AE 5311 - Homework 7

James Grisham

4/2/2013

Problem 1

Problem Statement

A tapered rigid bar with a length L is supported by 2 springs with stiffnesses k_L and k_R at the left and right ends, respectively. The bar has a solid circular cross-section with a radius that varies from r_1 on the left side to r_2 on the right side. Find the natural frequencies and the associated mode shapes using the following data: $L = 4$ m, $r_1 = 2$ cm, $r_2 = 5$ cm, mass density: 2700 kg/m^3, $k_L = 10000$ N/m, and $k_R = 20000$ N/m. Specifically,

(**a**). Derive EOM by Newton's Law, use any 2-dof of your choice.

(**b**). Find natural frequencies and associated mode shapes.

(**c**). Sketch the mode shapes.

Solution

Part A

The center of mass of the bar must be determined. The variation in radius is given by

$$r(x) = \frac{r_2 - r_1}{L}\,x + r_1 \tag{1}$$

where $0 \leq x \leq L$.

The volume of the tapered bar is given by

$$V = \pi \int_0^L [r(x)]^2\,dx$$

Therefore, the mass of the bar is given by

$$m = \pi\rho \int_0^L [r(x)]^2\,dx = \pi\rho \int_0^L \left[\frac{r_2 - r_1}{L}\,x + r_1\right]^2 dx = \pi\rho \int_0^L \left[\frac{r_2^2 - 2r_1 r_2 + r_1^2}{L^2}\,x^2 + 2\,r_1\,\frac{r_2 - r_1}{L}\,x + r_1^2\right] dx$$

Evaluating the integral yields

$$m = \pi\rho \left[\frac{r_2^2 - 2r_1 r_2 + r_1^2}{L^2}\,\frac{x^3}{3} + r_1\,\frac{r_2 - r_1}{L}\,x^2 + r_1^2\,x\right]_0^L$$

Evaluating the upper and lower limits and simplifying, the mass of the bar is given by

$$m = \frac{\pi \rho L}{3}({r_2}^2 + r_1\, r_2 + {r_1}^2)$$

This result is easily verified by looking up the formula for the volume of a frustum of a right circular cone. The x-coordinate of the center of mass is given by

$$\overline{x} = \frac{\displaystyle\int_0^L x\, m(x)\, dx}{\displaystyle\int_0^L m(x)\, dx}$$

Evaluating the integrals and simplifying,

$$\overline{x} = \frac{L}{4}\,\frac{\left({r_1}^2 + 2\, r_1\, r_2 + 3\, {r_2}^2\right)}{({r_1}^2 + r_1\, r_2 + {r_2}^2)}$$

There are multiple choices for the degrees of freedom. In this case, the vertical displacements of the ends are chosen as the two degrees of freedom.

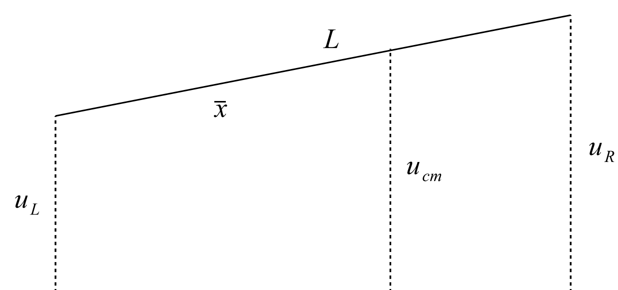

Figure 1. Geometry.

From the above geometry,

$$\sin\theta = \frac{u_R - u_L}{L}$$

using the small-angle approximation, the equation becomes

$$\theta = \frac{u_R - u_L}{L} \tag{2}$$

Also, from the geometry,

$$\sin\theta = \frac{u_{cm} - u_L}{\overline{x}} = \frac{u_R - u_L}{L}$$

$$\therefore$$

$$u_{cm} = \frac{\overline{x}}{L}(u_R - u_L) + u_L \tag{3}$$

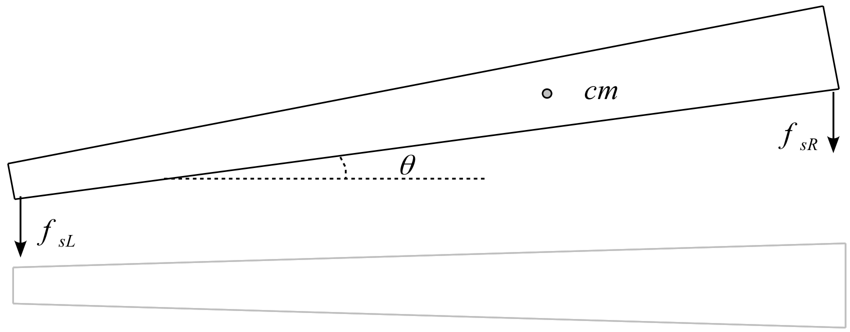

Figure 2. Free body diagram.

Summing the forces in the y-direction,

$$+\uparrow \sum F_y = -f_{sL} - f_{sR} = m\ddot{u}_{cm} \tag{4}$$

Now, summing the moments about the center of mass,

$$+\,\text{CCW} \sum M_{cm} = f_{sL}\overline{x}\cos\theta - f_{sR}(L-\overline{x})\cos\theta = I_{cm}\ddot{\theta} \tag{5}$$

The forces in the springs are

$$\begin{aligned} f_{sL} &= k_L u_L \\ f_{sR} &= k_R u_R \end{aligned} \tag{6}$$

Using the small-angle approximation, $\cos\theta \approx 1$, and remembering Eq. 2, Eq. 5 becomes

$$\boxed{-\frac{I_{cm}}{L}\ddot{u}_L + \frac{I_{cm}}{L}\ddot{u}_R - \overline{x}k_L u_L + (L-\overline{x})k_R u_R = 0} \tag{7}$$

Similarly,

$$\boxed{m\left(1-\frac{\overline{x}}{L}\right)\ddot{u}_L + m\frac{\overline{x}}{L}\ddot{u}_R + k_L u_L + k_R u_R = 0} \tag{8}$$

In matrix form,

$$\boxed{\overbrace{\begin{bmatrix} -I_{cm}/L & I_{cm}/L \\ m\,(1-\overline{x}/L) & m\overline{x}/L \end{bmatrix}}^{\mathbf{M}} \begin{Bmatrix} \ddot{u}_L \\ \ddot{u}_R \end{Bmatrix} + \overbrace{\begin{bmatrix} -\overline{x}k_L & (L-\overline{x})k_R \\ k_L & k_R \end{bmatrix}}^{\mathbf{K}} \begin{Bmatrix} u_L \\ u_R \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}} \tag{9}$$

The only unknown quantity in the mass and stiffness matrices is the moment of inertia. To determine the eigenvalues and eigenvectors, the moment of inertia about an axis that is perpendicular to the page that passes through the center of mass must be determined.

For a circular cone of radius a and height h, the moment of inertia about the axis through the center of mass and perpendicular to the axis of the cone is given by

$$I = \int x^2\,dm$$

where

$$dm = \rho \, dV$$

An expression for the radius variation is given by Eq. 1. Considering disks of area $\pi[r(x)]^2$ and thickness dx,

$$dV = \pi[r(x)]^2 \, dx$$

$$\therefore$$

$$I = \pi\rho \int_0^L x^2 \left[r(x)\right]^2 \, dx$$

Now, the moment of inertia about the center of mass will be determined using the parallel axis theorem.

$$I_{cm} = I - m\overline{x}^2 = 48.7 \text{ kg}\cdot\text{m}^2$$

Part B

$$|\mathbf{K} - \lambda\mathbf{M}| = 0$$

Solving the eigenvalue problem using MATLAB,

$$\lambda_1 = 675.6 \quad \text{and} \quad \lambda_2 = 2203$$

Therefore,

$$\boxed{\omega_1 = 25.99 \text{ rad/s} \quad \text{and} \quad \omega_2 = 46.94 \text{ rad/s}} \tag{10}$$

$$\boxed{\phi_1 = \begin{Bmatrix} 1.0000 \\ 0.8168 \end{Bmatrix} \quad \text{and} \quad \phi_2 = \begin{Bmatrix} 1.0000 \\ -0.6121 \end{Bmatrix}} \tag{11}$$

Part C

Sketching the mode shapes using the eigenvectors from Eq. 11,

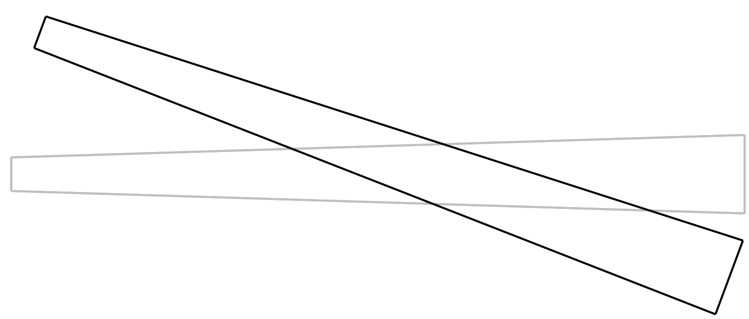

Figure 3. Mode 1.

Figure 4. Mode 2.

Problem 2

Problem Statement

Solve Problem 1 using the **M** and **K** matrices given in Note-17A. Show that the eigenvalues are the same as those obtained in Problem 1. The eigenvectors may be different (they depend on the coordinates chosen), but with proper scaling, show that the mode shapes are the same as those obtained in Problem 1.

Solution

The **M** and **K** matrices given in Note-17A are

$$\mathbf{M} = \begin{bmatrix} (mb^2 + I_0)/(a+b)^2 & -(I_0 - abm)/(a+b)^2 \\ -(I_0 - abm)/(a+b)^2 & (ma^2 + I_0)/(a+b)^2 \end{bmatrix}$$

and

$$\mathbf{K} = \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix}$$

Applying these matrices to problem 1, $b = L - \overline{x}$, $a = \overline{x}$, $k_1 = k_L$, $k_2 = k_R$, and $I_0 = I_{cm}$. Using MATLAB to determine eigenvalues and eigenvectors,

$$\lambda_1 = 675.6 \quad \text{and} \quad \lambda_2 = 2203$$

Therefore,

$$\boxed{\omega_1 = 25.99 \text{ rad/s} \quad \text{and} \quad \omega_2 = 46.94 \text{ rad/s}}$$

These numbers match the result from Problem 1 perfectly (see Eq. 10). The eigenvectors are

$$\phi_1 = \begin{Bmatrix} -0.1701 \\ -0.1390 \end{Bmatrix} \quad \text{and} \quad \phi_2 = \begin{Bmatrix} -0.3549 \\ 0.2172 \end{Bmatrix}$$

Scaling the first eigenvector,

$$\phi_1 = \frac{1}{-0.1701} \begin{Bmatrix} -0.1701 \\ -0.1390 \end{Bmatrix} = \begin{Bmatrix} 1.0000 \\ 0.8168 \end{Bmatrix}$$

Doing the same for the second eigenvector,

$$\phi_2 = \frac{1}{-0.3549} \begin{Bmatrix} -0.3549 \\ 0.2172 \end{Bmatrix} = \begin{Bmatrix} 1.0000 \\ -0.6121 \end{Bmatrix}$$

After being scaled, the eigenvectors are

$$\boxed{\phi_1 = \begin{Bmatrix} 1.0000 \\ 0.8168 \end{Bmatrix} \quad \text{and} \quad \phi_2 = \begin{Bmatrix} 1.0000 \\ -0.6121 \end{Bmatrix}} \tag{12}$$

These eigenvectors are exactly the same as the eigenvectors from Problem 1 (see Eq. 11). Sketching these mode shapes yields drawings identical to those in Figure 3 and Figure 4. The MATLAB code used for calculations can be seen in the appendix.

Problem 3

Problem Statement

A uniform rigid bar with a length L and mass m is supported by 2 springs k_L and k_R at the left and right ends, respectively. The bar carries 2 spring-mass oscillators: (k_a, m_a) at $x = a$ and (k_b, m_b) at $x = b$ (measured from the left end). Derive the EOM by Newton's law.

Solution

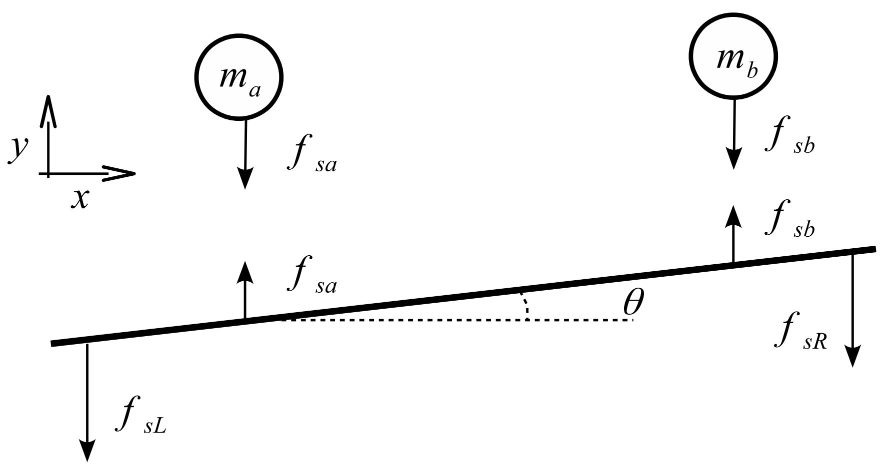

Figure 5. Free body diagram.

Summing the forces in the y-direction for the bar,

$$+\uparrow \sum F_y = -f_{sL} - f_{sR} + f_{sa} + f_{sb} = m\ddot{u}_c$$

$$m\ddot{u}_c + f_{sL} + f_{sR} - f_{sa} - f_{sb} = 0 \tag{13}$$

Where the spring forces are given by

$$\begin{aligned} f_{sL} = k_L \Delta_L \quad \text{and} \quad f_{sR} = k_R \Delta_R \\ f_{sa} = k_a \Delta_a \quad \text{and} \quad f_{sb} = k_b \Delta_b \end{aligned} \tag{14}$$

Looking at the geometry, letting $L/2 = \ell$, and using the small angle approximation (i.e., $\sin\theta \approx \theta$),

$$\Delta_L = u_c - \ell\theta \tag{15}$$

$$\Delta_R = u_c + \ell\theta \tag{16}$$

$$\Delta_a = u_a - u_1$$

where u_a is the absolute displacement of m_a and u_1 is the point at which the spring is connected. The coordinates of the connection point are found by examining the geometry,

$$u_1 = u_c - (\ell - a)\theta$$

$$\therefore$$

$$\Delta_a = u_a - u_c + \ell\theta - a\theta \tag{17}$$

Similarly for m_b,

$$\Delta_b = u_b - u_2$$

$$u_2 = u_c + (b - \ell)\theta$$

$$\Delta_b = u_b - u_c + \ell\theta - b\theta \tag{18}$$

Inserting (15), (16), (17), and (18) into (14) and then inserting that result into (13) and simplifying yields

$$\boxed{\begin{aligned} m\ddot{u}_c + [k_L + k_R + k_a + k_b]\, u_c + [\ell(-k_L + k_R - k_a - k_b) + ak_a + bk_b]\,\theta \\ -k_a u_a - k_b u_b = 0 \end{aligned}} \tag{19}$$

Now, the second EOM will be determined by summing the moments about the center of mass,

$$+\text{CCW} \sum M_c = f_{sL}\ell\cos\theta - f_{sa}(\ell - a)\cos\theta + f_{sb}(b - \ell)\cos\theta - f_{sR}\ell\cos\theta = I\ddot{\theta}$$

Applying the small angle approximation and rearranging, the EOM becomes

$$I\ddot{\theta} - f_{sL}\ell + f_{sR}\ell + f_{sa}(\ell - a) - f_{sb}(b - \ell) = 0 \tag{20}$$

Inserting the expressions for spring forces,

$$I\ddot{\theta} - k_L\Delta_L\ell + k_R\Delta_R\ell + k_a\Delta_a(\ell - a) - k_b\Delta_b(b - \ell) = 0 \tag{21}$$

Again, inserting (15), (16), (17), and (18) into (21) and simplifying,

$$\boxed{\begin{aligned} I\ddot{\theta} + [\ell(-k_L + k_R - k_a - k_b) + ak_a + bk_b]\, u_c \\ + \left[\ell^2(k_L + k_R + k_a + k_b) - 2a\ell k_a - 2b\ell k_b + a^2 k_a + b^2 k_b\right]\theta \\ + [k_a(\ell - a)]\, u_a + [k_b(\ell - b)]\, u_b \end{aligned}} \tag{22}$$

The third and fourth EOMs are determined by summing the forces in the y-direction for masses a and b,

$$+\uparrow \sum F_y = f_{sa} = m_a \ddot{u}_a$$

$$\boxed{m_a \ddot{u}_a - k_a u_c + k_a(\ell - a)\theta + k_a u_a = 0} \tag{23}$$

Similarly,

$$+\uparrow \sum F_y = f_{sb} = m_b \ddot{u}_b$$

$$\boxed{m_b \ddot{u}_b - k_b u_c + k_b(\ell - b)\theta + k_b u_b = 0} \tag{24}$$

Problem 4

Problem Statement

Verify EOM derived in Problem 3 by re-deriving the EOM using Lagrange's equations. (You may use the general approach given in Note-17A).

Solution

To derive the EOMs using Lagrange's equations, the general approach given in Note-17A will be employed. Assigning generalized coordinates:

$$\begin{aligned} v_1 &= u_c \quad \text{(translation of the center of mass)} \\ v_2 &= \theta \quad \text{(rotation of rigid bar)} \\ v_3 &= u_a \quad \text{(absolute displacement of } m_a) \\ v_4 &= u_b \quad \text{(absolute displacement of } m_b) \end{aligned}$$

The kinetic energy of the system is

$$\mathcal{T} = \frac{1}{2} m \dot{v}_1^2 + \frac{1}{2} I \dot{v}_2^2 + \frac{1}{2} m_a \dot{v}_3^2 + \frac{1}{2} m_b \dot{v}_4^2 \tag{25}$$

The potential energy of the system is

$$\mathcal{V} = \frac{1}{2} k_L {\Delta_1}^2 + \frac{1}{2} k_R {\Delta_2}^2 + \frac{1}{2} k_a {\Delta_3}^2 + \frac{1}{2} k_b {\Delta_4}^2 \tag{26}$$

Now, the displacements can be written as

$$v_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \{\mathbf{v}\} = \{\mathbf{Tm}_1\}\{\mathbf{v}\}$$

$$v_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} \{\mathbf{v}\} = \{\mathbf{Tm}_2\}\{\mathbf{v}\}$$

$$v_3 = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \{\mathbf{v}\} = \{\mathbf{Tm}_3\}\{\mathbf{v}\}$$

$$v_4 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \{\mathbf{v}\} = \{\mathbf{Tm}_4\}\{\mathbf{v}\}$$

Now,

$$\begin{aligned}
\dot{v}_1^2 &= \dot{v}_1^\top \dot{v}_1 \\
&= (\mathbf{Tm}_1\dot{\mathbf{v}})^\top(\mathbf{Tm}_1\dot{\mathbf{v}}) \\
&= \dot{\mathbf{v}}^\top\mathbf{Tm}_1{}^\top\mathbf{Tm}_1\dot{\mathbf{v}} \\
&= \dot{\mathbf{v}}^\top(\mathbf{Tm}_1{}^\top\mathbf{Tm}_1)\dot{\mathbf{v}}
\end{aligned}$$

Therefore, the kinetic energy can be written as

$$\begin{aligned}
\mathcal{T} &= \frac{1}{2}m\dot{\mathbf{v}}^\top(\mathbf{Tm}_1{}^\top\mathbf{Tm}_1)\dot{\mathbf{v}} + \frac{1}{2}I\dot{\mathbf{v}}^\top(\mathbf{Tm}_2{}^\top\mathbf{Tm}_2)\dot{\mathbf{v}} \\
&\quad + \frac{1}{2}m_a\dot{\mathbf{v}}^\top(\mathbf{Tm}_3{}^\top\mathbf{Tm}_3)\dot{\mathbf{v}} + \frac{1}{2}m_b\dot{\mathbf{v}}^\top(\mathbf{Tm}_4{}^\top\mathbf{Tm}_4)\dot{\mathbf{v}}
\end{aligned}$$

where

$$\dot{\mathbf{v}} = \begin{Bmatrix} \dot{v}_1 \\ \dot{v}_2 \\ \dot{v}_3 \\ \dot{v}_4 \end{Bmatrix}$$

Now, turning to the potential energy, (15) - (18) can be written in terms of the generalized coordinates as

$$\begin{aligned}
\Delta_1 &= v_1 - \ell v_2 \\
\Delta_2 &= v_1 + \ell v_2 \\
\Delta_3 &= -v_1 + (\ell - a)v_2 + v_3 \\
\Delta_4 &= -v_1 + (\ell - b)v_2 + v_4
\end{aligned}$$

Using an approach similar to before,

$$\begin{aligned}
\Delta_1 &= \overbrace{\begin{bmatrix} 1 & -\ell & 0 & 0 \end{bmatrix}}^{\mathbf{Tk}_1}\mathbf{v} \\
\Delta_2 &= \begin{bmatrix} 1 & \ell & 0 & 0 \end{bmatrix}\mathbf{v} \\
\Delta_3 &= \begin{bmatrix} -1 & (\ell - a) & 1 & 0 \end{bmatrix}\mathbf{v} \\
\Delta_4 &= \begin{bmatrix} -1 & (\ell - b) & 0 & 1 \end{bmatrix}\mathbf{v}
\end{aligned}$$

$$\therefore$$

$$\Delta_j = \mathbf{Tk}_j\mathbf{v}$$

The mass matrix for this system is given by

$$\mathbf{M} = \sum_{i=1}^{N} \vec{m}_i(\mathbf{Tm}_i{}^\top\mathbf{Tm}_i)$$

where N is the number of degrees of freedom and $\vec{m} = \begin{bmatrix} m & I & m_a & m_b \end{bmatrix}$ so that $m_1 = m$, $m_2 = I$, and so on.

Similarly, the stiffness matrix of the system is given by

$$\mathbf{K} = \sum_{j=1}^{N} \vec{k}_j (\mathbf{T}\mathbf{k}_j{}^\top \mathbf{T}\mathbf{k}_j)$$

where $\vec{k} = \begin{bmatrix} k_L & k_R & k_a & k_b \end{bmatrix}$. The mass matrix is given by

$$\mathbf{M} = \begin{pmatrix} m & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & m_a & 0 \\ 0 & 0 & 0 & m_b \end{pmatrix}$$

Also, the stiffness matrix is

$$\mathbf{K} = \begin{pmatrix} k_L + k_R + k_a + k_b & k_R\,\ell - k_L\,\ell + k_a\,(a-\ell) + k_b\,(b-\ell) & -k_a & -k_b \\ k_R\,\ell - k_L\,\ell + k_a\,(a-\ell) + k_b\,(b-\ell) & k_L\,\ell^2 + k_R\,\ell^2 + k_a\,(a-\ell)^2 + k_b\,(b-\ell)^2 & -k_a\,(a-\ell) & -k_b\,(b-\ell) \\ -k_a & -k_a\,(a-\ell) & k_a & 0 \\ -k_b & -k_b\,(b-\ell) & 0 & k_b \end{pmatrix}$$

The terms in the mass and stiffness matrices are identical to the terms in the EOMs obtained in Problem 3. To verify the methods are equivalent, the numerical results were compared using MATLAB (see Appendix for code). The output from the code is provided below:

```
-----------------------------------------------------------------
Results from application of Newton's Law:
-----------------------------------------------------------------
Eigenvectors:
   -0.0458   -0.0543   -0.0113    0.1218
   -0.1095    0.0307    0.0386   -0.0239
   -0.0559    0.7557   -0.5985    0.2600
    0.1361    0.2092    0.3274    0.1747

lambda_1 = 439.7
lambda_2 = 916.6
lambda_3 = 1112
lambda_4 = 2141

omega_1 = 20.97 rad/s
omega_2 = 30.27 rad/s
omega_3 = 33.35 rad/s
omega_4 = 46.27 rad/s

f_1 = 3.337 Hz
f_2 = 4.818 Hz
f_3 = 5.308 Hz
f_4 = 7.365 Hz

-----------------------------------------------------------------
Results from application of Lagrange's Equations:
-----------------------------------------------------------------
Eigenvectors:
   -0.0458   -0.0543   -0.0113    0.1218
   -0.1095    0.0307    0.0386   -0.0239
   -0.0559    0.7557   -0.5985    0.2600
    0.1361    0.2092    0.3274    0.1747

lambda_1 = 439.7
lambda_2 = 916.6
lambda_3 = 1112
lambda_4 = 2141

omega_1 = 20.97 rad/s
omega_2 = 30.27 rad/s
omega_3 = 33.35 rad/s
omega_4 = 46.27 rad/s

f_1 = 3.337 Hz
f_2 = 4.818 Hz
f_3 = 5.308 Hz
f_4 = 7.365 Hz
```

By examining the output of the code, and by examining the mass and stiffness matrices, it is clear that the methods yield a result that is equivalent.

Problem 5

Problem Statement

For the system of Problem 3, find the natural frequencies and associated mode shapes. Also sketch the mode shapes. Use the following data:

$$L = 4 \text{ m}, \quad m = 50 \text{ kg}, \quad k_L = 10000 \text{ N/m}, \quad k_R = 20000 \text{ N/m}$$

$$k_a = 1000 \text{ N/m}, \quad m_a = 1 \text{ kg}, \quad k_b = 5000 \text{ N/m}, \quad m_b = 5 \text{ kg}, \quad a = 1 \text{ m}, \text{ and } \quad b = 3 \text{ m}$$

Solution

A MATLAB code was written to perform all of the necessary calculations (see Appendix for code). The natural frequencies are

```
omega_1 = 20.97 rad/s
omega_2 = 30.27 rad/s
omega_3 = 33.35 rad/s
omega_4 = 46.27 rad/s

f_1 = 3.337 Hz
f_2 = 4.818 Hz
f_3 = 5.308 Hz
f_4 = 7.365 Hz
```

The eigenvectors are

$$\Phi = \begin{bmatrix} -0.0458 & -0.0543 & -0.0113 & 0.1218 \\ -0.1095 & 0.0307 & 0.0386 & -0.0239 \\ -0.0559 & 0.7557 & -0.5985 & 0.2600 \\ 0.1361 & 0.2092 & 0.3274 & 0.1747 \end{bmatrix}$$

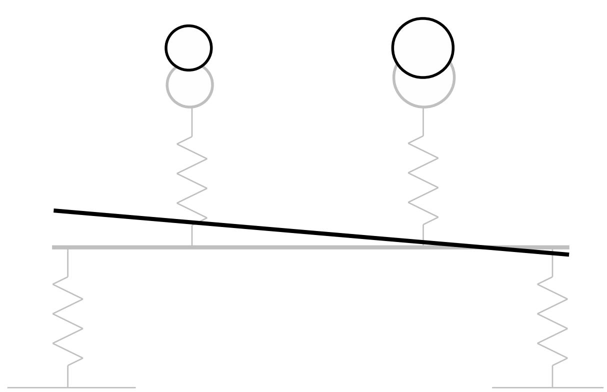

Figure 6. Mode 1.

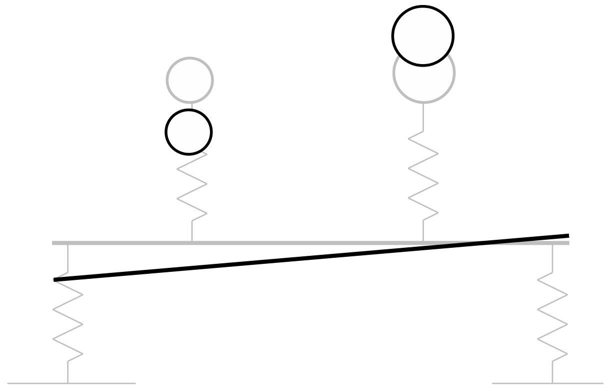

Figure 7. Mode 2.

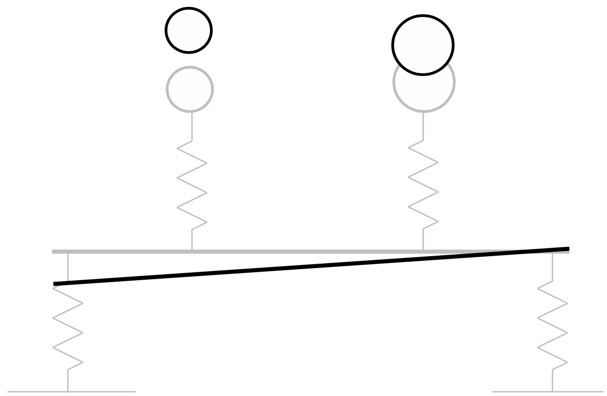

Figure 8. Mode 3.

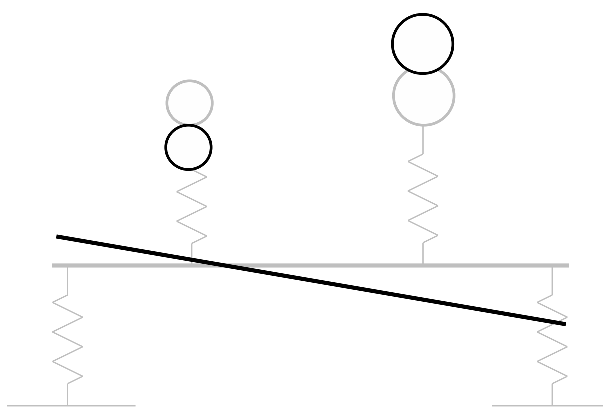

Figure 9. Mode 3.

Problem 6

Problem Statement

For the system of Problem 3, find the dynamics response due to the following initial conditions: $u_a(0) = 2$ cm, $\dot{u}_b(0) = -1$ m/s. All other IC's are zero. Solve the problem by long hand calculation or by MASDAN using the data given in Problem 5.

Solution

To solve the problem, the `ode45` function was used and validated using the `MDOFODE45` function that was distributed with MASDAN.

Results from `ode45`:

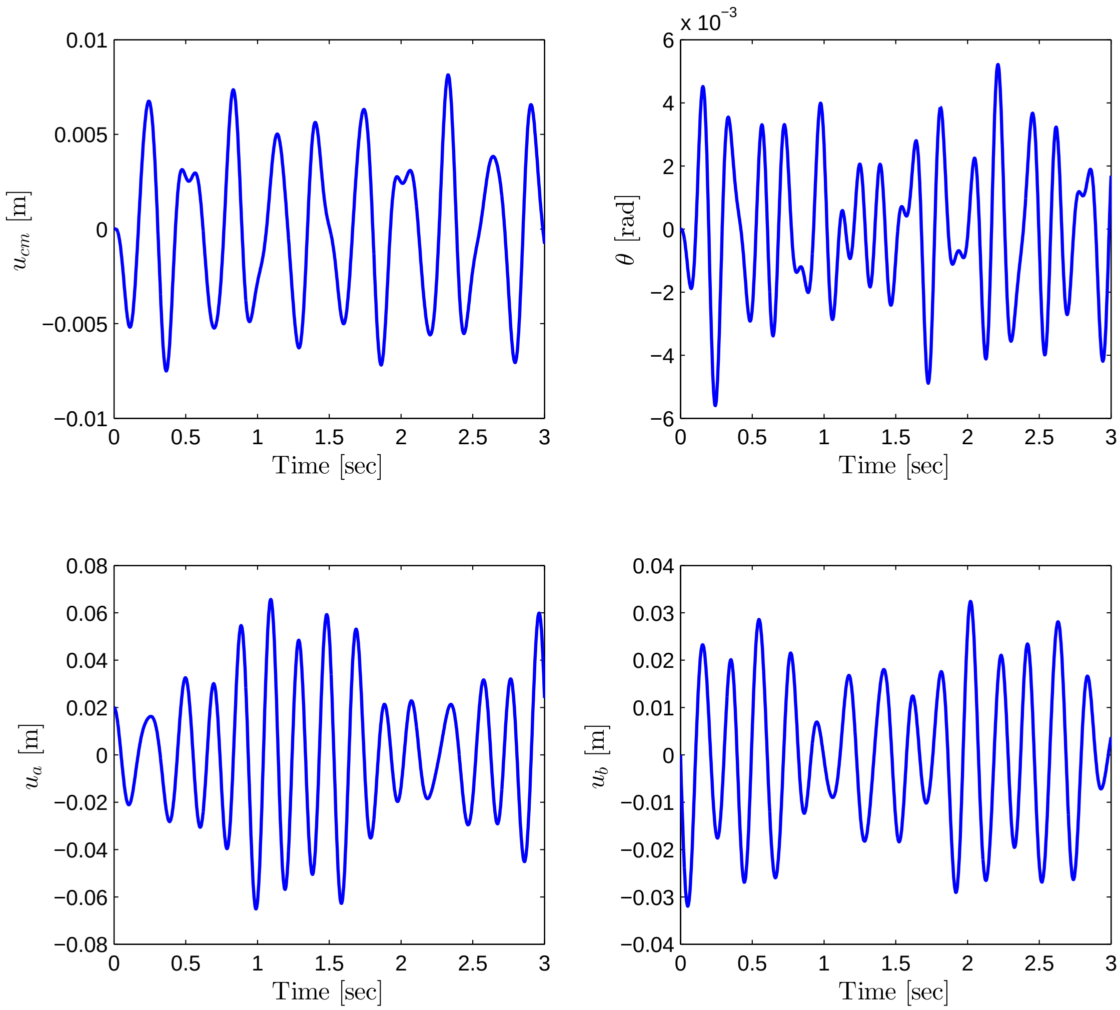

Figure 10. Result from ode45.

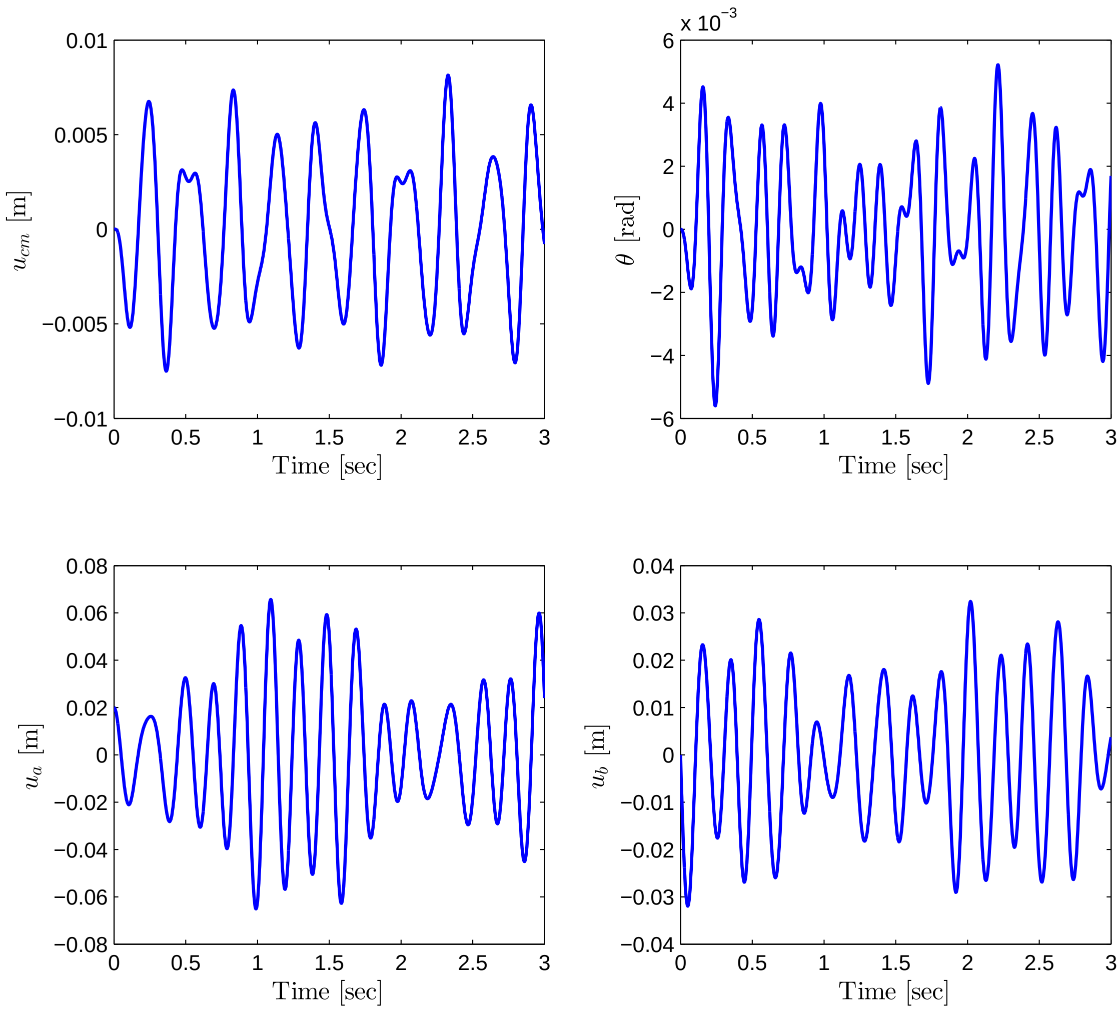

Figure 11. Result from MDOFODE45.

The plots match perfectly. To further validate the solution, the variation of total energy was plotted. The plot can be seen below.

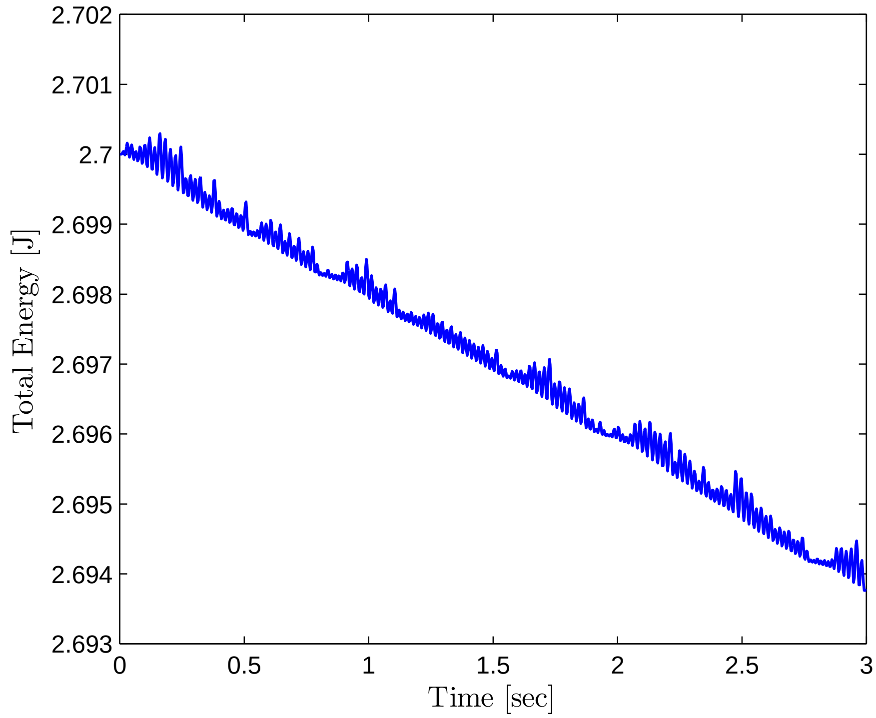

Figure 12. Variation of total energy with time.

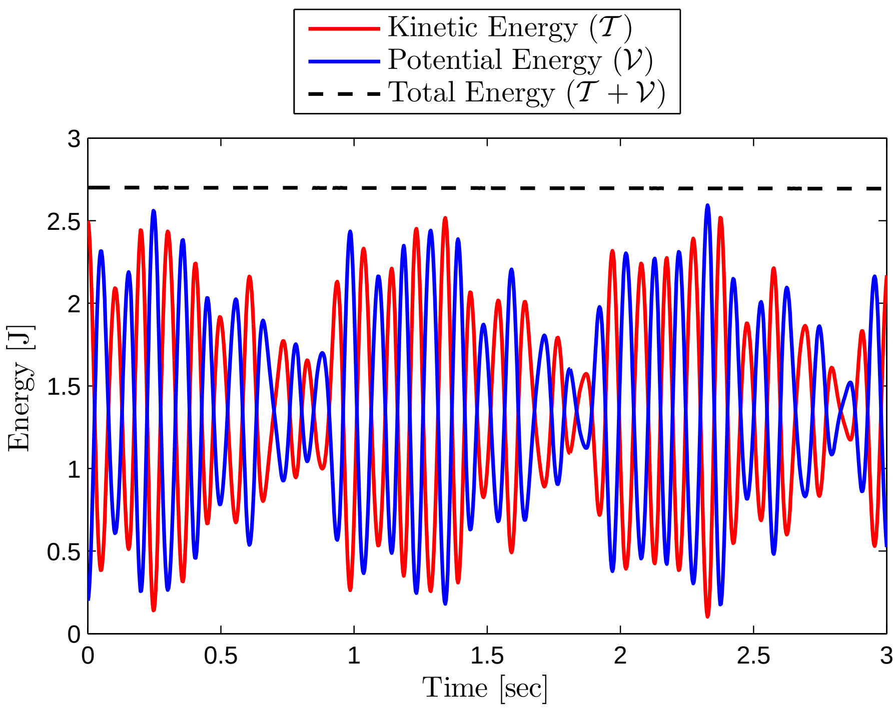

Figure 13. Variation of potential, kinetic, and total energies as functions of time.

By looking at Figure 13, the variation in total energy is very small compared to variation in kinetic and potential energies.

MATLAB code

Problems 1 and 2

```
%--------------------------------------------------------------------------
%% Clearing workspace
%--------------------------------------------------------------------------
clc,clear,close all

%--------------------------------------------------------------------------
%% Inputs
%--------------------------------------------------------------------------

L = 4;                                                       % m
r_1 = 2e-2;                                                  % m
r_2 = 5e-2;                                                  % m
rho = 2700;                                                  % kg/m^3
k_L = 10000;                                                 % N/m
k_R = 20000;                                                 % N/m

%--------------------------------------------------------------------------
%% Calculations
%--------------------------------------------------------------------------

% Finding the total mass
m = pi*rho*L/3*(r_2^2 + r_1*r_2 + r_1^2);
fprintf('m = %2.1f kg\n\n',m)

% Verifying analytical result with numerical result
mf = @(x) pi*rho*((r_2 - r_1)/L*x + r_1).^2;
m_numerical = integral(mf,0,L);
fprintf('Numerical result for total mass: \nm_numerical = %2.1f kg\n\n',...
    m_numerical)

% Finding the center of mass
xbar = L/4*(r_1^2 + 2*r_1*r_2 + 3*r_2^2)/(r_1^2 + r_1*r_2 + r_2^2);
fprintf('x_cm = %2.2f m \n\n',xbar)

% Verifying analytical result with numerical result
N = @(x) x.*(pi*rho*((r_2 - r_1)/L*x + r_1).^2);
D = @(x) pi*rho*((r_2 - r_1)/L*x + r_1).^2;
xbar_numerical = integral(N,0,L)/integral(D,0,L);
fprintf('Numerical result for center of mass:\nx_cm = %2.2f m\n\n',...
    xbar_numerical)

% Finding the moment of inertia

% Moment of inertia by integration
f = @(x) x.^2.*(pi*rho*((r_2-r_1)./L.*x + r_1).^2);
I_numerical = integral(f,0,L);
I_cm = I_numerical - m*xbar^2;
fprintf('I_cm = %2.1f kg*m^2\n\n',I_cm)

% Constructing mass matrix and stiffness matrix from EOM
M = [-I_cm/L            I_cm/L;
      m*(1 - xbar/L)    m*xbar/L];
```

```
K = [-xbar*k_L    (L - xbar)*k_R;
        k_L            k_R];

% Finding eigenvectors and eigenvalues
[Vec Val] = eigs(K,M);

% Sorting eigenvalues
lambda_1 = min(Val(Val~=0));
lambda_2 = max(Val(Val~=0));
fprintf('lambda_1 = %3.0f and lambda_2 = %3.0f \n\n',lambda_1,lambda_2)

% Separating eigenvectors
phi_1 = Vec(:,2);
phi_2 = Vec(:,1);

% Natural frequencies
omega_1 = sqrt(lambda_1);
omega_2 = sqrt(lambda_2);

% Printing natural frequencies
fprintf('omega_1 = %2.2f rad/s \n\nomega_2 = %2.2f rad/s \n\n',...
    omega_1,omega_2)

% Constructing geometry
c(1,:) = [0 -r_1];
c(2,:) = [L -r_2];
c(3,:) = [L r_2];
c(4,:) = [0 r_1];
c(5,:) = c(1,:);

% Shifting geometry up by 0.5 m
c(:,2) = c(:,2) + .5;

% Plotting system
figure('Position',[735 50 850 290])
hold on
plot(c(:,1),c(:,2),'-ok','LineWidth',1.5)
set(gca,'YLim',[0,max(c(:,2))+.25])
set(gca,'XLim',[min(c(:,1))-.25 max(c(:,1))+.25])

% From note 17A
b = L - xbar;
a = xbar;
k1 = k_L;
k2 = k_R;
I0 = I_cm;

% Mass matrix
Mnew = [(m*b^2 + I0)/(a + b)^2, -(I0 - a*b*m)/(a + b)^2;
        -(I0 - a*b*m)/(a + b)^2, (m*a^2 + I0)/(a + b)^2];

% Stiffness matrix
Knew = [k1, 0;
        0, k2];

% Finding eigenvectors and eigenvalues
[newVec newVal] = eigs(Knew,Mnew);

```

```
% Printing eigenvalues from equation in Note 17A
disp('Using the equation from Note 17A,')
fprintf('\nomega_1 = %2.2f rad/s \n\nomega_2 = %2.2f rad/s \n\n',...
    sqrt(newVal(2,2)),sqrt(newVal(1,1)))
```

Problems 3, 4, 5 and 6

```

function Hw7Prob3
%--------------------------------------------------------------------------
%% Clearing workspace
%--------------------------------------------------------------------------
clc,clear,close all

%--------------------------------------------------------------------------
%% Inputs
%--------------------------------------------------------------------------

L = 4;                                                      % m
k_a = 1000;                                                 % N/m
k_b = 5000;                                                 % N/m
k_L = 10000;                                                % N/m
k_R = 20000;                                                % N/m
a = 1;                                                      % m
b = 3;                                                      % m
m_a = 1;                                                    % kg
m_b = 5;                                                    % kg
m = 50;                                                     % kg
t_final = 3;                                                % sec

% Initial displacement of mass a
initial_displacement = 2e-2;                                % m

% Initital velocity of mass b
initial_velocity = -1;                                      % m/s

% Switch to save plots: 1 -> save, any other number won't save plots
saveplots = menu('Save plots?','Yes','No');

%--------------------------------------------------------------------------
%% Calculations
%--------------------------------------------------------------------------

% Moment of inertia
I = 1/12*m*L^2;

% % Matrices from Newton's law derivation
M1(1,1) = m;
M1(2,2) = I;
M1(3,3) = m_a;
M1(4,4) = m_b;

ell = L/2;

% K1 = zeros(size(M1));
```

```
K1(1,1) = k_L + k_R + k_a + k_b;
K1(1,2) = ell*(-k_L + k_R - k_a - k_b) + a*k_a + b*k_b;
K1(1,3) = -k_a;
K1(1,4) = -k_b;
K1(2,1) = ell*(-k_L + k_R - k_a - k_b) + a*k_a + b*k_b;
K1(2,2) = ell^2*(k_L + k_R + k_a + k_b) - 2*a*ell*k_a - 2*b*ell*k_b + ...
    a^2*k_a + b^2*k_b;
K1(2,3) = (ell - a)*k_a;
K1(2,4) = (ell - b)*k_b;
K1(3,1) = -k_a;
K1(3,2) = (ell - a)*k_a;
K1(3,3) = k_a;
K1(4,1) = -k_b;
K1(4,2) = (ell - b)*k_b;
K1(4,4) = k_b;

[Vec1 Val1] = eigs(K1,M1);

% Sorting eigenvalues
Val1 = sort(diag(Val1));

% Natural freqencies
W = sqrt(Val1);
F = W/(2*pi);

% Outputting results
disp('-------------------------------------------------------------------')
disp('Results from application of Newton''s Law: ')
disp('-------------------------------------------------------------------')
disp('Eigenvectors: ')
disp(Vec1)
fprintf(['lambda_1 = %3.1f\nlambda_2 = %3.1f\nlambda_3 = %4.0f\n',...
    'lambda_4 = %4.0f\n\n'],Val1)
fprintf(['omega_1 = %2.2f rad/s\nomega_2 = %2.2f rad/s\n',...
    'omega_3 = %2.2f rad/s\nomega_4 = %2.2f rad/s\n\n'],W)
fprintf('f_1 = %3.3f Hz\nf_2 = %3.3f Hz\nf_3 = %3.3f Hz\nf_4 = %3.3f Hz\n\n',F)

% Matrices from Lagrange's equation

% Creating data structure
Ndof = 4;
mdof = struct('m',[],'k',[],'Tm',[],'Tk',[]);

mdof(1).Tm = [1 0 0 0];
mdof(2).Tm = [0 1 0 0];
mdof(3).Tm = [0 0 1 0];
mdof(4).Tm = [0 0 0 1];

mdof(1).m = m;
mdof(2).m = I;
mdof(3).m = m_a;
mdof(4).m = m_b;

mdof(1).k = k_L;
mdof(2).k = k_R;
mdof(3).k = k_a;
mdof(4).k = k_b;

mdof(1).Tk = [1 -ell 0 0];
```

```
mdof(2).Tk = [1 ell 0 0];
mdof(3).Tk = [-1 (ell - a) 1 0];
mdof(4).Tk = [-1 (ell - b) 0 1];

M2 = zeros(size(M1));
K2 = M2;

for n = 1:Ndof
    M2 = M2 + mdof(n).m*(mdof(n).Tm')*mdof(n).Tm;
    K2 = K2 + mdof(n).k*(mdof(n).Tk')*mdof(n).Tk;
end

[Vec2 Val2] = eigs(K2,M2);

% Sorting eigenvalues
Val2 = sort(diag(Val2));
W = sqrt(Val2);
F = W/(2*pi);

% Outputting results
disp('-------------------------------------------------------------------')
disp('Results from application of Lagrange''s Equations: ')
disp('-------------------------------------------------------------------')
disp('Eigenvectors: ')
disp(Vec2)
fprintf(['lambda_1 = %3.1f\nlambda_2 = %3.1f\nlambda_3 = %4.0f\n',...
    'lambda_4 = %4.0f\n\n'],Val2)
fprintf(['omega_1 = %2.2f rad/s\nomega_2 = %2.2f rad/s\n',...
    'omega_3 = %2.2f rad/s\nomega_4 = %2.2f rad/s\n\n'],W)
fprintf('f_1 = %3.3f Hz\nf_2 = %3.3f Hz\nf_3 = %3.3f Hz\nf_4 = %3.3f Hz\n\n',F)

%--------------------------------------------------------------------------
%% Calling MDOFODE45
%--------------------------------------------------------------------------

u0 = [0; 0; initial_displacement; 0];
v0 = [0; 0; 0; initial_velocity];
P0 = [0; 0; 0; 0];
Ps = P0;
Ws = 0;
Pc = Ps;
Wc = Ws;
C = zeros(size(M1));
tSPAN = linspace(0,t_final,1e3);

% Function call to MDOFODE45
[u,t] = MDOFODE45(tSPAN,M1,K1,C,u0,v0,P0,Ps,Ws,Pc,Wc);

figure('Position',[780 50 800 690])
subplot(2,2,1)
plot(t,u(:,1),'-b','LineWidth',1.5)
xlabel('Time [sec]','Interpreter','LaTeX','FontSize',12)
ylabel('$u_{cm}$ [m]','Interpreter','LaTeX','FontSize',12)

subplot(2,2,2)
plot(t,u(:,2),'-b','LineWidth',1.5)
xlabel('Time [sec]','Interpreter','LaTeX','FontSize',12)
ylabel('$\theta$ [rad]','Interpreter','LaTeX','FontSize',12)

```

```
subplot(2,2,3)
plot(t,u(:,3),'-b','LineWidth',1.5)
xlabel('Time [sec]','Interpreter','LaTeX','FontSize',12)
ylabel('$u_a$ [m]','Interpreter','LaTeX','FontSize',12)

subplot(2,2,4)
plot(t,u(:,4),'-b','LineWidth',1.5)
xlabel('Time [sec]','Interpreter','LaTeX','FontSize',12)
ylabel('$u_b$ [m]','Interpreter','LaTeX','FontSize',12)

% Saving plot
if saveplots == 1
    ImgPath = ['C:\Users\James\Desktop\School\Courses\UTA\',...
        'AE 5311 - Structural Dynamics\Images\'];
    set(gcf,'PaperPositionMode','auto')
    print(gcf, '-depsc', [ImgPath,'Hw7Prob6_MDOFODE45.eps'])
end

%------------------------------------------------------------------------
%% Calling my function for ode45
%------------------------------------------------------------------------

IC = [0; 0; initial_displacement; 0;
    0; 0; 0; initial_velocity];
[tn,un] = ode45(@myfunHw7Prob6,tSPAN,IC);

figure('Position',[780 50 800 690])
subplot(2,2,1)
plot(tn,un(:,1),'-b','LineWidth',1.5)
xlabel('Time [sec]','Interpreter','LaTeX','FontSize',12)
ylabel('$u_{cm}$ [m]','Interpreter','LaTeX','FontSize',12)

subplot(2,2,2)
plot(tn,un(:,2),'-b','LineWidth',1.5)
xlabel('Time [sec]','Interpreter','LaTeX','FontSize',12)
ylabel('$\theta$ [rad]','Interpreter','LaTeX','FontSize',12)

subplot(2,2,3)
plot(tn,un(:,3),'-b','LineWidth',1.5)
xlabel('Time [sec]','Interpreter','LaTeX','FontSize',12)
ylabel('$u_a$ [m]','Interpreter','LaTeX','FontSize',12)

subplot(2,2,4)
plot(tn,un(:,4),'-b','LineWidth',1.5)
xlabel('Time [sec]','Interpreter','LaTeX','FontSize',12)
ylabel('$u_b$ [m]','Interpreter','LaTeX','FontSize',12)

% Saving plot
if saveplots == 1
    set(gcf,'PaperPositionMode','auto')
    print(gcf, '-depsc', [ImgPath,'Hw7Prob6_ode45.eps'])
end

% Plotting total energy for the system

% Kinetic energy
T = 1/2*m*un(:,5).^2 + 1/2*I*un(:,6).^2 + 1/2*m_a*un(:,7).^2 ...
    + 1/2*m_b*un(:,8).^2;

```

```
% Lengths springs are stretched
Delta_L = un(:,1) - ell*un(:,2);
Delta_R = un(:,1) + ell*un(:,2);
Delta_a = -un(:,1) + (ell - a)*un(:,2) + un(:,3);
Delta_b = -un(:,1) + (ell - b)*un(:,2) + un(:,4);

% Potential energy
V = 1/2*k_L*Delta_L.^2 + 1/2*k_R*Delta_R.^2 + 1/2*k_a*Delta_a.^2 + ...
    1/2*k_b*Delta_b.^2;

% Total Energy
E = T + V;
figure
plot(tn,E,'LineWidth',1.1)
xlabel('Time [sec]','Interpreter','LaTeX','FontSize',12)
ylabel('Total Energy [J]','Interpreter','LaTeX','FontSize',12)

% Saving plot
if saveplots == 1
    set(gcf,'PaperPositionMode','auto')
    print(gcf, '-depsc', [ImgPath,'Hw7Prob6_E.eps'])
end

% Kinetic and potential energies
figure
plot(tn,T,'-r',tn,V,'-b',tn,E,'--k','LineWidth',1.5)
xlabel('Time [sec]','Interpreter','LaTeX','FontSize',12)
ylabel('Energy [J]','Interpreter','LaTeX','FontSize',12)
lh = legend('Kinetic Energy ($\mathcal{T}$)',...
    'Potential Energy ($\mathcal{V}$)',...
    'Total Energy ($\mathcal{T} + \mathcal{V}$)');
set(lh,'Location','NorthOutside')
set(lh,'Interpreter','LaTeX')
set(lh,'FontSize',12)

% Saving plot
if saveplots == 1
    set(gcf,'PaperPositionMode','auto')
    print(gcf, '-depsc', [ImgPath,'Hw7Prob6_TandV.eps'])
end

%--------------------------------------------------------------------------
%% Nested function for ode45
%--------------------------------------------------------------------------

    function dx = myfunHw7Prob6(t,x)

        dx = zeros(8,1);

        % Velocities
        dx(1) = x(5);
        dx(2) = x(6);
        dx(3) = x(7);
        dx(4) = x(8);

        % u_c double dot
        dx(5) = 1/m*(-(k_L + k_R + k_a + k_b)*x(1) - ...
            (-ell*k_L + ell*k_R - ell*k_a - ell*k_b + a*k_a + b*k_b)...
            *x(2) - (-k_a)*x(3) - (-k_b)*x(4));
```

```

        % theta double dot
        dx(6) = 1/I*(-(ell*(-k_L + k_R - k_a - k_b) + a*k_a + b*k_b)*...
            x(1) - (ell^2*(k_L + k_R + k_a + k_b) - 2*a*ell*k_a - ...
            2*b*ell*k_b + a^2*k_a + b^2*k_b)*x(2) - (k_a*(ell - a))*x(3)...
            - k_b*(ell - b)*x(4));

        % u_a double dot
        dx(7) = 1/m_a*(k_a*x(1) - k_a*(ell - a)*x(2) - k_a*x(3));

        % u_b double dot
        dx(8) = 1/m_b*(k_b*x(1) - k_b*(ell - b)*x(2) - k_b*x(4));

    end

end
```