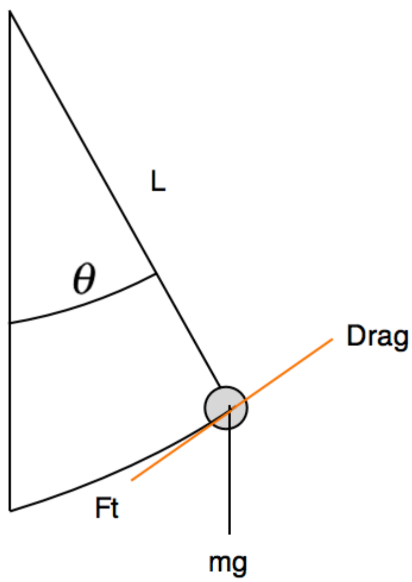
Assignment 8: Ordinary Differential Equations

Date Due: Solution Instructor: Trani

Problem 1

The differential equation to predict the angular acceleration ($\ddot{\theta}$)of a rigid pendulum can be derived using a free body diagram and Newton's second law of motion shown in Figure 1.



$$\sum F_t = ma_t = -mg\sin(\theta) - Kl\dot{\theta}$$

$$ma_t = ml\ddot{\theta} = -mg\sin(\theta) - Kl\dot{\theta}$$

$$\ddot{\theta} = \frac{-g\sin(\theta)}{l} - \frac{K\dot{\theta}}{m}$$

Figure 1. Free-Body Diagram for a Rigid Pendulum.

$$\sum F_t = ma_t = -mg\sin(\theta) - Kl\dot{\theta}$$

$$ma_t = ml\ddot{\theta} = -mg\sin(\theta) - Kl\dot{\theta}$$

$$\ddot{\theta} = \frac{-g\sin(\theta)}{l} - \frac{K\dot{\theta}}{m} \qquad (1)$$

where: K is the drag damping factor, θ is the angular position (in radians) between the vertical and the position of the rigid pendulum, $\dot{\theta}$ is the angular velocity, m is the mass of the pendulum (kg) and l is the length of the pendulum (meters). The units of the Equation (1) are consistent to make the units of $\ddot{\theta}$ rad/s2.

Task 1

Create a Matlab script and a function that solves the set of two differential equations shown above. The objective is to calculate the angular position (θ) and the angular speed ($\dot{\theta}$) of the pendulum. Solve the differential equations for the following initial conditions:

$\dot{\theta} = 0$ and $\theta = 0.20$ radians. In this solution assume the values of K to be 0.36 kg/s, g=9.81 m/s^2, l = 0.85 meters, and m =32 kg.

```
% Main file to investigate pendulum dynamics
% A. Trani
% November 10, 2013

clear
% Solution to a set of equations of the form:

% y(1) = angular speed (rad/s)
% y(2) = angular displacement (rad)
%
% ydot(1) = - g * sin(y(2)) / l - K * y(1) /m;
% ydot(2) = y(1)
%
% subject to initial conditions:
%
% y (t=0) = yo
%
% where:


global K l m g

% Define Initial Conditions of the Problem
```


```
% Define Initial Conditions of the Problem
```


```

yo = [0.2 0.0];            % yo are the initial displacement an
to = 0.0;                  % to is the initial time to solve this e
tf = 20;                   % tf is the final time (seconds)

% define parameters

K = 0.36 ;          % damping constant  (kg/s)
m = 32;           % mass (kg)
l = 0.85;          % length in meters
g = 9.81;         % m/s-s
tspan =[to tf];

[t,y] = ode45('pendulum',tspan,yo); % call ODE solver

% Plot the results of the numerical integration procedure

figure
plot(t,y(:,1))
xlabel('Time (seconds)')
ylabel('Angular Speed (rad/s)')
grid
```

Figure 2. Main Program to Estimate the Pendulum Dynamics.

```
% Two first-order DEQ to solve the rigid pendulum

function ydot = pendulum(t,y)

global K l m g

% y(1) = angular speed (rad/s)
% y(2) = angular displacement (rad)
%
% ydot(1) = - g * sin(y(2)) / l + K * y(1) /m;
% ydot(2) = y(1)
%
%
ydot(1) = - g * sin(y(2)) / l - K * y(1) /m;
ydot(2) = y(1);

ydot = ydot';
```

Figure 3. Function to Calculate Pendulum Parameters.

Task 2

Plot the values of the state variables position (θ) and the angular speed ($\dot{\theta}$) as a function of time (from 0-60 seconds). Label the plots accordingly. Also plot the angular acceleration and plot it with respect to time. Estimate the maximum velocity for the pendulum,for the given initial conditions.

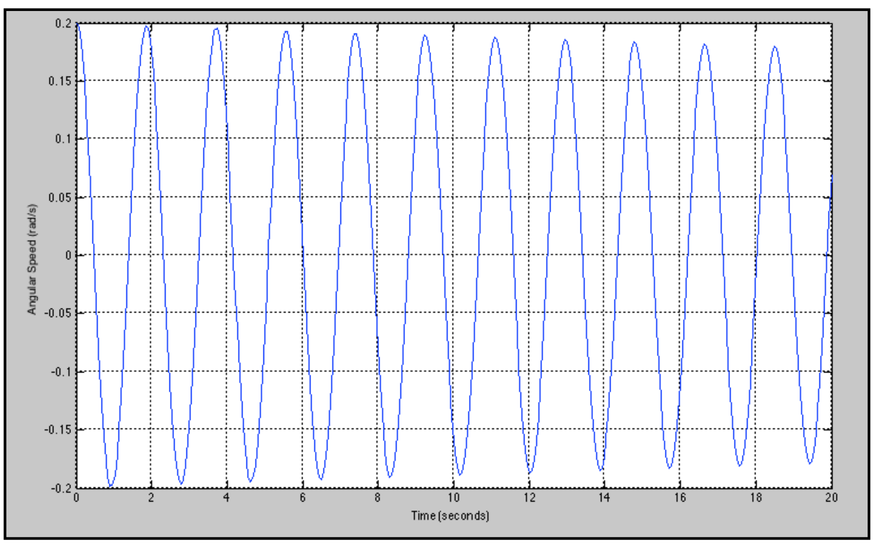


Figure 4. Plot of Angular Speed over Time.

Task 3

Make a plot of angular speed vs. angular position (so called Phase plot). Comment on the response observed.

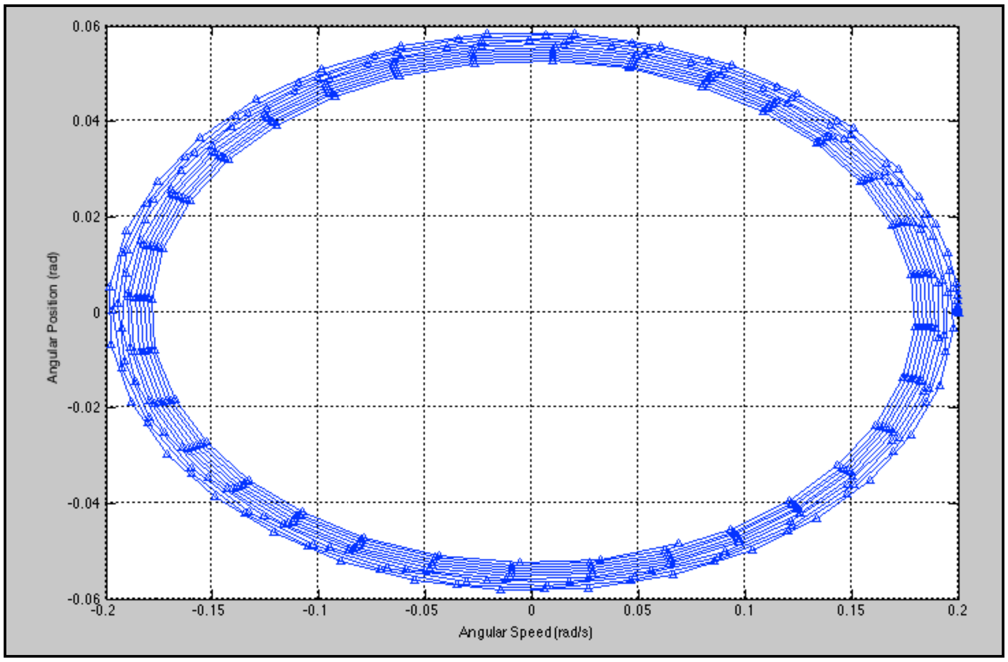


Figure 5. Plot of Angular Speed vs. Angular Position.

Problem 2

A civil engineer needs to model the dynamics of a building subject to seismic loads. The building and its foundation is modeled as series of spring and dampers as shown in Figure 6.

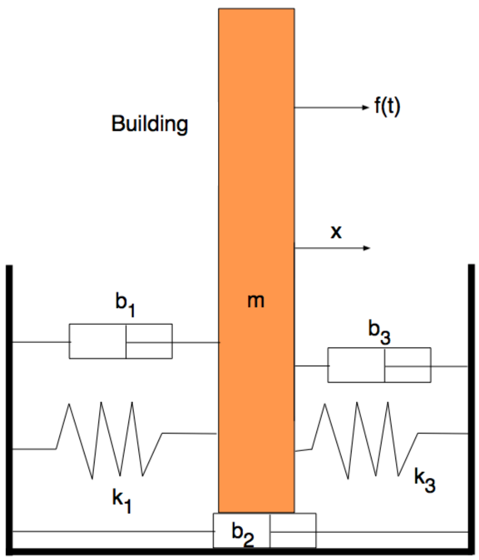


Figure 6. Building and Foundation System for Problem 2.

Task 1

Use a free-body diagram with all forces acting on the building system (shown in Figure 2) to verify the basic equation of motion. In this analysis the displacement (x) towards the right is positive. Use Newton's second law of Physics to verify the that the equation of motion for the acceleration of the building at the foundation is:

$$\ddot{x} = \frac{f(t)}{m} + \left(\frac{b_3 - b_2 - b_1}{m}\right)\dot{x} + \left(\frac{k_2 - k_1}{m}\right)x$$

where:

$\ddot{x} =$ lateral acceleration of the building system (m/s^2)

$\dot{x} =$ lateral speed of building system (m/s)

$x =$ lateral displacement of building system (m)

The values of the constants are:

m = 2000000;	% mass (kg)
b1 = 500000;	% damper constant (N / (m/s))
b2 = 550000;	% damper constant (N / (m/s))
b3 = 600000;	% damper constant (N / (m/s))
k1 = 9e6;	% spring constant (N/m)
k2 = 7e6;	% spring constant (N/s)

Task 2

Using the Matlab ODE solver function, construct a script and a function to model the building system and solve the differential equations of motion of the building system to find the dynamic response of x and $\dot{x}$ as a function of time.

```
% Main file to execute the mass-spring-damper system

% Solution to a set of eauations of the form:

% let y(1) = position (m)
%     y(2) = speed (m/s)
%
% ydot(1) = y(2);
% ydot(2) = F(t)/m + (k2-k1)/m y(1) + (b3-b2-b1)/m y(2)
%
% subject to initial conditions:
%
% y (t=0) = yo
%
% where:

global m b1 b2 b3 k1 k2

% Define Initial Conditions of the Problem

yo = [0.1 0];              % yo are the initial displacement and velocity
to = 0.0;                   % to is the initial time to solve this equation
tf = 50.0;                   % tf is the final time
```

```

% define values of m, b and k

% m = 500;        % mass (kg)
% b1 = 500;       % damper constant (N / (m/s))
% b2 = 550;       % damper constant (N / (m/s))
% b3 = 600;       % damper constant (N / (m/s))
% k1 = 1000;         % spring constant (N/m)
% k2 = 200;        % spring constant (N/s)

m = 2000000;       % mass (kg)
b1 = 500000;          % damper constant (N / (m/s))
b2 = 550000;          % damper constant (N / (m/s))
b3 = 600000;          % damper constant (N / (m/s))
k1 = 9e6;             % spring constant (N/m)
k2 = 7e6;             % spring constant (N/s)

tspan =[to tf];

[t,y] = ode45('rateEq_2springs_3dashpots',tspan,yo);    % call ODE solver

% Plot the results of the numerical integration procedure
```


```
[t,y] = ode45('rateEq_2springs_3dashpots',tspan,yo);    % call ODE solver

% Plot the results of the numerical integration procedure
```


```

figure
plot(t,y(:,1),'o-')
xlabel('Time (seconds)','fontsize',20)
ylabel('Position (m)','fontsize',20)
grid


figure
plot(t,y(:,2),'^--r')
xlabel('Time (seconds)','fontsize',20)
ylabel('Speed (m/s)','fontsize',20)
grid
```

Figure 7. Matlab Code - Main File - for Problem 2.

```
% Two first-order DEQ to solve the mass-spring-damper system
function yprime = rateEq_2springs_3dashpots(t,y)

global  m b1 b2 b3 k1 k2
% define the forcing function, f(t), here
% f = 100;
time = [0 1.2 4.6 7.6 10.2 13 18 23.2 25.3 30 100] ;
ftable = [0 0.3 4.2 1.1 4.5 0.8 3.6 0.2 0.3 0 0] * 1e5; % Newtons

f = interp1(time,ftable,t, 'pchp');
% f=0;

% define the rate equations
% y(1) = position (m)
% y(2) = speed (m/s)

yprime(1) = y(2);
yprime(2) = f/m + (k2-k1)/m * y(1) + (b3-b2-b1)/m * y(2);

yprime = yprime';
```

Figure 8. Matlab Function for Problem 2.

Task 4

Simulate a small seismic event by adding numerical values to the external forcing function $f(t)$ as shown in the table below. It is recommended that adding the forcing function you model the response by interpolating values of force over time. For example, at time 5 seconds, the value of the seismic force is unknown in the table and thus needs to be calculated by interpolation.

Plot speed and lateral displacement of the building system and record the maximum values observed. Find the time when the system lateral displacement has reached 5% of the maximum input displacement (assumed to be 10 cm). Comment on the observed behavior.

Table 1. Forcing Function Values of Simulated Seismic Event.

Time of Event (seconds)	Force (N)
0.0	0.0
1.2	30000.0
4.6	420000.0
7.6	110000.0
10.2	450000.0
13.0	80000.0
18.0	360000.0
23.2	20000.0
25.3	30000.0
30.0	0.0
100.0	0.0

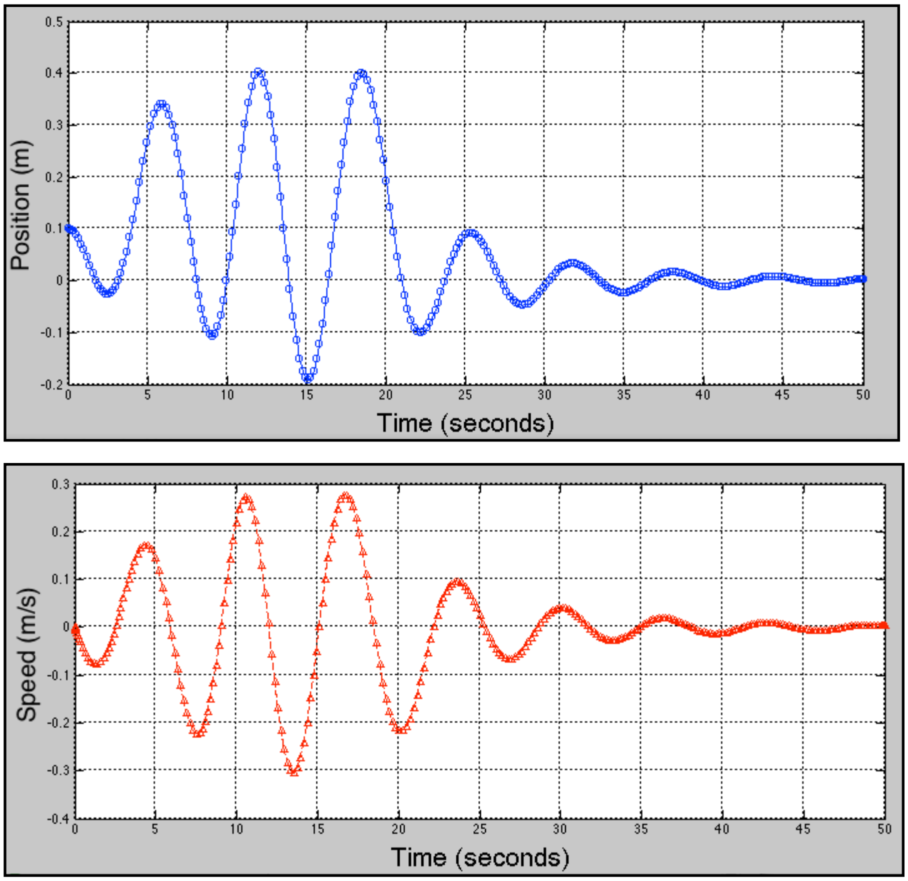


Figure 9. Position and Speed Plot for Problem 2.