CHAPTER 12

TIME DOMAIN: MODAL STATE SPACE FORM

12.1 Introduction

In Chapter 7 we derived the equations of motion in modal form for the system in Figure 12.1. In this chapter we will convert the modal form to state space modal form and obtain the closed form transient solution for the forcing function and initial conditions described in Figure 12.1. MATLAB will then be used to solve the same equations using the ode45 function.

12.2 Equations of Motion – Modal Form

The applied step forces are as shown in Figure 12.1. The initial conditions of position and velocity for each of the three masses are displayed in Table 12.1, the same as Figure 9.1 and Table 9.1.

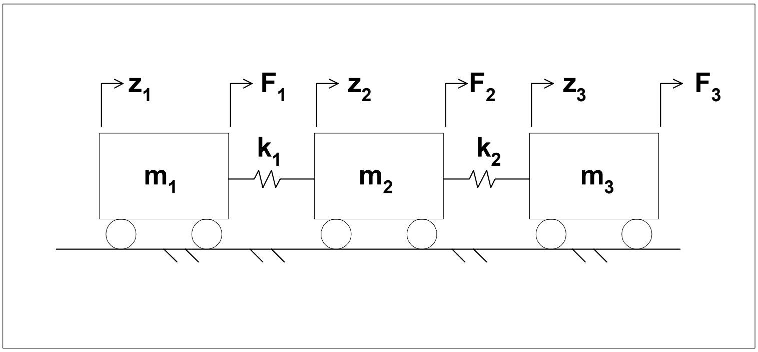

Figure 12.1: Step forces applied to tdof system.

Mass 1	Mass 2	Mass 3
$z_{01} = 0$	$z_{02} = -1$	$z_{03} = 1$
$\dot{z}_{01} = -1$	$\dot{z}_{02} = 2$	$\dot{z}_{03} = -2$

Table 12.1: Initial conditions applied to tdof system.

Repeating results from Chapter 9, where we developed the modal form of the equations of motion:

The force vector in principal coordinates from (9.8) is:

$$\mathbf{F}_p = \mathbf{z}_n^T \mathbf{F} = \begin{bmatrix} F_{p1} \\ F_{p2} \\ F_{p3} \end{bmatrix} = \frac{1}{\sqrt{m}} \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & 0 & \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{6}} & \frac{-2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ -2 \end{bmatrix} = \frac{1}{\sqrt{m}} \begin{bmatrix} \frac{-\sqrt{3}}{3} \\ \frac{3\sqrt{2}}{2} \\ \frac{-\sqrt{6}}{6} \end{bmatrix} \qquad (12.1)$$

With initial conditions from (9.6), (9.7):

$$\mathbf{z}_{po} = \sqrt{m} \begin{bmatrix} 0 \\ \frac{-\sqrt{2}}{2} \\ \frac{\sqrt{6}}{2} \end{bmatrix}, \quad \dot{\mathbf{z}}_{po} = \sqrt{m} \begin{bmatrix} \frac{-\sqrt{3}}{3} \\ \frac{\sqrt{2}}{2} \\ \frac{-7\sqrt{6}}{6} \end{bmatrix} \qquad (12.2)$$

Using the results of the eigenvalue solution, we can write the homogeneous equations of motion by inspection. The forcing function can be added to the right-hand side, knowing $\mathbf{F}_p$:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \qquad (12.3)$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\omega_2^2 & -2\zeta_2\omega_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -\omega_3^2 & -2\zeta_3\omega_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \begin{bmatrix} 0 \\ F_{p1} \\ 0 \\ F_{p2} \\ 0 \\ F_{p3} \end{bmatrix} u \qquad (12.4)$$

with initial conditions of:

$$\mathbf{x}_{po} = \begin{bmatrix} z_{po1} \\ \dot{z}_{po1} \\ z_{po2} \\ \dot{z}_{po2} \\ z_{p03} \\ \dot{z}_{p03} \end{bmatrix} = \sqrt{m} \begin{bmatrix} 0 \\ \frac{-\sqrt{3}}{3} \\ \frac{-\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \\ \frac{\sqrt{6}}{2} \\ \frac{-7\sqrt{6}}{6} \end{bmatrix} \tag{12.5}$$

12.3 Solving Equations of Motion Using Laplace Transforms

Now that we know the complete state space equations of motion in principal coordinates and the initial conditions on the six states in principal coordinates, the equations can be solved in the time domain. The first order equations of motion above are similar in nature to the second order equations of motion in Table 7.2. The three **sets** of first order equations in modal state space form are uncoupled as were the three second order equations of motion in modal form (7.89).

Expanding the three sets of equations:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= F_{p1}u \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= -\omega_2^2 x_3 - 2\zeta_2\omega_2 x_4 + F_{p2}u \\ \dot{x}_5 &= x_6 \\ \dot{x}_6 &= -\omega_3^2 x_5 - 2\zeta_3\omega_3 x_6 + F_{p3}u \end{aligned} \tag{12.6a-f}$$

Taking the Laplace transform of the first two equations above:

$$\begin{aligned} sx_1(s) - x_1(0) &= x_2(s) \\ sx_2(s) - x_2(0) &= F_{p1}u(s) = \frac{F_{p1}}{s} \end{aligned} \tag{12.7a,b}$$

Solving for $x_1(s)$:

$$sx_1(s) - x_1(0) = x_2(s)$$

$$s\left[sx_1(s) - x_1(0)\right] - x_2(0) = F_{p1}u(s) = \frac{F_{p1}}{s}$$

$$s^2 x_1(s) = \frac{F_{p1}}{s} + sx_1(0) + x_2(0) \qquad (12.8a\text{-}f)$$

$$x_1(s) = \frac{F_{p1}}{s^3} + \frac{sx_1(0)}{s^2} + \frac{x_2(0)}{s^2}$$

$$x_1(s) = \frac{-\sqrt{3}}{3s^3\sqrt{m}} + \frac{0}{s} - \frac{\sqrt{3m}}{3s^2}$$

The three terms on the right-hand side of (12.8f) represent the displacement of the first mode of vibration due to the force, initial displacement and initial velocity, respectively. This equation for $x_1(s)$ is the same as for $z_{p1}(s)$ in (9.17). Using the same back-transformation yields the identical result for the principal displacement as for $z_{p1}(t)$ in (9.20).

$$\begin{aligned} x_1(t) &= \frac{-t^2}{2\sqrt{3m}} + 0 - \frac{\sqrt{3m}\ t}{3} \\ &= \frac{-t^2}{2\sqrt{3}} + 0 - \frac{\sqrt{3}\ t}{3} \end{aligned} \qquad (12.9)$$

The two sets of equations for modes 2 and 3 can be solved for $x_3(t)$ and $x_5(t)$ in a similar fashion, again giving results which are the same as for $z_{p2}(t)$ and $z_{p3}(t)$ in (9.27) and (9.34). The three velocity states in principal coordinates can be defined by differentiating the displacement states.

Summarizing the solution in principal state space coordinates:

$$
\mathbf{x}(t) = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} -\frac{t^2}{2\sqrt{3}} - \frac{\sqrt{3}\ t}{3} \\ -\frac{t}{\sqrt{3}} - \frac{\sqrt{3}}{3} \\ \frac{3\sqrt{2}}{2} - \frac{3\sqrt{2}}{2}\cos t - \frac{\sqrt{2}}{2}\cos t + \frac{\sqrt{2}}{2}\sin t \\ \frac{3\sqrt{2}}{2}\sin t + \frac{\sqrt{2}}{2}\sin t + \frac{\sqrt{2}}{2}\cos t \\ -\frac{\sqrt{6}}{18} + \frac{\sqrt{6}}{18}\cos\sqrt{3}\,t + \frac{\sqrt{6}}{2}\cos\sqrt{3}\,t - \frac{7}{\sqrt{3}}\sin\sqrt{3}\,t \\ -\frac{\sqrt{6}\sqrt{3}}{18}\sin\sqrt{3}\,t + \frac{\sqrt{6}\sqrt{3}}{2}\sin\sqrt{3}\,t - \frac{7\sqrt{3}}{\sqrt{3}}\cos\sqrt{3}\,t \end{bmatrix} \qquad (12.10a\text{-}f)
$$

Let us assume that we are interested in three displacements and three velocities; the output matrix is shown below in (12.11), repeated from (10.38):

$$
\mathbf{C} = \begin{bmatrix} x_{n11} & 0 & x_{n12} & 0 & x_{n13} & 0 \\ 0 & x_{n11} & 0 & x_{n12} & 0 & x_{n13} \\ x_{n21} & 0 & x_{n22} & 0 & x_{n23} & 0 \\ 0 & x_{n21} & 0 & x_{n22} & 0 & x_{n23} \\ x_{n31} & 0 & x_{n32} & 0 & x_{n33} & 0 \\ 0 & x_{n31} & 0 & x_{n32} & 0 & x_{n33} \end{bmatrix} \qquad (12.11)
$$

$$\begin{bmatrix} z_1 \\ \dot{z}_1 \\ z_2 \\ \dot{z}_2 \\ z_3 \\ \dot{z}_3 \end{bmatrix} = \mathbf{Cx} = \begin{bmatrix} x_{n11} & 0 & x_{n12} & 0 & x_{n13} & 0 \\ 0 & x_{n11} & 0 & x_{n12} & 0 & x_{n13} \\ x_{n21} & 0 & x_{n22} & 0 & x_{n23} & 0 \\ 0 & x_{n21} & 0 & x_{n22} & 0 & x_{n23} \\ x_{n31} & 0 & x_{n32} & 0 & x_{n33} & 0 \\ 0 & x_{n31} & 0 & x_{n32} & 0 & x_{n33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{6}} & 0 \\ 0 & \frac{1}{\sqrt{3}} & 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & 0 & 0 & 0 & \frac{-2}{\sqrt{6}} & 0 \\ 0 & \frac{1}{\sqrt{3}} & 0 & 0 & 0 & \frac{-2}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & 0 & \frac{-1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{6}} & 0 \\ 0 & \frac{1}{\sqrt{3}} & 0 & \frac{-1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{6}} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} \tag{12.12}$$

With (12.12) we have the complete time domain results in physical coordinates.

12.4 MATLAB Code tdofss_modal_time_ode45.m – Time Domain Modal Contributions

12.4.1 Modal State Space Model Setup, Code Listing

This first section executes **tdofss_eig.m** to calculate the eigenvalues and eigenvectors. It then sets up the 6x6 system matrix and defines three individual mode 2x2 submatrices.

The force vector in physical coordinates is defined, applying step forces as defined in Figure 12.1. It is transformed to a forcing function in principal coordinates and expanded to 6x1 size by padding with zeros. To specify the input matrices for each of the three modes, three 2x1 submatrices are defined.

The output matrix is setup as a 3x6 matrix, to calculate displacements. Once again, three submatrices of 3x2 size are defined for the individual modes.

```
%	tdofss_modal_time_ode45.m	state space modal form transfer function analysis
%	 of tdof model, proportional damping, modal contribution plotting

	clf;

%	run tdofss_eig.m to provide eigenvalues and eigenvectors

	tdofss_eig;

	global  a_ss a1_ss a2_ss a3_ss b b1 b2 b3 u

%	note, this is the point where we would start if we had eigenvalue results from ANSYS,
%	using the eigenvalues and eigenvectors to define state space equations in
%	principal coordinates

%	define damping ratio to be used for proportional damping in the state space equation
%	in principal coordinates

	zeta = input('input zeta, 0.02 = 2% of critical damping (default) ...  ');

	if (isempty(zeta))
	zeta = 0.02;
	else
	end

%	setup 6x6 state-space system matrix for all three modes in principal
%	coordinates, a_ss

	a_ss = [    0        1        0          0            0          0
	            0        0        0          0            0          0
	            0        0        0          1            0          0
	            0        0     -w2^2    -2*zeta*w2        0          0
	            0        0        0          0            0          1
	            0        0        0          0          -w3^2    -2*zeta*w3];

%	setup three 2x2 state-space matrices, one for each individual mode

	a1_ss = a_ss(1:2,1:2);

	a2_ss = a_ss(3:4,3:4);

	a3_ss = a_ss(5:6,5:6);

%	transform the 3x1 force vector in physical coordinates to principal coordinates and
%	then insert the principal forces in the appropriate rows in the state-space
%	6x1 input matrix, padding with zeros as appropriate

	F = [1 0 -2]';

	Fp = xn'*F;

%	expand the force vectors in principal coordinates from 3x1 to 6x1, padding with zeros

	b = [0 Fp(1) 0 Fp(2) 0 Fp(3)]';     %  principal forces applied to all masses
```

```
            b1 = b(1:2);

            b2 = b(3:4);

            b3 = b(5:6);

%           the output matrix c is setup in one step, to allow the "bode" command to
%           output the desired physical coordinates directly without having to go
%           through any intermediate steps.

%           setup the output matrix for displacement transfer functions, each row
%           represents the position outputs of mass 1, mass 2 and mass 3
%           velocities not included, so c is only 3x6 instead of 6x6

            c = [xn(1,1)   0   xn(1,2)   0   xn(1,3)   0
                 xn(2,1)   0   xn(2,2)   0   xn(2,3)   0
                 xn(3,1)   0   xn(3,2)   0   xn(3,3)   0];

            c1 = c(:,1:2);

            c2 = c(:,3:4);

            c3 = c(:,5:6);

%           define direct transmission matrix d

            d = 0;
```

12.4.2 Problem Setup, Initial Conditions, Code Listing

Now that the model is in place, we can solve for transient response. The input scalar, “u” is set to “1,” for a unity step function. The total time is set and a vector of time span from 0 to 10 seconds (default) is setup for input to the ode routine.

The two 3x1 initial condition displacement and velocity vectors with initial displacements and velocities from Figure 12.1 are set up, then transformed to principal coordinates. Next the 6x1 initial condition vector is constructed from appropriate elements of the two 3x1 vectors. We are now ready to solve the problem.

```
%           transient response using the ode45 command

            u = 1;

            ttotal = input('Input total time for Simulation, default = 10 sec, ...  ');

            if (isempty(ttotal))
            ttotal = 10;
            else
```

```
        end

        tspan = [0 ttotal];

%       calculate the initial conditions in principal coordinates using the inverse of the
%       normalized modal matrix

        x0phys = [0 -1 1]';             %       initial condition position, physical coord

        x0dphys = [-1 2 -2]';           %       initial condition velocity, physical coord

        x0 = inv(xn)*x0phys;

        x0d = inv(xn)*x0dphys;

%       create the initial condition state vector

        x0ss = [x0(1) x0d(1) x0(2) x0d(2) x0(3) x0d(3)];

        x0ss1 = x0ss(1:2);

        x0ss2 = x0ss(3:4);

        x0ss3 = x0ss(5:6);
```

12.4.3 Solving Equations Using ode45, Code Listing

The ode45 "options" parameter, which can be used to control many options for use in the solution, is set to a null vector.

Next, the total response in principal coordinates and the three individual mode responses in principal coordinates are calculated using MATLAB's ode45 differential equation solver. Four functions, listed separately in the following sections, are used by ode45 to define the equations to solve.

The responses in principal coordinates are then transformed to physical coordinates.

```
%       use the ode45 non-stiff differential equation solver

        options = [ ];                                  %  no options specified

%       total response, principal coord, states are modes of vibration

        [t,x] = ode45('tdofssmodalfun',tspan,x0ss,options);

%  mode 1 response, principal coord

        [t1,x1] = ode45('tdofssmodal1fun',tspan,x0ss1,options);

%  mode 2 response, principal coord
```

```
        [t2,x2] = ode45('tdofssmodal2fun',tspan,x0ss2,options);

%  mode 3 response, principal coord

        [t3,x3] = ode45('tdofssmodal3fun',tspan,x0ss3,options);

%  total response, physical coord

        z_ode = c*x';

%  mode 1 response, physical coord

        z_ode1 = c1*x1';

%  mode 2 response, physical coord

        z_ode2 = c2*x2';

%  mode 3 response, physical coord

        z_ode3 = c3*x3';
```

12.4.4 Plotting, Code Listing

```
%       plot displacements in principal coordinates

        subplot(1,1,1);

        plot(t1,x1(:,1),'k+-',t2,x2(:,1),'kx-',t3,x3(:,1),'k-')
        title('Displacements in Principal Coordinate System, ode45')
        xlabel('Time, sec')
        ylabel('Displacements')
        legend('zp1','zp2','zp3',2)
        grid

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

        axis([0 1 -2 2]);

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

%       plot displacements in physical coordinates

        plot(t,z_ode(1,:),'k+-',t,z_ode(2,:),'kx-',t,z_ode(3,:),'k-')
        title('Displacements in Physical Coordinate System, ode45')
        xlabel('Time, sec')
        ylabel('Displacements')
        legend('z1','z2','z3',3)
        grid

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

%       load previous closed-form solutions for tplot, z1, z2, z3 if zeta = 0
```

```
        if  zeta == 0

        load tdof_modal_time_z1z2z3;

        plot(t,z_ode(1,:),'k-',t,z_ode(2,:),'k-',t,z_ode(3,:),'k-',tplot,z1,'k.-',tplot,z2, …
                'k.-',tplot,z3,'k.-')
        title('Displacements in Physical Coordinate System from ode45 (ode) …
                 and Closed Form (cf)')
        xlabel('Time, sec')
        ylabel('Vibration Displacements')
        legend('ode dof 1','ode dof 2','ode dof 3','cf dof 1','cf dof 2','cf dof 3')
        grid

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

        else
        end

%       plot the modal contributions to the motion of masses 1, 2 and 3

        plot(t1,z_ode1(1,:),'k+-',t2,z_ode2(1,:),'kx-',t3,z_ode3(1,:),'k-')
        title('Displacement of dof 1 for Modes 1, 2 and 3, ode45')
        xlabel('Time, sec')
        ylabel('Displacements')
        legend('Mode 1','Mode 2','Mode 3')
        grid

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

        plot(t1,z_ode1(2,:),'k+-',t2,z_ode2(2,:),'kx-',t3,z_ode3(2,:),'k-')
        title('Displacement of dof 2 for Modes 1, 2 and 3, ode45')
        xlabel('Time, sec')
        ylabel('Displacements')
        legend('Mode 1','Mode 2','Mode 3')
        grid

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

        plot(t1,z_ode1(3,:),'k+-',t2,z_ode2(3,:),'kx-',t3,z_ode3(3,:),'k-')
        title('Displacement of dof 3 for Modes 1, 2 and 3, ode45')
        xlabel('Time, sec')
        ylabel('Displacements')
        legend('Mode 1','Mode 2','Mode 3')
        grid
```

12.4.5 Functions Called: tdofssmodalfun.m, tdofssmodal1fun.m, tdofssmodal2fun.m, tdofssmodal3fun.m

The ode45 differential equation solver calls function files depending on which solution is being performed. The four functions for calculating the system response as well as individual responses of modes 1, 2 and 3 are listed below. Each simply defines the state equation where the derivative of the state vector is equal to the system matrix times the states plus the input matrix times the input: $\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$. The "global" assignments make all the variables defined available both to the calling program and to the function.

System response:

```
        function xprime = tdofssmodalfun(t,x)

%       function for calculating the transient response of tdof_ss_modal_time_ode45.m

        global  a_ss a1_ss a2_ss a3_ss b b1 b2 b3 u

        xprime = a_ss*x + b*u;
```

Mode 1 response:

```
        function xprime = tdofssmodal1fun(t1,x1)

%       function for calculating the transient response of tdof_ss_modal_time_ode45.m

        global  a_ss a1_ss a2_ss a3_ss b b1 b2 b3 u

        xprime = a1_ss*x1 + b1*u;
```

Mode 2 response:

```
        function xprime = tdofssmodal2fun(t2,x2)

%       function for calculating the transient response of tdof_ss_modal_time_ode45.m

        global  a_ss a1_ss a2_ss a3_ss b b1 b2 b3 u

        xprime = a2_ss*x2 + b2*u;
```

Mode 3 response:

```
        function xprime = tdofssmodal3fun(t3,x3)

%       function for calculating the transient response of tdof_ss_modal_time_ode45.m

        global  a_ss a1_ss a2_ss a3_ss b b1 b2 b3 u

        xprime = a3_ss*x3 + b1*u;
```

12.5 Plotted Results

The following figures should be compared with Figures 9.2 through 9.7, which were plotted using the closed form modal solutions.

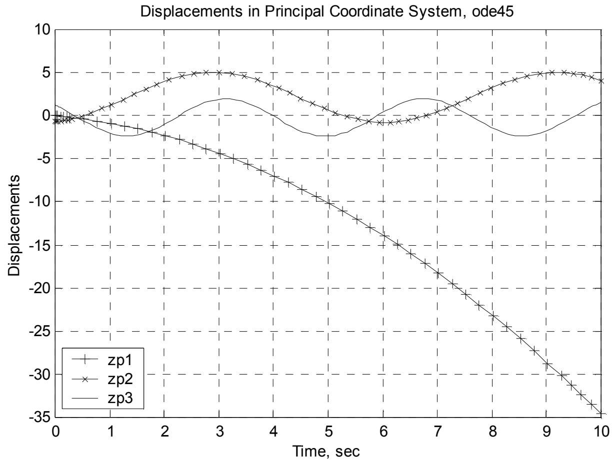

Figure 12.2: Displacements in principal coordinate system using ode45.

The motions of the rigid body and two oscillatory modes are clearly seen.

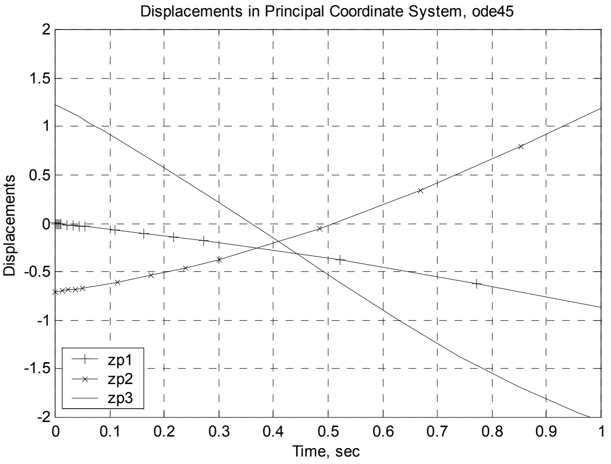

Figure 12.3: Displacements in principal coordinate system, expanded scales to see initial conditions.

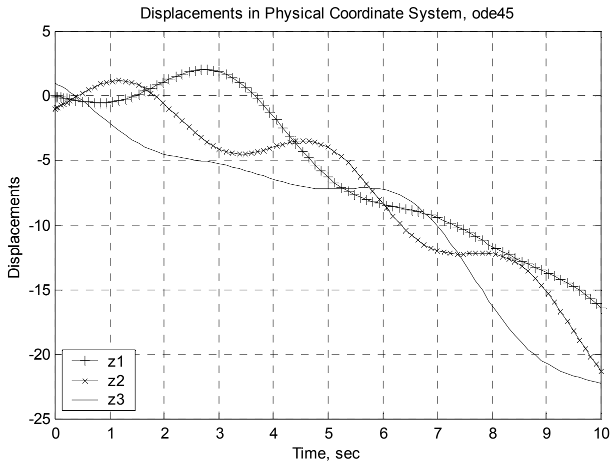

Figure 12.4: Displacements in physical coordinate system.

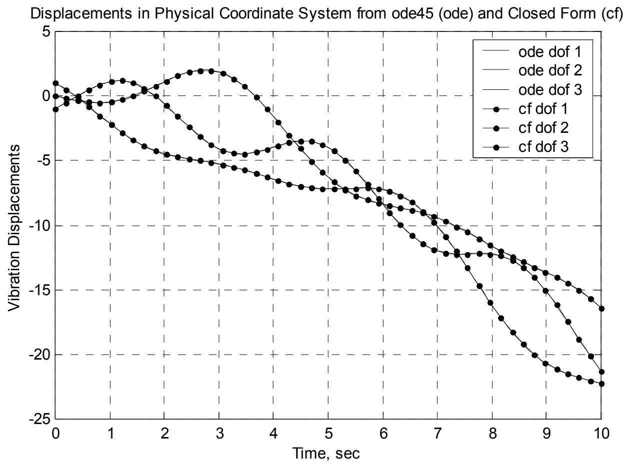

Figure 12.5: Displacements in physical coordinate system – comparing closed form solution from Chapter 7.

The three plots below show how one can study the motions of degrees of freedom due to individual modes. Use zeta = 0 when running **tdofss_modal_time_ode45.m** in order to plot the closed form solution.

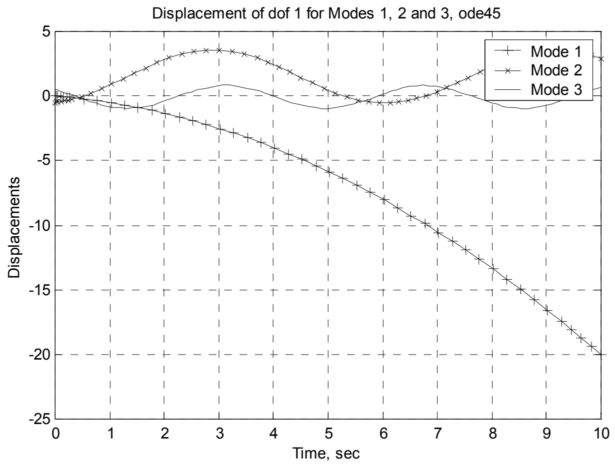

Figure 12.6: Displacement of mass 1 for modes 1, 2 and 3.

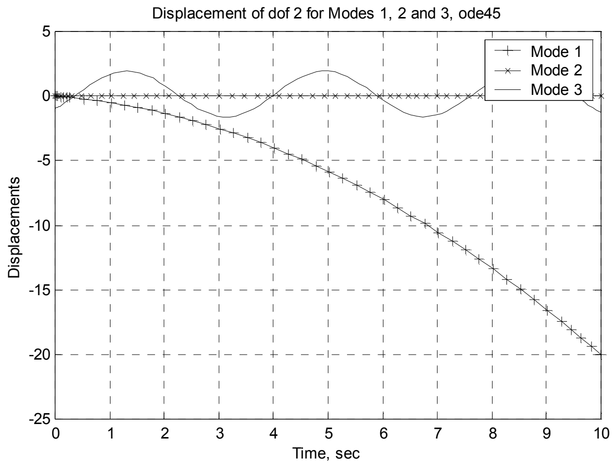

Figure 12.7: Displacement of mass 2 for modes 1, 2 and 3.

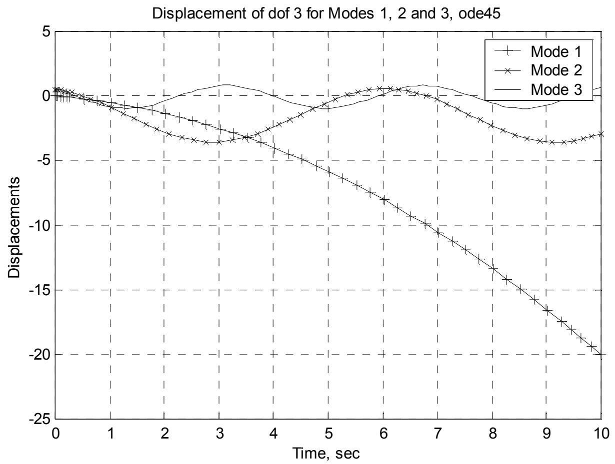

Figure 12.8: Displacement of mass 3 for modes 1, 2 and 3.

Problem

P12.1 Using the initial conditions and forcing functions from P7.4, solve for the time domain response of the states in principal coordinates in closed form using Laplace transforms. Define the output matrix if the outputs required are the displacements of both masses.