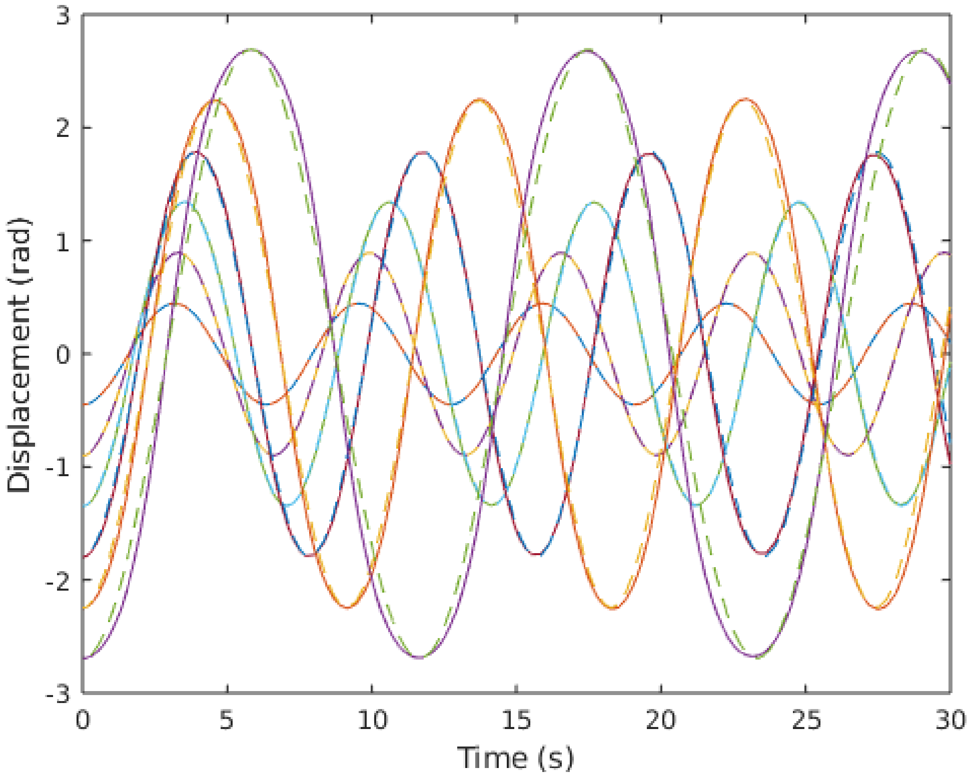
Assignment 1---Brady Metherall

Table of Contents

Question 1---Simple Pendulum

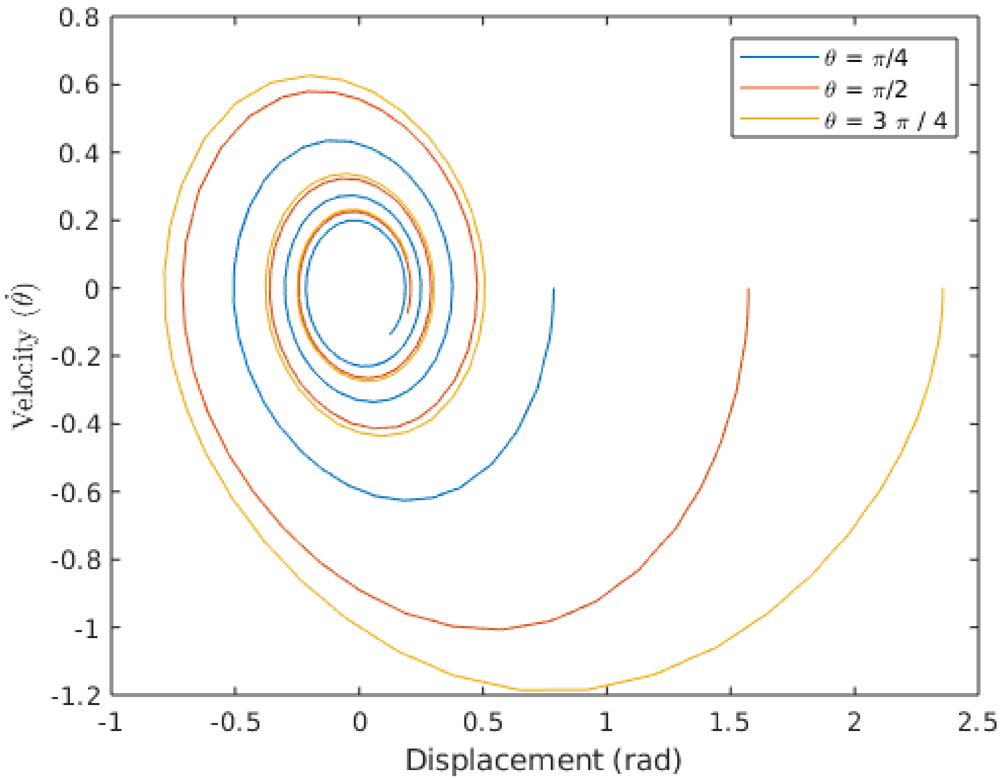
```
% Solve theta'' + omega_0^2 sin(theta) = 0
% as the first order system:
% theta' = phi
% phi' = -omega_0^2 sin(theta)

omega_0 = 1;

% Loop over 6 initial displacements in increments of pi/7
for theta_0 = (1:6).*pi/7 % Initial angular displacement
    t_range = [0 30]; % Set domain
    IC = [-theta_0 0]; % Set initial conditions
    [t, theta] = ode45(@(t,y) [y(2); -omega_0^2.*sin(y(1))], t_range,
 IC); % Numerically solve the first order system described above

    figure(1)
    plot(t, theta(:,1)) % Plot numerical solution
    hold on
    plot(t, -theta_0 * cos(t * pi * omega_0 / (2 *
 ellipticK(sin(theta_0 / 2)^2))), '--') % Plot analytic approximation

    xlabel('Time (s)');
    ylabel('Displacement (rad)');
end

% For large amplidude (large theta_0) it's clear from the graph that
 the soution is not sinusoidal
```



Question 2---Damped Pendulum

```
% Solve theta'' + omega_0^2 sin(theta) + gamma * |theta'| * theta' = 0
% as the first order system:
% theta' = phi
% phi' = -omega_0^2 sin(theta) - gamma * |theta'| * theta'

clear

omega_0 = 1;
gamma = 1/2; % Damping coefficient

% Loop over 3 initial displacements in increments of pi/4
for theta_0 = (1:3).*pi/4 % Initial angular displacement
    t_range = [0 20]; % Set domain
    IC = [theta_0 0]; % Set initial conditions
    [~, theta] = ode45(@(t,y) [y(2); -omega_0^2.*sin(y(1)) - gamma *
 abs(y(2)) * y(2)], t_range, IC); % Numerically solve the first order
 system described above

    figure(2)
    plot(theta(:,1), theta(:,2)) % Plot numerical solution
    hold on

    xlabel('Displacement (rad)');
    ylabel('Velocity ($\dot{\theta}$)', 'Interpreter','latex');
```

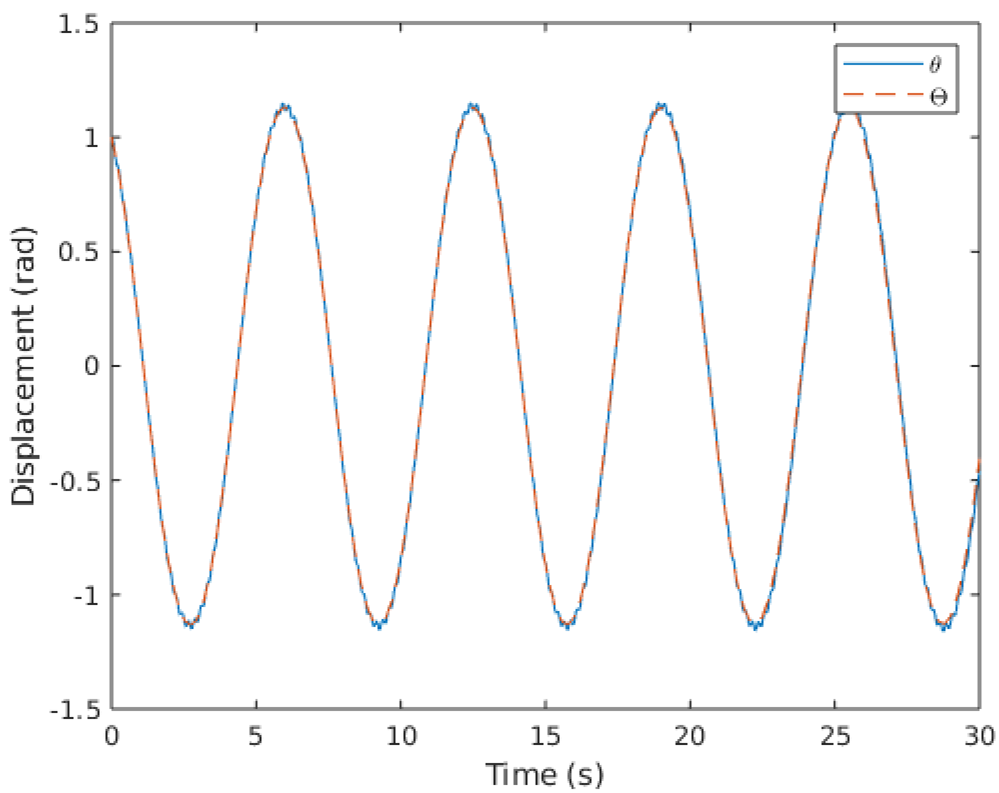
```
end

legend('\theta = \pi/4', '\theta = \pi/2', '\theta = 3 \pi / 4')
```

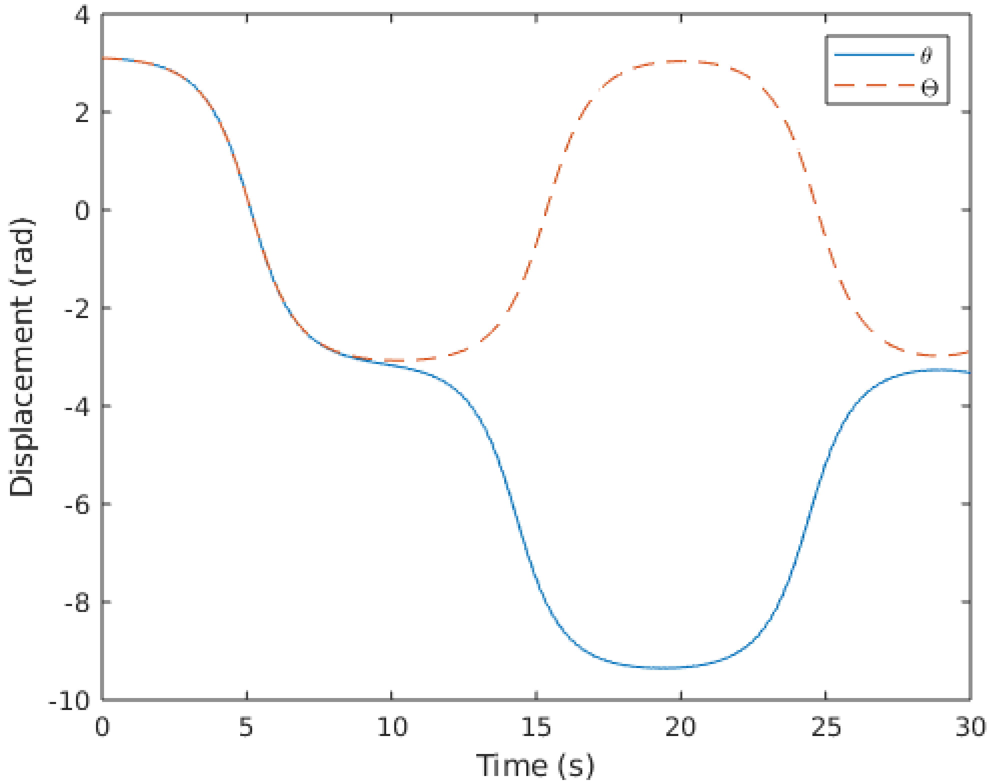


Question 3---Driven Pendulum

```
% Solve theta'' + omega_0^2 sin(theta) + b omega / L sin(theta)
 sin(omega t) = 0
% as the first order system:
% theta' = phi
% phi' = -omega_0^2 sin(theta) - b omega / L sin(theta) sin(omega t)
% Additionally, the time-averaged solution can be found by solving
% Theta'' + omega_0^2 sin(Theta) + b^2 omega^2 / (2 L^2) sin(Theta)
 cos(Theta) = 0
% as the first order system:
% Theta' = Phi
% Phi' = -omega_0^2 sin(Theta) - b^2 omega^2 / (2 L^2) sin(Theta)
 cos(Theta)

clear

omega_0 = 1;
L = 1; % Length of pendulum
b = 0.02; % Strength of driving force
omega = 30; % Driving frequency
```

```
theta_0 = [1 3.1]; % Initial angular displacement
t_range = [0 30]; % Set domain

for i = 1:2
    IC = [theta_0(i) 0]; % Set initial conditions for full solution
    [t1, theta] = ode45(@(t,y) [y(2); -omega_0^2 * sin(y(1)) -
 b * omega^2 / L * sin(y(1)) * sin(omega * t)], t_range, IC); %
 Numerically solve the full system described above
    IC = [theta_0(i) -b * omega / L * sin(theta_0(i))]; % Set initial
 conditions for time-averaged solution
    [t2, theta_slow] = ode45(@(t,y) [y(2); -omega_0^2 * sin(y(1)) -
 b^2 * omega^2 / (2 * L^2) * sin(y(1)) * cos(y(1))], t_range, IC); %
 Numerically solve the time-averaged system described above

    figure(i+2)
    plot(t1, theta(:,1)) % Plot full numerical solution
    hold on
    plot(t2, theta_slow(:,1), '--') % Plot time-averaged numerical
 solution

    xlabel('Time (s)');
    ylabel('Displacement (rad)');
    legend('\theta', '\Theta')
end
```

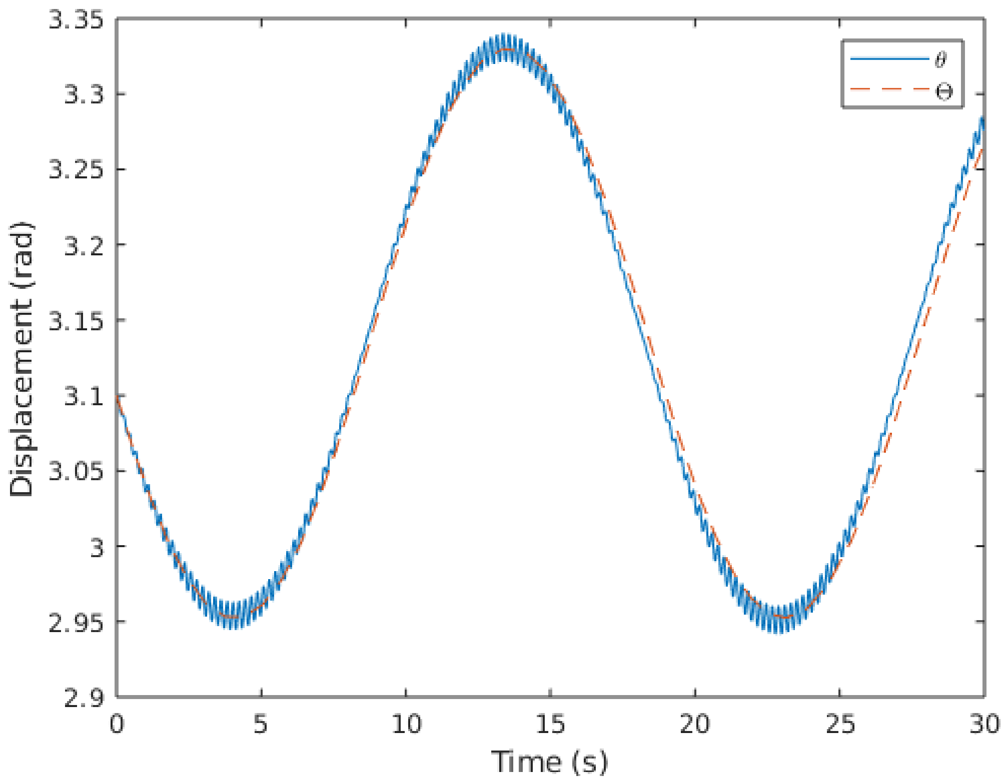




Question 4---Driven Pendulum (Cont.)

```
% This has the same setup as question 3, but with a stronger driving
 force (parameter b)

clear

omega_0 = 1;
L = 1; % Length of pendulum
b = 0.05; % Strength of driving force
omega = 30; % Driving frequency

theta_0 = 3.1; % Initial angular displacement
t_range = [0 30]; % Set domain

IC = [theta_0 0]; % Set initial conditions
[t1, theta] = ode45(@(t,y) [y(2); -omega_0^2 * sin(y(1)) - b
 * omega^2 / L * sin(y(1)) * sin(omega * t)], t_range, IC); %
 Numerically solve the full system described above
IC = [theta_0 -b * omega / L * sin(theta_0)];
[t2, theta_slow] = ode45(@(t,y) [y(2); -omega_0^2 * sin(y(1)) - b^2
 * omega^2 / (2 * L^2) * sin(y(1)) * cos(y(1))], t_range, IC); %
 Numerically solve the time-averaged system described above

figure(5)
```

```
plot(t1, theta(:,1)) % Plot the full numerical solution
hold on
plot(t2, theta_slow(:,1), '--') % Plot the time-averaged numerical
 solution

xlabel('Time (s)');
ylabel('Displacement (rad)');
legend('\theta', '\Theta')
```



Question 5---Animation

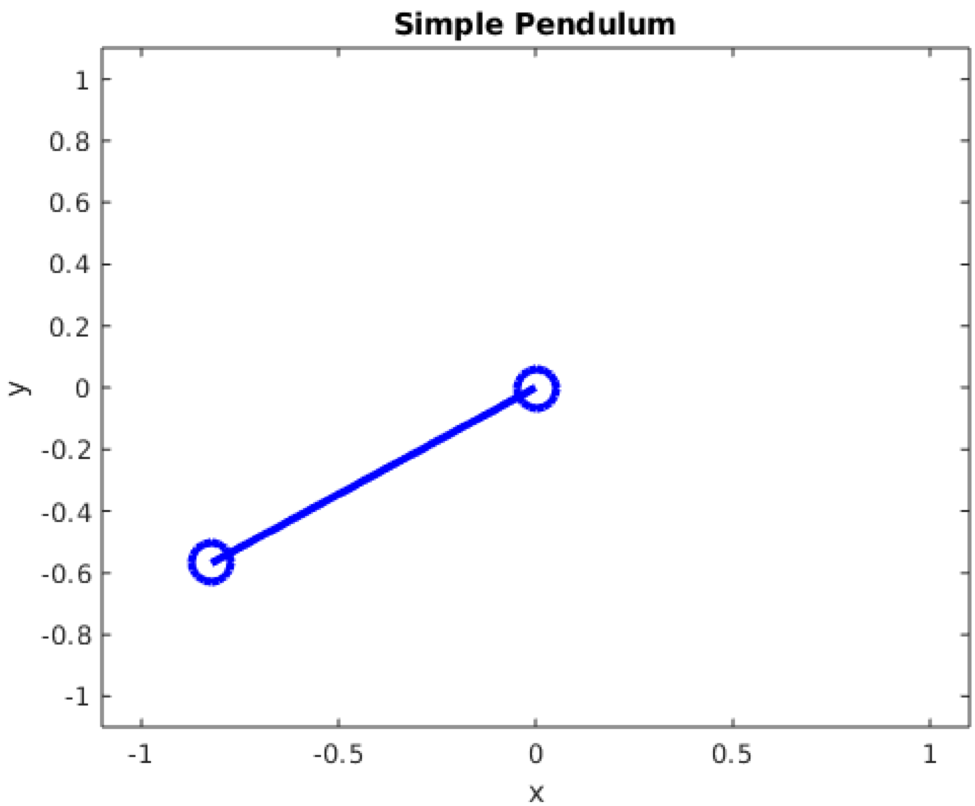
```
% Animate the solutions to questions one and two

clear

omega_0 = 1;
theta_0 = 1; % Initial angular displacement

t_range = [0 60]; % Set domain
IC = [-theta_0 0]; % Set initial conditions

% Question one:
[~, theta] = ode45(@(t,y) [y(2); -omega_0^2.*sin(y(1))], t_range,
 IC); % Numerically solve the system from question one

figure(6)
```

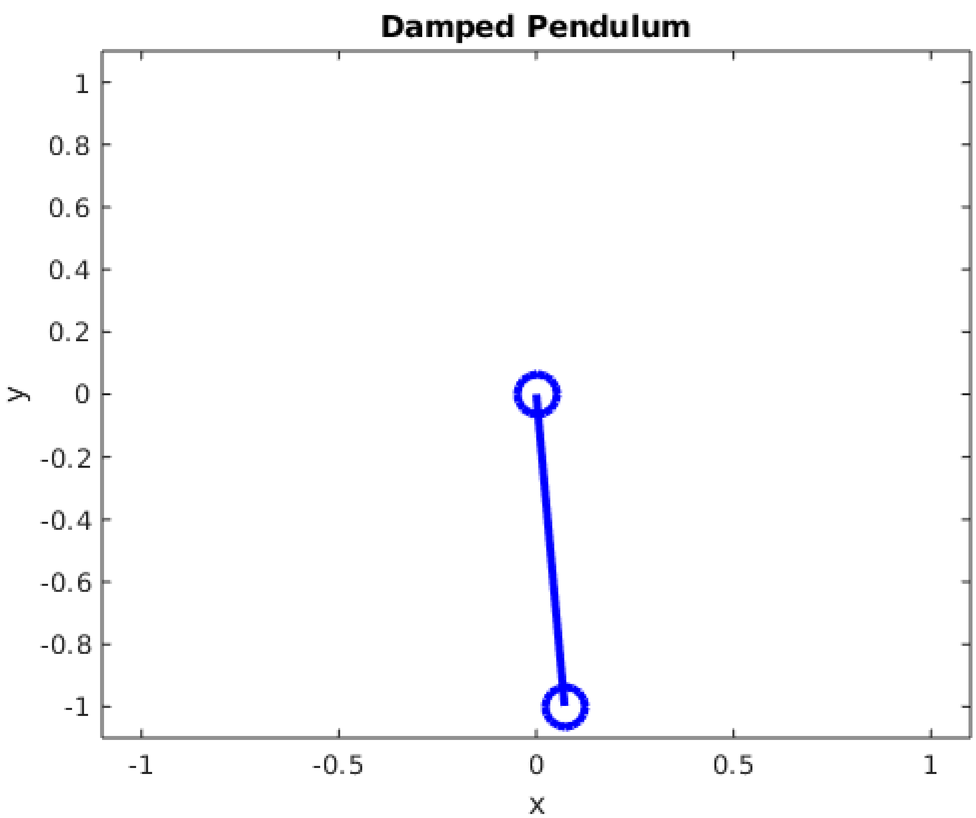
```
animate(sin(theta(:,1)), -cos(theta(:,1)), 'Simple Pendulum') %
 Animate the solution


% Question two:
gamma = 1/2; % Damping coefficient

[~, theta] = ode45(@(t,y) [y(2); -omega_0^2.*sin(y(1)) - gamma *
 abs(y(2)) * y(2)], t_range, IC); % Numerically solve the system from
 question two

figure(7)
animate(sin(theta(:,1)), -cos(theta(:,1)), 'Damped Pendulum') %
 Animate the solution

%close all
%clear
```

Animate.m

```
% Brady Metherall
% Create an animated plot given x and y coordinates

function animate(x, y, plot_title)
    % animate takes two vectors as input, x and y coordinates
    % animate is a void function, nothing is returned
```

```
    % Plot axes
    hold off
    plot([],[],'bo','MarkerSize',15,'LineWidth',3);
    axis([-1.1 1.1 -1.1 1.1]); % Set plot range to be -1 to 1 in both
 x and y (with some tolerance)
    xlabel('x');
    ylabel('y');

    % Animate solution
    for i = 1:length(x) % Loop through each element of the vectors x
 and y
        if (i == 1)
            h = plot([0 x(i)],[0
 y(i)],'bo-','MarkerSize',15,'LineWidth',3);
            axis([-1.1 1.1 -1.1 1.1]); % Set plot range to be -1 to 1
 in both x and y (with some tolerance)
            xlabel('x');
            ylabel('y');
            if nargin < 3 % If the title is not passed from the
 function, assign a default value
                plot_title = 'Animation';
            end
            title(plot_title);
        else
            set(h,'XData',[0 x(i)],'YData',[0 y(i)]);
        end
        pause(0.02);
    end
end
```


Simple Pendulum
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
y
-1
-0.5
0
0.5
1
x


Damped Pendulum
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
y
-1
-0.5
0
0.5
1
x

Published with MATLAB® R2019b