for a frequency of 0.25 rad/sec (tdofpz3x3.m).**

```

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

```

**Table 3.1: Poles and zeros of z11 transfer function, MATLAB listing from tdfpz3x3.m.**

$$\begin{array}{l}
 \text{z11 Undamped Zero/pole/gain:} \\
 \frac{(s^2 + 0.382)(s^2 + 2.618)}{s^2 (s^2 + 1) (s^2 + 3)}
 \end{array}$$

**Table 3.2: zpk form of z11 transfer function, MATLAB listing from tdfpz3x3.m.**

Taking the expression for z11 from the zpk MATLAB listing in [Table 3.2](#), expand the terms to show explicitly the pole and zero values from [Table 3.1](#), substituting  $s=0.25j$  to calculate the frequency response value at 0.25 rad/sec.

$$\begin{aligned}
 z11 &= \frac{(s^2 + 0.382)(s^2 + 2.618)}{s^2 (s^2 + 1)(s^2 + 3)} \\
 &= \frac{(s + 0.618j)(s - 0.618j)(s + 1.618j)(s - 1.618j)}{s^2 (s + 1j)(s - 1j)(s + 1.732j)(s - 1.732j)} \quad (3.28) \\
 &= \frac{(0.25j + 0.618j)(0.25j - 0.618j)(0.25j + 1.618j)(0.25j - 1.618j)}{(0.25j)^2 (0.25j + 1j)(0.25j - 1j)(0.25j + 1.732j)(0.25j - 1.732j)} \\
 &= \frac{-0.816}{0.172} = -4.74
 \end{aligned}$$

Taking the magnitude and phase of  $z_{11}$ :

$$\begin{aligned} |z_{11}| &= 4.74 \\ \angle z_{11} &= -180^\circ \end{aligned} \tag{3.29}$$

The frequency response plot from MATLAB code `tdofxfer.m` in Figure 3.4 shows a magnitude of 4.79 (our 4.74 above differs because of rounding errors). The phase plot, not shown here but available by running `tdofxfer.m`, shows  $-180^\circ$ .

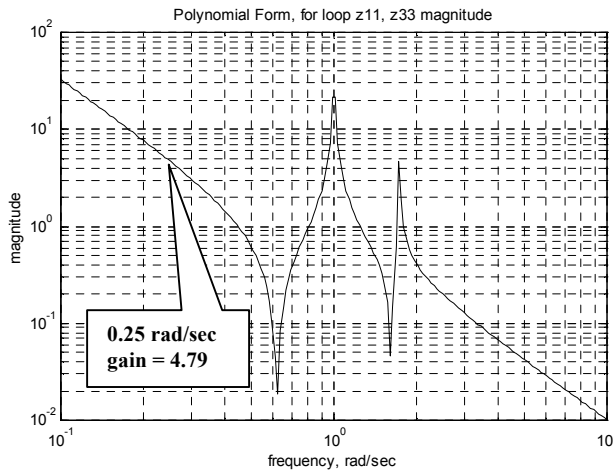


Figure 3.4:  $z_{11}$  frequency response highlighting magnitude at 0.25 rad/sec.

### 3.5 MATLAB Code `tdofxfer.m` – Plot Frequency Responses

#### 3.5.1 Code Description

Five different methods of calculating the frequency responses are used in the `tdofxfer.m` code, starting with the simplest and most straightforward method, but not necessarily the most efficient, then going to more sophisticated and efficient methods. The methods are:

- 1) Polynomial descriptions of the transfer functions: Using a for-loop to cycle through the frequency vector. MATLAB’s complex algebra capabilities are used to evaluate the frequency response at each frequency.
- 2) Polynomial descriptions of the transfer functions: Using MATLAB’s vector capabilities instead of a for-loop to

calculate the frequency response at the frequencies in the frequency vector.

- 3) MATLAB's "transfer function" representations of the transfer functions: MATLAB's automatic bode plotting capability is used, where MATLAB chooses the frequency range to use and automatically plots results.
- 4) Transfer function representations of the transfer functions: MATLAB's bode plotting capability is used, but this time defining outputs and frequency range with the "bode" command, controlling the output for later plotting.
- 5) MATLAB's "zero/pole/gain, zpk" form of the input is used.

Because the plotting commands are so lengthy, they will be not be listed. See the downloaded code for the complete code listing.

### 3.5.2 Polynomial Form, For-Loop Calculation, Code Listing

The "polynomial form" shown below uses (2.28) through (2.36) to define the four distinct frequency responses of the system, allowing the user to specify **any** values of masses, dampers and springs. MATLAB's complex number calculation capabilities are used by defining a vector of radian frequencies "w" and substituting "j\*w" for "s." A "for-loop" is then used to cycle through each frequency in the "w" vector and calculate the complex value for the frequency response at that frequency. Because MATLAB does not know how large all of the vectors defined within the "for-loop" are going to be, it resizes each vector during each calculation, a very time-consuming (relatively speaking) operation. We could speed up the operation by defining null vectors of the proper size for each of the "for-loop" variables before the for-loop was entered. This would still require going through the for-loop for every entry in the "w" vector, but would eliminate having to resize the vectors at each calculation. Following the for-loop, magnitudes and phases are calculated using MATLAB's "abs" and "angle" commands and are available for plotting.

```
%      "Polynomial Form, for-loop" frequency response plotting
%      assign values for masses, damping, and stiffnesses

m1 = 1;
m2 = 1;
m3 = 1;
c1 = 0;
c2 = 0;
k1 = 1;
k2 = 1;
```

```

% 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 200 defines 200 frequency points.

w = logspace(-1,1,200);

% pre-calculate the radians to degree conversion

rad2deg = 180/pi;

% Use a for-loop to cycle through all the frequencies, using MATLAB's
% complex algebra capabilities to evaluate.

for cnt = 1:length(w)

% define s as the imaginary operator times each frequency

s = j*w(cnt);

% define the frequency responses to be evaluated

den(cnt) = s^2*(s^4*(m1*m2*m3) + s^3*(m2*m3*c1 + m1*m3*c1 + m1*m2*c2
+ m1*m3*c2) + s^2*(m1*m3*k1 + m1*m3*k2 + m1*m2*k2 ...
+ m2*c1*c2 + m3*c1*c2 + m1*c1*c2 + k1*m2*m3) ...
+ s*(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));

z11bf(cnt) = ((m2*m3)*s^4 + (m3*c1 + m3*c2 + m2*c2)*s^3 ...
+ (c1*c2 + m2*k2 + m3*k1 + m3*k2)*s^2 ...
+ (c1*k2 + c2*k1)*s + (k1*k2))/den(cnt);

z21bf(cnt) = ((m3*c1)*s^3 + (c1*c2 + m3*k1)*s^2 + (c1*k2 + c2*k1)*s ...
+ (k1*k2))/den(cnt);

z31bf(cnt) = ((c1*c2)*s^2 + (c1*k2 + c2*k1)*s + (k1*k2))/den(cnt);

z22bf(cnt) = ((m1*m3)*s^4 + (m1*c2 + m3*c1)*s^3 + (m1*k2 + c1*c2 + ...
m3*k1)*s^2
+ (c1*k2 + c2*k1)*s + (k1*k2))/den(cnt);

% calculate the magnitude and phase of each frequency response

z11bfmag(cnt) = abs(z11bf(cnt));

z21bfmag(cnt) = abs(z21bf(cnt));

z31bfmag(cnt) = abs(z31bf(cnt));

z22bfmag(cnt) = abs(z22bf(cnt));

z11bfphs(cnt) = angle(z11bf(cnt))*rad2deg;

z21bfphs(cnt) = angle(z21bf(cnt))*rad2deg;

```

```

z31bfphs(cnt) = angle(z31bf(cnt))*rad2deg;

z22bfphs(cnt) = angle(z22bf(cnt))*rad2deg;

end           %           end of for-loop

```

### 3.5.3 Polynomial Form, Vector Calculation, Code Listing

This section of code defines the transfer functions as in the previous section but instead of using the for-loop for obtaining complex values of the desired quantities at each frequency, this code uses MATLAB's vector calculation capability. MATLAB can perform operations on vectors directly, very quickly and without having to resize anything as discussed in the previous section. In order to define a vector operation between two vectors, precede the operation symbol (\*, /, ^, etc) with a period ("."). This period tells MATLAB to perform an element-by-element operation on or between corresponding elements of the vector(s). For example, to square every element of a vector, "vec", use the command "vec.^2," and to multiply two elements, "vec1" and "vec2" element by element, use the command "vec1.\*vec2." This vector calculation capability will be used wherever appropriate in the balance of the code in the text.

```

%      "Polynomial Form, Vector" method - using MATLAB's vector capabilities instead
%      of the "for" loop.

%      assign values for masses, damping, and stiffnesses

m1 = 1;
m2 = 1;
m3 = 1;
c1 = 0;
c2 = 0;
k1 = 1;
k2 = 1;

%      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 200 defines 200 frequency points.

w = logspace(-1,1,200);

%      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 the frequency responses to be evaluated, using the "." prefix
%      in front of each operator to indicate that each

%      define the frequency responses to be evaluated

den = s.^2.*(s.^4*(m1*m2*m3) + s.^3*(m2*m3*c1 + m1*m3*c1 + m1*m2*c2
+ m1*m3*c2) + s.^2*(m1*m3*k1 + m1*m3*k2 + m1*m2*k2 ...
+ m2*c1*c2 + m3*c1*c2 + m1*c1*c2 + k1*m2*m3) ...
+ s*(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));

z11bfv = ((m2*m3)*s.^4 + (m3*c1 + m3*c2 + m2*c2)*s.^3 ...
+ (c1*c2 + m2*k2 + m3*k1 + m3*k2)*s.^2 ...
+ (c1*k2 + c2*k1)*s + (k1*k2))./den;

z21bfv = ((m3*c1)*s.^3 + (c1*c2 + m3*k1)*s.^2 + (c1*k2 + c2*k1)*s ...
+ (k1*k2))./den;

z31bfv = ((c1*c2)*s.^2 + (c1*k2 + c2*k1)*s + (k1*k2))./den;

z22bfv = ((m1*m3)*s.^4 + (m1*c2 + m3*c1)*s.^3 + (m1*k2 + c1*c2 + m3*k1)*s.^2
+ (c1*k2 + c2*k1)*s + (k1*k2))./den;

%      calculate the magnitude and phase of each frequency response

z11bfvmag = abs(z11bfv);

z21bfvmag = abs(z21bfv);

z31bfvmag = abs(z31bfv);

z22bfvmag = abs(z22bfv);

z11bfvphs = angle(z11bfv)*rad2deg;

z21bfvphs = angle(z21bfv)*rad2deg;

z31bfvphs = angle(z31bfv)*rad2deg;

z22bfvphs = angle(z22bfv)*rad2deg;

```

### 3.5.4 Transfer Function Form – Bode Calculation, Code Listing

This section uses MATLAB's automatic "bode" calculation and plotting capability, as well as the "transfer function" form of input, where the numerator "num" and denominator "den" of each transfer function are input as row vectors in coefficients of descending powers of "s." Using the "bode" command with no left-hand arguments results in MATLAB choosing the frequency range to use and automatically generating plots of magnitude and phase.

```

%      using MATLAB's automatic "bode" plotting capability, defining the transfer
%      functions in "transfer function" form by row vectors of coefficients of "s"

%      assign values for masses, damping, and stiffnesses

m1 = 1;
m2 = 1;
m3 = 1;
c1 = 0;
c2 = 0;
k1 = 1;
k2 = 1;

%      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)];

%      the bode command with no left hand side arguments automatically chooses
%      frequency limits and plots results

grid on
bode(z11num,den);

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

bode(z21num,den);

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

bode(z31num,den);

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

bode(z22num,den);

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

```

### 3.5.5 Transfer Function Form, Bode Calculation with Frequency, Code Listing

This section also uses MATLAB's "bode" plotting capability with the transfer function form of the input but defines magnitude and phase vectors for output and specifies the frequency vector to use. This code also calculates and plots the low and high frequency asymptotes for the four unique transfer functions.

```

%      using MATLAB's "bode" plotting capability, defining the transfer
%      functions in "transfer function" form by row vectors of coefficients of
%      "s" and defining output vectors for magnitude and phase as well as a
%      defined range of radian frequencies

%      assign values for masses, damping, and stiffnesses

m1 = 1;
m2 = 1;
m3 = 1;
c1 = 0;
c2 = 0;
k1 = 1;
k2 = 1;

%      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)];

%      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 200 defines 200 frequency points.

w = logspace(-1,1,200);

%      calculate the rigid-body motions for low and high frequency portions
%      of all the frequency responses, the denominator entries are vectors with
%      entries being coefficients of the "s" terms in the low and high frequency
%      asymptotes, starting with the highest power of "s" and ending with the
%      "0"th power of "s" or the constant term

```

```
z11num_lo = [1];
z11den_lo = [3 0 0];           % -1/(3*w^2)
z11num_hi = [1];
z11den_hi = [1 0 0];         % -1/(w^2)
z21num_lo = [1];
z21den_lo = [3 0 0];         % -1/(3*w^2)
z21num_hi = [1];
z21den_hi = [1 0 0 0 0];     % -1/(3*w^4)
z31num_lo = [1];
z31den_lo = [3 0 0];         % -1/(3*w^2)
z31num_hi = [1];
z31den_hi = [1 0 0 0 0 0];   % -1/(w^2)
z22num_lo = [1];
z22den_lo = [3 0 0];         % -1/(3*w^2)
z22num_hi = [1];
z22den_hi = [1 0 0];         % -1/(w^2)
% define the "tf" models from "num, den" combinations
z11tf = tf(z11num,den);
z21tf = tf(z21num,den);
z31tf = tf(z31num,den);
z22tf = tf(z22num,den);
z11tf_lo = tf(z11num_lo,z11den_lo);
z11tf_hi = tf(z11num_hi,z11den_hi);
z21tf_lo = tf(z21num_lo,z21den_lo);
z21tf_hi = tf(z21num_hi,z21den_hi);
z31tf_lo = tf(z31num_lo,z31den_lo);
z31tf_hi = tf(z31num_hi,z31den_hi);
```

```

z22tf_lo = tf(z22num_lo,z22den_lo);
z22tf_hi = tf(z22num_hi,z22den_hi);
% use the bode command with left hand magnitude and phase vector arguments
% to provide values for further analysis/plotting

[z11mag,z11phs] = bode(z11tf,w);
[z21mag,z21phs] = bode(z21tf,w);
[z31mag,z31phs] = bode(z31tf,w);
[z22mag,z22phs] = bode(z22tf,w);

[z11maglo,z11phslo] = bode(z11tf_lo,w);
[z21maglo,z21phslo] = bode(z21tf_lo,w);
[z31maglo,z31phslo] = bode(z31tf_lo,w);
[z22maglo,z22phslo] = bode(z22tf_lo,w);

[z11maghi,z11phshi] = bode(z11tf_hi,w);
[z21maghi,z21phshi] = bode(z21tf_hi,w);
[z31maghi,z31phshi] = bode(z31tf_hi,w);
[z22maghi,z22phshi] = bode(z22tf_hi,w);
% calculate the magnitude in decibels, db

z11magdb = 20*log10(z11mag);
z21magdb = 20*log10(z21mag);
z31magdb = 20*log10(z31mag);
z22magdb = 20*log10(z22mag);

z11maglodb = 20*log10(z11maglo);
z21maglodb = 20*log10(z21maglo);
z31maglodb = 20*log10(z31maglo);
z22maglodb = 20*log10(z22maglo);

z11maghidb = 20*log10(z11maghi);
z21maghidb = 20*log10(z21maghi);
z31maghidb = 20*log10(z31maghi);

```

```
z22maghidb = 20*log10(z22maghi);
```

### 3.5.6 Zero/Pole/Gain Function Form, Bode Calculation with Frequency, Code Listing

This section also uses MATLAB's "bode" plotting capability. This time, with the zero/pole/gain form of the input. It defines magnitude and phase vectors for output and specifies the frequency vector to use.

```
%      using MATLAB's "bode" plotting capability, defining the transfer
%      functions in "zero/pole/gain" form by column vectors of poles and zeros
%      and defining output vectors for magnitude and phase as well as a
%      defined range of radian frequencies

%      assign values for masses, damping, and stiffnesses

m1 = 1;
m2 = 1;
m3 = 1;
c1 = 0;
c2 = 0;
k1 = 1;
k2 = 1;

m = m1;
k = k1;

%      define column vectors of poles and zeros from previous derivation
%
%      there are three ways to make a column vector:
%
%      1)      define a row vector and then transpose it:
%
%      p = [0 0 1*j -1*j sqrt(3*k/m)*j -sqrt(3*k/m)*j]';
%
%      2)      define a column vector by using semi-colons between elements:
%
%      p = [0; 0; 1*j; -1*j; sqrt(3*k/m)*j; -sqrt(3*k/m)*j];
%
%      3)      define a column vector directly:
%
%      p = [      0
%             0
%             1*j
%            -1*j
%           sqrt(3*k/m)*j
%          -sqrt(3*k/m)*j ];
%
%      zeros for z1/f1; quartic so four zeros

z1_1 = -sqrt((-3*k-sqrt(5)*k)/(2*m));
```

```

z11_2 = sqrt((-3*k-sqrt(5)*k)/(2*m));
z11_3 = -sqrt((-3*k+sqrt(5)*k)/(2*m));
z11_4 = sqrt((-3*k+sqrt(5)*k)/(2*m));
% zeros for z2/f1; quadratic so two zeros
z21_1 = -sqrt(-k/m);
z21_2 = sqrt(-k/m);
% zeros for z3/f1; no zeros, so use empty brackets
z31_1 = [];
% zeros for z2/f2: quadratic so two zeros
z22_1 = -sqrt(-k/m);
z22_2 = sqrt(-k/m);
%
z11 = [z11_1 z11_2 z11_3 z11_4];
z21 = [z21_1 z21_2];
z31 = z31_1;
z22 = [z22_1 z22_2];
p = [0 0 1*j -1*j sqrt(3*k/m)*j -sqrt(3*k/m)*j]';
gain = 1;
% use the zpk command to define the four pole/zero/gain systems
sys11pz = zpk(z11,p,gain);
sys21pz = zpk(z21,p,gain);
sys31pz = zpk(z31,p,gain);
sys22pz = zpk(z22,p,gain);
% 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 200 defines 200 frequency points.
w = logspace(-1,1,200);
% use the bode command with left hand magnitude and phase vector arguments
% to provide values for further analysis/plotting
[z11mag,z11phs] = bode(sys11pz,w);
[z21mag,z21phs] = bode(sys21pz,w);

```

```

[z31mag,z31phs] = bode(sys31pz,w);
[z22mag,z22phs] = bode(sys22pz,w);
% calculate the magnitude in decibels, db
z11magdb = 20*log10(z11mag);
z21magdb = 20*log10(z21mag);
z31magdb = 20*log10(z31mag);
z22magdb = 20*log10(z22mag);

```

### 3.5.7 Code Output – Frequency Response Magnitude and Phase Plots

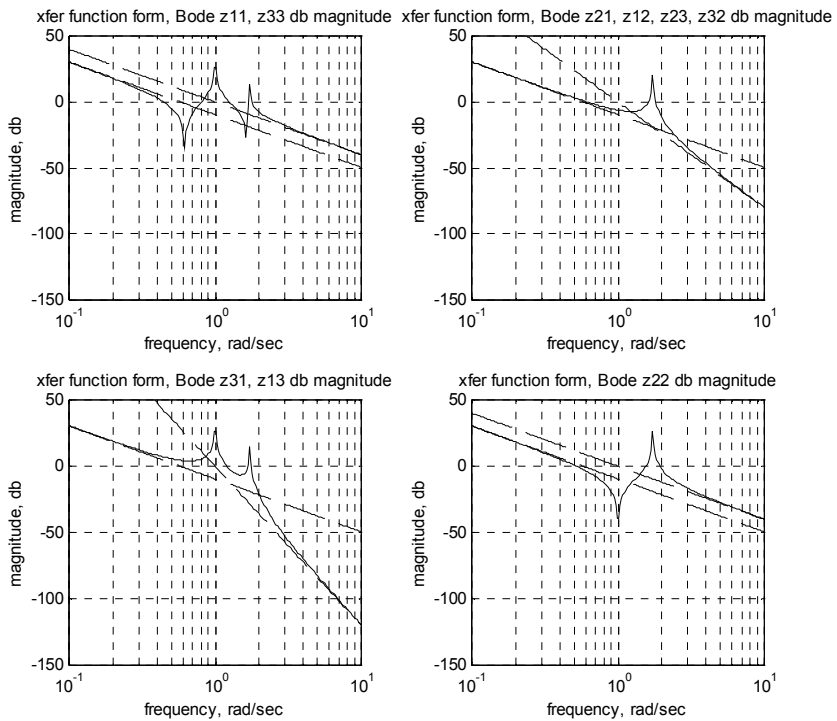
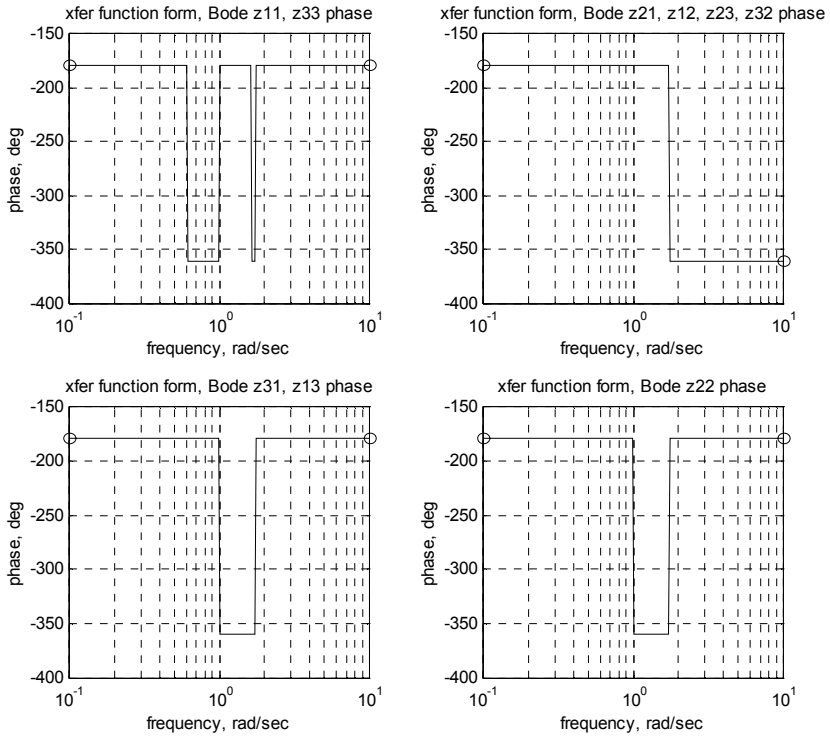


Figure 3.5: Magnitude versus frequency for four distinct frequency responses, including low and high frequency asymptotes.

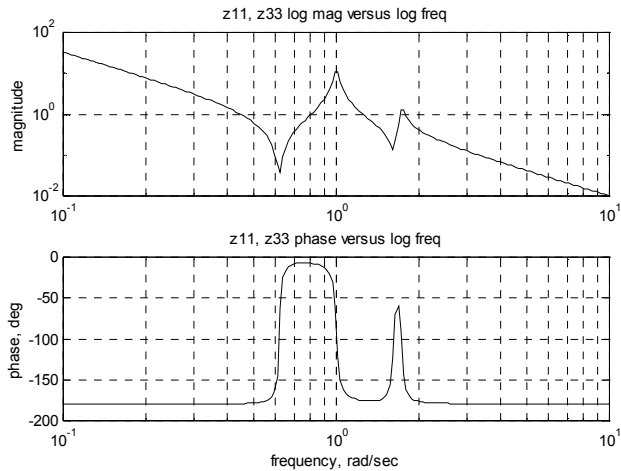


**Figure 3.6: Phase versus frequency for four distinct frequency responses, including low and high frequency asymptotes.**

### 3.6 Other Forms of Frequency Response Plots

Other forms of frequency response plots are shown for a damping value of 2% of critical damping for each mode. The code used for the plots is from Chapter 11, `tdofss_modal_xfer_modes.m`.

### 3.6.1 Log Magnitude versus Log Frequency



**Figure 3.7: Log magnitude versus log frequency.**

Comments on the log-log plot:

- 1) The asymptotic behavior at the low and high frequency ends are clear by checking the slopes.
- 2) The log frequency scale spreads out the resonances, which otherwise would tend to clump at the lower end of the scale.
- 3) The log amplitude scale allows reading the gain directly without converting from db.
- 4) Adding the gain from the mechanics to the gain of the frequency response of the control system allows for definition of the overall series (multiplicative) frequency response.

### 3.6.2 db Magnitude versus Log Frequency

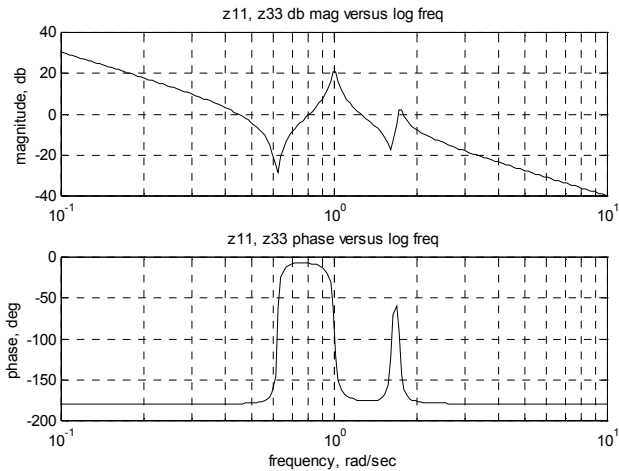


Figure 3.8: db magnitude versus log frequency.

Comments on the db-log plot:

- 1) The asymptotic behaviors at the low and high frequency ends are clear by checking the slopes, i.e.  
 $(1/\omega) = -20 \text{ db/decade}$ ,  $(1/\omega^2) = -40 \text{ db/decade}$ .
- 2) The log frequency scale spreads out the resonances, which otherwise would tend to clump at the lower end of the scale.
- 3) The db amplitude scale makes it necessary to convert to gain if needed.
- 4) The **product** of two individual frequency response gains can be found by **adding** their gains directly on the log scale.

### 3.6.3 db Magnitude versus Linear Frequency

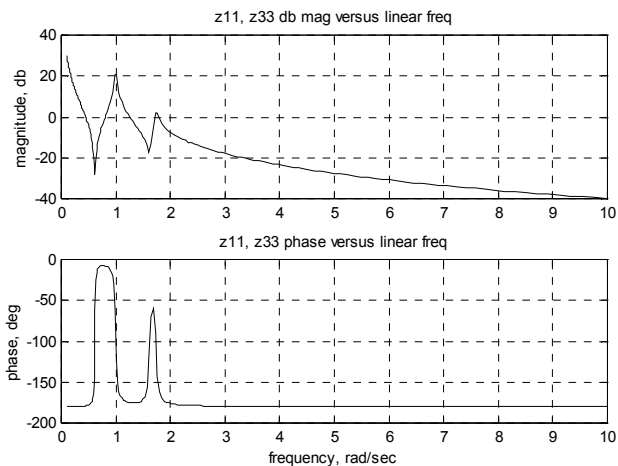
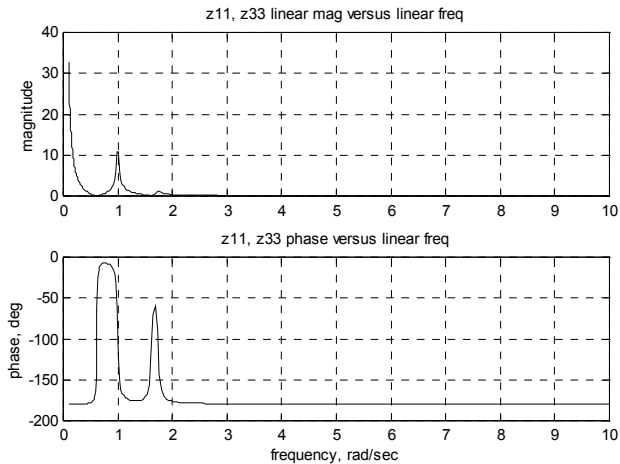


Figure 3.9: db magnitude versus linear frequency.

Comments on the db-linear plot:

- 1) The asymptotic behaviors at the low and high frequency ends are not clear.
- 2) The linear frequency scale tends to clump the resonances at the lower end of the scale, although the scale could be shortened since nothing significant is happening at the high end.
- 3) The db amplitude scale makes it necessary to convert to linear gain if specific gain values are needed.

### 3.6.4 Linear Magnitude versus Linear Frequency



**Figure 3.10: Linear magnitude versus linear frequency.**

Comments on the linear-linear plot:

- 1) The asymptotic behaviors at the low and high frequency ends are not clear.
- 2) The linear frequency scale tends to clump the resonances at the lower end of the scale, although the scale could be shortened since nothing significant is happening at the high end.
- 3) The linear amplitude scale enables reading gain values directly, but reading values for small gain values is difficult.
- 4) It is useful for directly adding the individual mode contributions of a frequency response to provide the overall response, shown in Chapter 8, Sections 8.7 and 8.8.

### 3.6.5 Real and Imaginary Magnitudes versus Log and Linear Frequency

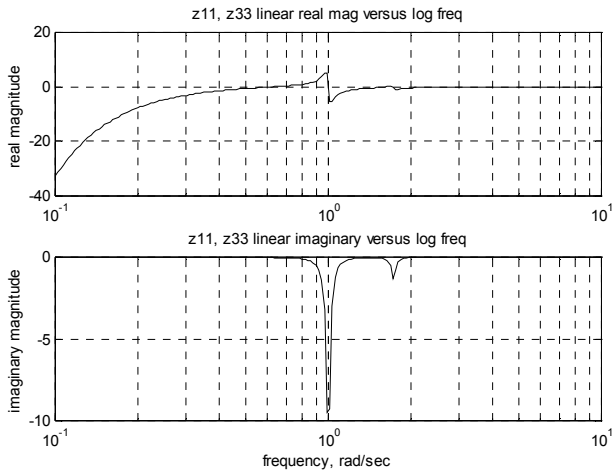


Figure 3.11: Real and imaginary magnitudes versus log frequency.

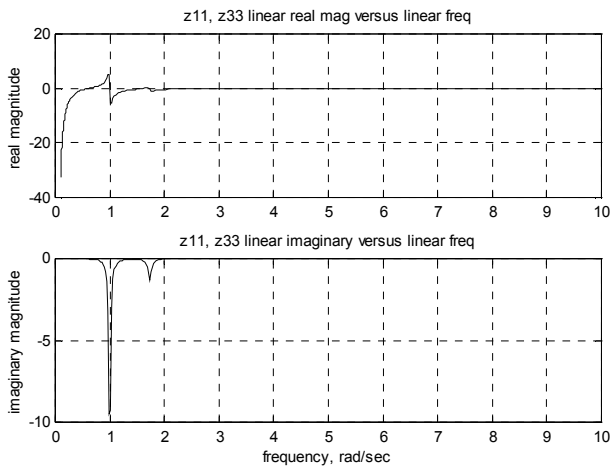


Figure 3.12: Real and imaginary magnitude versus linear frequency.

Comments on real versus linear frequency, imaginary versus linear frequency:

- 1) These plots are useful in understanding the amplitudes of transfer functions at resonance, as the peaks of the imaginary curve represent the amplitude at resonance.

- 2) While the imaginary plot peaks at each resonance, the real plot goes through zero at each resonance.

### 3.6.6 Real versus Imaginary (Nyquist)

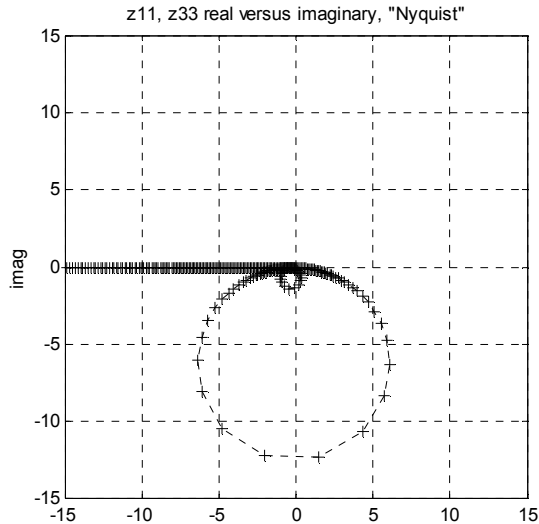


Figure 3.13: Real versus imaginary (Nyquist).

Comments on real versus imaginary:

- 1) Frequency is not plotted directly on the real/imaginary plot; each point on the plot represents a different frequency.
- 2) Plotting real versus imaginary is a very useful technique when identifying resonant characteristics. The two resonances can be readily seen, helping in identifying closely spaced resonances.
- 3) One method of identifying damping in a mode is to use the rate of change of amplitude versus frequency (Maia 1997).

### 3.7 Solving for Eigenvectors (Mode Shapes) Using the Transfer Function Matrix

We have reviewed transfer functions, poles, zeros and frequency responses. The next area we will cover in order to completely define the system is **eigenvectors**, or **mode shapes**. At each natural frequency, the eigenvector defines the relative motion between degrees of freedom. Understanding the distribution of motion in each mode of vibration is essential in order to intelligently modify the system's resonant characteristic to solve resonance problems.

Since eigenvectors define the **relative** motion between degrees of freedom, we need to choose a degree of freedom against which to measure the other motions. We can find the relative motion using **any** column of the transfer function matrix. Choosing  $z_1$  as the reference and solving for  $z_2/z_1$  and  $z_3/z_1$  using the first column of the transfer function matrix (we will compare results using the second column later to show that they give the same results):

$$\frac{\frac{z_2}{F_1}}{\frac{z_1}{F_1}} = \frac{z_{21}}{z_{11}} = \frac{mks^2 + k^2}{m^2s^4 + 3mks^2 + k^2} \quad (3.29)$$

$$\frac{\frac{z_3}{F_1}}{\frac{z_1}{F_1}} = \frac{z_{31}}{z_{11}} = \frac{k^2}{m^2s^4 + 3mks^2 + k^2} \quad (3.30)$$

Now that the ratios are known, we substitute the resonant frequencies (pole values) one at a time to define the mode shape at that frequency, dropping the second index,  $z_{21} \rightarrow z_2$ .

For mode 1: evaluated at  $s = j\omega_1 = 0$

$$\frac{z_2}{z_1} = \frac{mks^2 + k^2}{m^2s^4 + 3mks^2 + k^2} = \frac{k^2}{k^2} = 1 \quad (3.31)$$

$$z_2 = z_1 \quad (3.32)$$

$$\frac{z_3}{z_1} = \frac{k^2}{m^2 s^4 + 3mks^2 + k^2} = \frac{k^2}{k^2} = 1 \quad (3.33)$$

$$z_3 = z_1 \quad (3.34)$$

The interpretation of this mode shape is that at  $\omega_1$  the ratios of motion of mass 2 and mass 3 to mass 1 are equal and are equal to 1. This is the rigid body mode at 0 hz.

For mode 2: evaluated at  $s = j\omega_2 = j\sqrt{\frac{k}{m}}$

$$\frac{z_2}{z_1} = \frac{mks^2 + k^2}{m^2 s^4 + 3mks^2 + k^2} = \frac{mk\left(\frac{-k}{m}\right) + k^2}{m^2\left(\frac{k^2}{m^2}\right) + 3mk\left(\frac{-k}{m}\right) + k^2} = \frac{0}{k^2} \quad (3.35)$$

$$z_2 = 0 \quad (3.36)$$

$$\frac{z_3}{z_1} = \frac{k^2}{m^2 s^4 + 3mks^2 + k^2} = \frac{k^2}{m^2\left(\frac{k^2}{m^2}\right) + 3mk\left(\frac{-k}{m}\right) + k^2} = -1 \quad (3.37)$$

$$z_3 = -z_1 \quad (3.38)$$

The interpretation of this mode shape is that at  $\omega_2$  mass 2 has zero motion relative to mass 1 (it is stationary). Mass 3 is moving out of phase with mass 1 with equal amplitude.

For mode 3: evaluated at  $s = j\omega_3 = j\sqrt{\frac{3k}{m}}$

$$\begin{aligned} \frac{z_2}{z_1} &= \frac{mks^2 + k^2}{m^2 s^4 + 3mks^2 + k^2} = \frac{mk\left(\frac{-3k}{m}\right) + k^2}{m^2\left(\frac{9k^2}{m^2}\right) + 3mk\left(\frac{-3k}{m}\right) + k^2} \\ &= \frac{-3k^2 + k^2}{9k^2 - 9k^2 + k^2} = \frac{-2k^2}{k^2} = -2 \end{aligned} \quad (3.39)$$

$$z_2 = -2z_1 \quad (3.40)$$

$$\begin{aligned} \frac{z_3}{z_1} &= \frac{k^2}{m^2 s^4 + 3mks^2 + k^2} = \frac{k^2}{m^2 \left( \frac{9k^2}{m^2} \right) + 3mk \left( \frac{-3k}{m} \right) + k^2} \\ &= \frac{k^2}{k^2} = 1 \end{aligned} \quad (3.41)$$

$$z_3 = z_1 \quad (3.42)$$

The interpretation of this mode shape is that at  $\omega_3$  mass 2 is moving with twice the motion of mass 1 and out of phase with it and mass 3 is moving in phase with mass 1 and with the same amplitude.

Showing that the second column of the transfer function matrix could have been used and would have given the same eigenvectors:

$$\frac{\frac{z_2}{F_2}}{\frac{z_1}{F_2}} = \frac{z_2}{z_1} = \frac{m^2 s^4 + 2mks^2 + k^2}{mks^2 + k^2} \quad (3.43)$$

$$\frac{\frac{z_3}{F_2}}{\frac{z_1}{F_2}} = \frac{z_3}{z_1} = \frac{mks^2 + k^2}{mks^2 + k^2} = 1 \quad (3.44)$$

For mode 1,  $\omega_1 = 0$

$$\begin{aligned} \frac{z_2}{z_1} &= \frac{k^2}{k^2} = 1 \\ \frac{z_3}{z_1} &= 1 \end{aligned} \quad (3.45a,b)$$

For mode 2, evaluated at  $s = j\omega_2 = j\sqrt{\frac{k}{m}}$

$$\frac{z_2}{z_1} = \frac{m^2 s^4 + 2mks^2 + k^2}{mks^2 + k^2} = \frac{m^2 \left( \frac{k^2}{m^2} \right) + 2mk \left( \frac{-k}{m} \right) + k^2}{mk \left( \frac{-k}{m} \right) + k^2} \quad (3.46)$$

$$= \frac{k^2 - 2k^2 + k^2}{-k^2 + k^2} = \frac{0}{0}$$

$$\frac{z_3}{z_1} = 1 \quad (3.47)$$

For mode 3,  $s = j\omega_3 = j\sqrt{\frac{3k}{m}}$

$$\frac{z_2}{z_1} = \frac{m^2 s^4 + 2mks^2 + k^2}{mks^2 + k^2} = \frac{m^2 \left( \frac{9k^2}{m^2} \right) + 2mk \left( \frac{-3k}{m} \right) + k^2}{mk \left( \frac{-3k}{m} \right) + k^2} \quad (3.48)$$

$$= \frac{9k^2 - 6k^2 + k^2}{-3k^2 + k^2} = \frac{4k^2}{-2k^2} = -2$$

$$\frac{z_3}{z_1} = 1 \quad (3.49)$$

Summarizing the mode shapes in the **modal matrix**,  $\mathbf{z}_m$ , where the first through third columns represent mode shapes for the first three modes, respectively, and the first through third rows show the relative motion for the first through third dof's, respectively:

$$\mathbf{z}_m = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & -2 \\ 1 & -1 & 1 \end{bmatrix} \quad (3.50)$$

Figure 3.14 shows the mode shapes pictorially. There are many different eigenvector scaling, or normalizing techniques, to be discussed later. It is not important which normalization technique is used in visualizing mode shapes. However, in using the modal matrix to calculate responses, the normalization technique used is critical, as we will see in future chapters.

Because there is no damping, these modes are known as “normal” (as opposed to “complex”) modes. With a normal mode, if the masses are started with some multiple of the displacements of one of the modes, the system will respond at only that frequency. During that motion, the masses will all reach their maximum and minimum points at the same time. Mode shapes are plotted in Figure 3.14, assuming an arbitrary value of 1 for  $z_1$  :

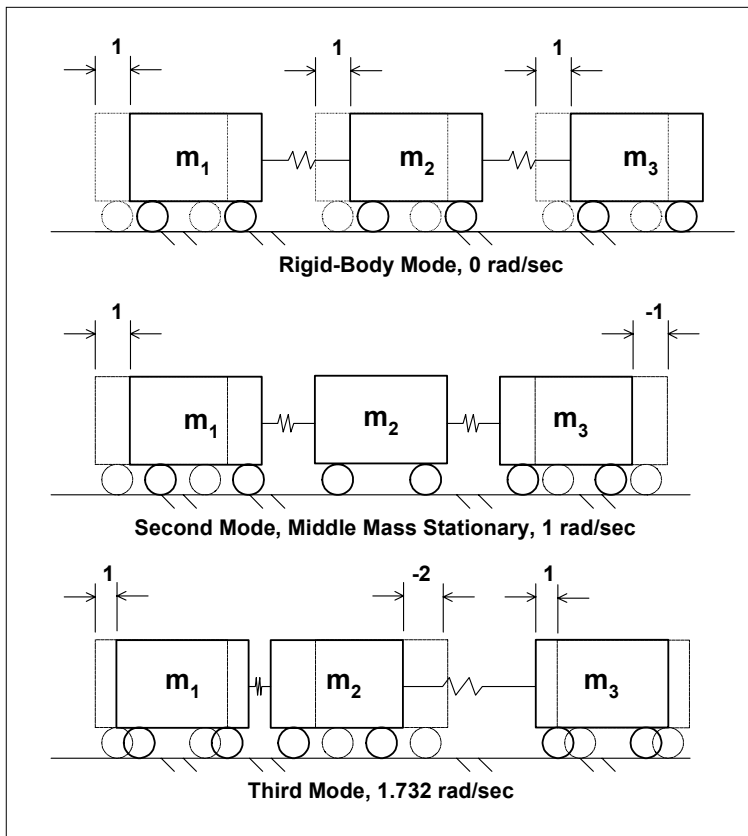


Figure 3.14: Mode shape plots.

## Problems

Note: All the problems refer to the two dof system shown in [Figure P2.2](#).

P3.1 Set  $m_1 = m_2 = m = 1$ ,  $k_1 = k_2 = k = 1$  and hand sketch the frequency responses for the undamped system.

P3.2 (MATLAB) Set  $m_1 = m_2 = m = 1$ ,  $k_1 = k_2 = k = 1$ , modify the **tdofxfer.m** code and plot the frequency responses of the two dof undamped system using the transfer function and zero/pole/gain forms of Sections 3.5.5 and 3.5.6.

P3.3 (MATLAB) Set  $m_1 = m_2 = m = 1$ ,  $k_1 = k_2 = k = 1$ , add damping to the model from P3.2 and plot the transfer functions in Nyquist form, being careful to use small enough frequency spacing to identify the resonances as shown in [Figure 3.13](#).

P3.4 (MATLAB) Set  $m_1 = m_2 = m = 1$ ,  $k_1 = k_2 = k = 1$ , choose one of the transfer functions for the undamped system and plot the poles and zeros in the s-plane. Choose a frequency on the positive imaginary axis and hand calculate the gain at that frequency. Correlate with the MATLAB calculated gain.

P3.5 Solve for the two eigenvectors for the system in P3.3 using the transfer function matrix. Hand plot the mode shapes as in [Figure 3.14](#).