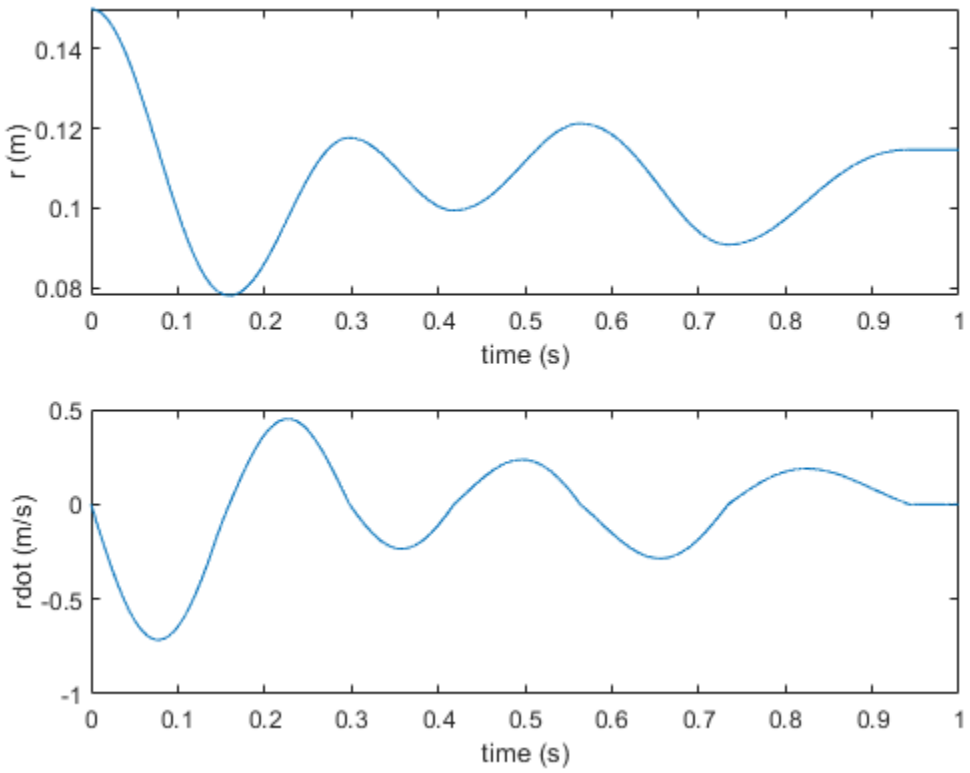
Table of Contents


```
%midterm.m
```

Problem 1c and 1d

```
clear; clc; close;
global mu g m k r0;
mu = .12; %friction coefficient
g = 9.81; %Gravity acceleration (m/s^2)
m = 2; %kg
k = 1000; %N/m
r0 = .1; %m

dt = 0.0001;  % Step size (s)
tf = 1;   % End time (s)
y0 = 1.5*r0;  % Initial position (m)
v0 = 0;   % Initial velocity (m/s)

% Generate time vector
t = 0:dt:tf;
N = length(t);

% Initial conditions
y0 = [y0 v0];

% Call 'ode45' to solve ODE.  'ode45' calls the function
% 'bead_wire_function' repeatedly, which returns the two
% derivatives at each time step.  'ode45' uses the returned
% derivatives to calculate the solution at each time step.
% The solution is returned in the variable y, which contains
% both y1 (x) and y2 (v).
[t,y] = ode45(@mid_1_function,t, y0);

% Extract solution from y
r = y(:,1);  % Position
r_dot = y(:,2);  % Velocity

figure(1)
subplot(2,1,1)
plot(t,r)
xlabel('time (s)')
ylabel('r (m)')
subplot(2,1,2)
plot(t,r_dot)
```

```
xlabel('time (s)')
ylabel('rdot (m/s)')

max_r = max(r)
min_r = min(r)
max_rdot = max(r_dot)
min_rdot = min(r_dot)
```



problem 3

```
clear; clc; close;
theta = 30*pi/180; %rad
L = .03; %m
R = .01; %m
w1 = 12; %rad/s
theta_dot = -7; %rad/s
theta_ddot = 30; %rad/s^2
m = .05; %kg
ax = R*theta_ddot*cos(theta)-R*theta_dot^2*sin(theta) ...
    +2*R*theta_dot*sin(theta)*w1-R*sin(theta)*w1^2;
ay = -R*theta_ddot*sin(theta)-R*theta_dot^2*cos(theta) ...
    +2*R*theta_dot*cos(theta)*w1-(L+R*cos(theta))*w1^2;
F = m.*[ax;ay;0];
mag_F = m*sqrt((ax)^2+(ay^2))
```

```
mag_F =

    0.3876
```

function for midterm_1c

```
% mid_1_function
% This is the function that 'ode45' calls to get the function
 derivatives
% at each time step.  'ode45' passes the current time and states (x
 and v,
% which are contained in y), and the function returns the current
 state
% derivatives (xdot and vdot, which are contained in ydot).
%
% OG Author:   Mark Colton
% Edits: Spencer Jensen
% Date:     2/25/21

function ydot = mid_1_function(t,y)

% Access the global variables (model parameters) defined in the main
% function
global mu g m k r0;

% Extract the current states from the current y vector
y1 = y(1);
y2 = y(2);
theta_dot = 1.2*6.5*cos(6.5*t);
theta_ddot = -1.2*6.5*6.5*sin(6.5*t);

% Find the state derivatives
y1dot = y2;
y2dot = theta_dot^2*y1-k/m*(y1-r0)- ...
    mu*sqrt((theta_ddot*y1+2*theta_dot*y2)^2+g^2)*sign(y2);

% Reassemble the state derivatives into a single vector, ydot, to pass
% back to 'ode45'
ydot = [y1dot; y2dot];
end


max_r =

    0.1500


min_r =

    0.0781
```

```
max_rdot =

    0.4502

min_rdot =

   -0.7156
```

Published with MATLAB® R2019a