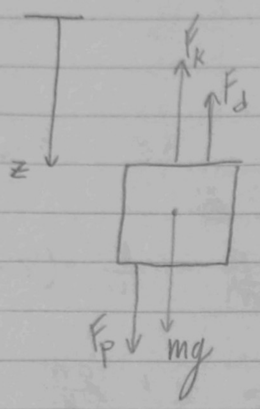
HW 7

1)



$F_k = -k(z - z_0)$, $z_0 \equiv$ uncompressed spring length
$F_d = -d\dot{z}$
$F_p(t)$ = forcing term

$g = \|\vec{g}\| = 9.8$

2) $\Sigma F = \boxed{m\ddot{z} = mg + F_p - d\dot{z} - k(z - z_0)}$

3) spring with mass at rest.

$\dot{z} = 0$, $F_p = 0$, $\Sigma F = 0$

$0 = mg - k(z_e - z_0)$, $mg = k(z_e - z_0)$, $\frac{mg}{k} = z_e - z_0$

$\boxed{z_e = \frac{mg}{k} + z_0}$

4) $\underline{z} = \begin{pmatrix} z \\ \dot{z} \end{pmatrix}$, $\dot{\underline{z}} = \begin{pmatrix} \dot{z} \\ \ddot{z} \end{pmatrix} = \boxed{g(z,u) = \begin{pmatrix} \dot{z} \\ g + \frac{u}{m} - \frac{d\dot{z}}{m} - \frac{k}{m}(z - z_0) \end{pmatrix}}$

5) $\dot{\underline{z}}^* = \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \dot{z} \\ g + \frac{u}{m} - \frac{d\dot{z}}{m} - \frac{k}{m}(z^* - z_0) \end{pmatrix}$, $\dot{z} = 0$, $u = 0$,

$\dot{z}^* = \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ g - \frac{k}{m}(z^* - z_0) \end{pmatrix}$, $g = \frac{k}{m}(z^* - z_0)$, $\frac{gm}{k} = (z^* - z_0)$

$\boxed{\frac{gm}{k} + z_0 = z^* = z_e}$, it makes sense that $z_e = z^*$, z_e is defined as the location where $\Sigma F = 0$ and this is also a requirement for equalibrium

6) let $x = z - z^*$, $z = x + z^*$

$\dot{x} = \dot{z}$, $\ddot{x} = \ddot{z}$

let $\underline{x} = \begin{pmatrix} x \\ \dot{x} \end{pmatrix}$, $\dot{\underline{x}} = \begin{pmatrix} \dot{x} \\ \ddot{x} \end{pmatrix} = \begin{pmatrix} \dot{z} \\ g - \frac{d\dot{z}}{m} - \frac{k}{m}(z - z_0) \end{pmatrix}$

$z - z_0 = x + z^* - z_0 = x + \left(\frac{gm}{k} + z_0\right) - z_0 = x + \frac{gm}{k}$

$\dot{\underline{x}} = \begin{pmatrix} \dot{x} \\ g - \frac{d\dot{x}}{m} - \frac{k}{m}\left(x + \frac{gm}{k}\right) \end{pmatrix} = \begin{pmatrix} \dot{x} \\ g - \frac{d\dot{x}}{m} - \frac{k}{m}x - g \end{pmatrix}$

$$\boxed{\dot{\underline{x}} = \begin{pmatrix} \dot{x} \\ -\frac{d\dot{x}}{m} - \frac{kx}{m} \end{pmatrix}}$$

7) $V = KE + PE$, $KE = \frac{1}{2} m \dot{x}^2$, $PE = \frac{1}{2} k x^2$

$V = \frac{1}{2} m \dot{x}^2 + \frac{1}{2} k x^2$

$V(\underline{0}) = 0$, $\nabla V = \begin{pmatrix} kx \\ m\dot{x} \end{pmatrix}$, $\dot{V} = [\nabla V]^T \dot{\underline{x}} =$

$kx\dot{x} + m\dot{x}\left(-\frac{d\dot{x}}{m} - \frac{kx}{m}\right) = kx\dot{x} - d\dot{x}\dot{x} - kx\dot{x} = -d\dot{x}\dot{x}$

assuming $d \geq 0$, $\dot{V} \leq 0$, so we know the system is stable, However since $\dot{V} = 0$ for $x = 0$ or $\dot{x} = 0$, Lyapunovs function cannot prove aysmptotic stability

8) $\underline{A} = \begin{bmatrix} \frac{\partial \dot{x}_1}{\partial x_1} & \frac{\partial \dot{x}_1}{\partial x_2} \\ \frac{\partial \dot{x}_2}{\partial x_1} & \frac{\partial \dot{x}_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} \frac{\partial \dot{x}}{\partial x} & \frac{\partial \dot{x}}{\partial \dot{x}} \\ \frac{\partial \ddot{x}}{\partial x} & \frac{\partial \ddot{x}}{\partial x} \end{bmatrix}$

$= \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{d}{m} \end{bmatrix}$

our system is linear already since $\underline{A}\,\underline{x} = \dot{\underline{x}}$ for all $\underline{x}$ so any conclusions we draw will be global.

$A - \lambda I = \begin{vmatrix} -\lambda & 1 \\ -\frac{k}{m} & -\frac{d}{m} - \lambda \end{vmatrix} = \frac{d\lambda}{m} + \lambda^2 + \frac{k}{m}$

$\lambda = \dfrac{-\frac{d}{m} \pm \sqrt{\frac{d}{m}^2 - 4\frac{k}{m}}}{2}$ let $b = \frac{d}{m}$, $c = \frac{k}{m}$

if $b^2 - 4c < 0$

$\text{real}(\lambda) = -\frac{1}{2}b < 0$

if $b^2 - 4c = 0$

$\lambda = -\frac{1}{2}b < 0$

if $b^2 - 4c > 0$

$b^2 - 4c < b^2$

so $-b - \sqrt{b^2 - 4c} < 0$

So in any case, $\lambda < 0$ and we can conclude global aysmptotic stability

Table of Contents

clean up

```
clear all
close all
clc
```

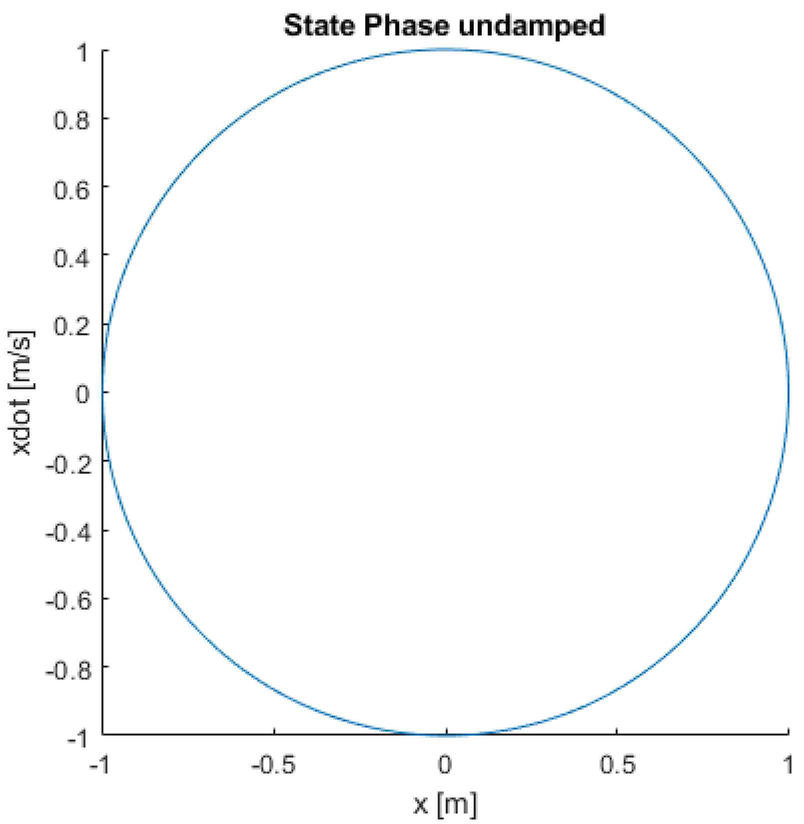
Solve ODE

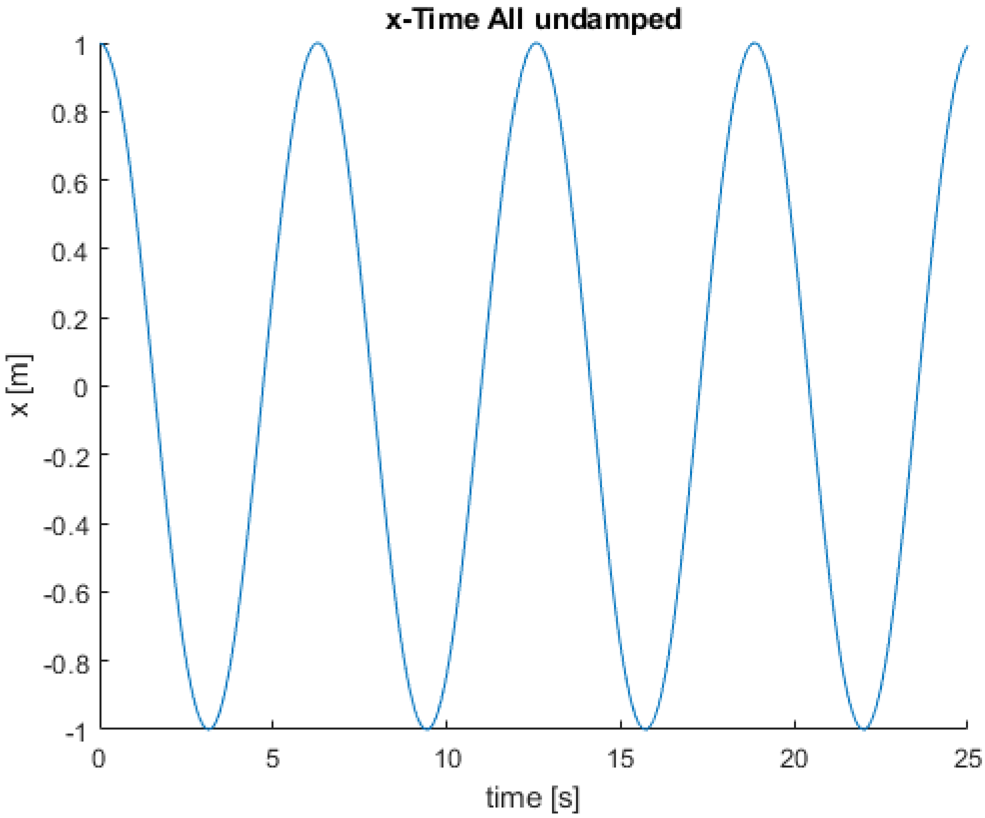
```
options = odeset('RelTol', 1e-8,'AbsTol',1e-8);


tspan = [0,25];
X0 = [1;0];
w=1;

runs = {0;.25;1;2};
n = length(runs);
runs = cat(2,cell(4,1),runs);
runs = cat(2,cell(4,1),runs);
runs = cat(2,runs,{"undamped";"underdamped";"critically
 damped";"overdamped"});

for i = 1:n
    z = runs{i,3};
    [t,X]=ode45(@eom,tspan,X0,options,z,w);
    runs{i,1} = X;
    runs{i,2} = t;
end
```

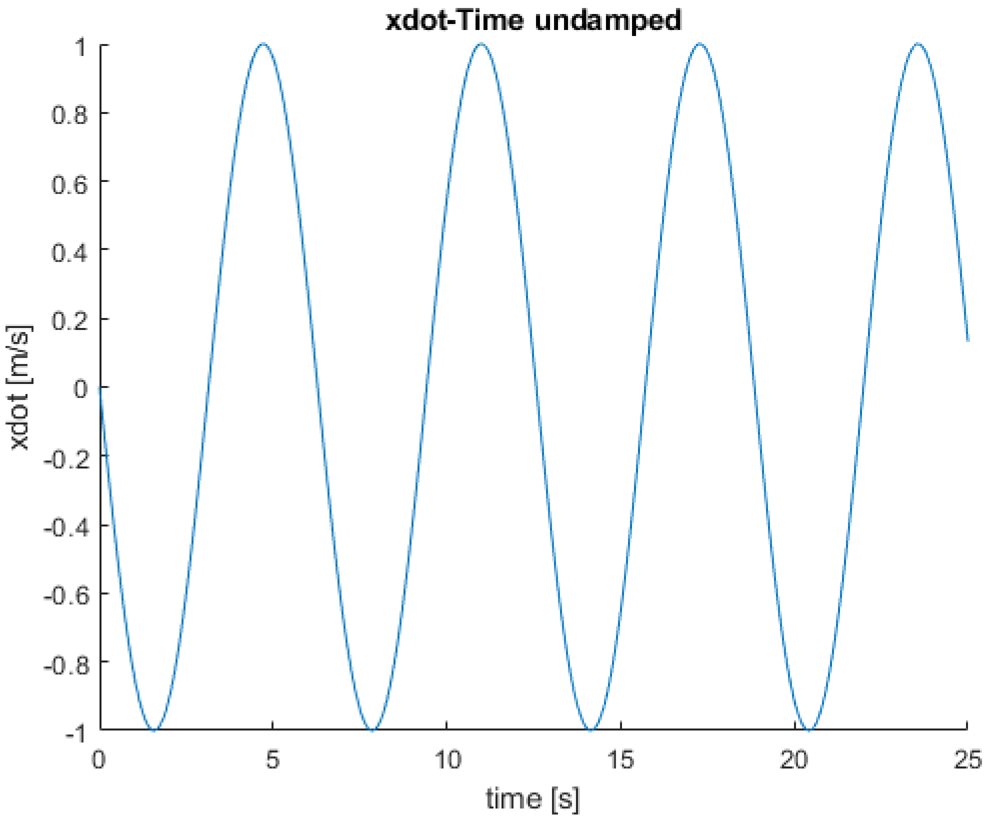
plot each

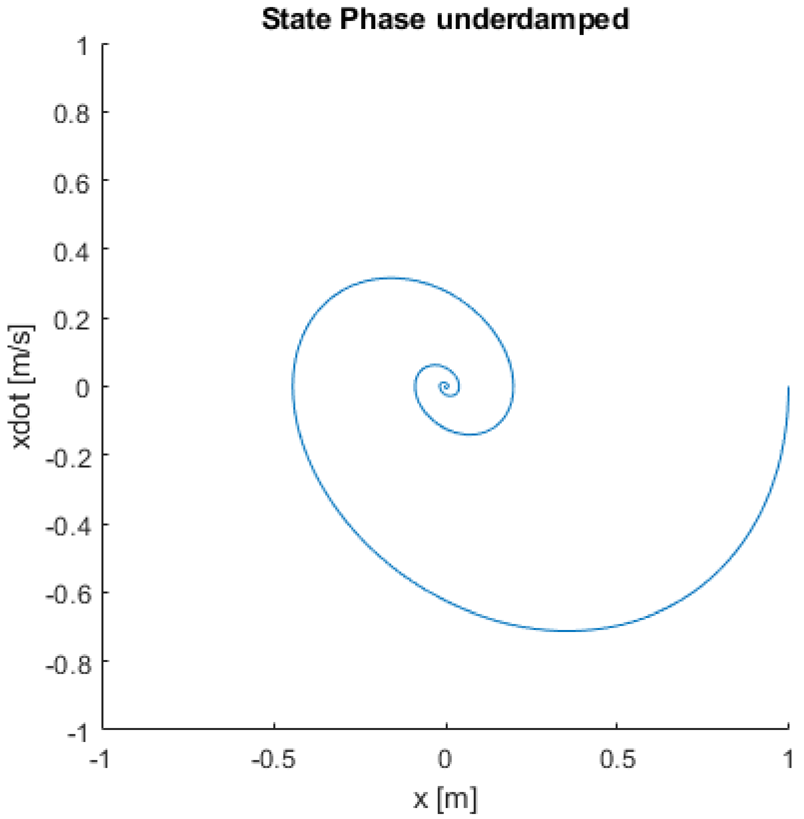
```
for i = 1:n
    figure
    axe1 = axes;
    title("State Phase "+string(runs{i,4}))
    xlabel("x [m]")
    ylabel("xdot [m/s]")
    axis('equal')
    ylim([-1,1])
    xlim([-1,1])
```

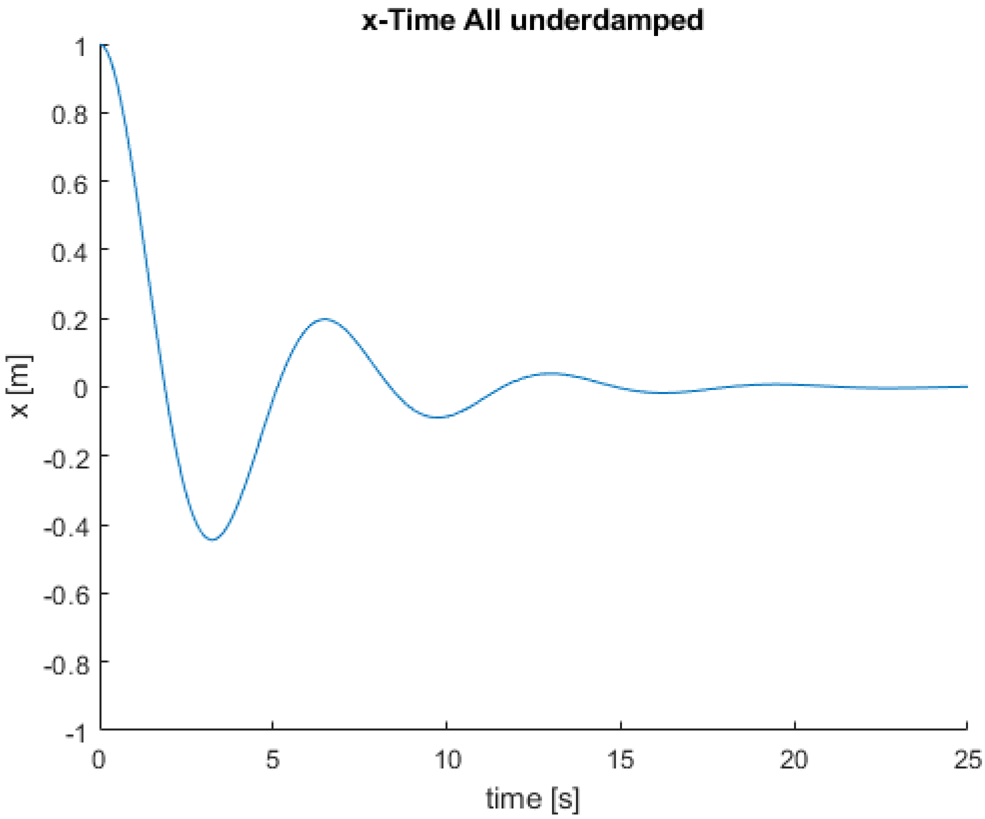
```
    hold on


    figure
    axe2 = axes;
    title("x-Time All "+string(runs{i,4}))
    xlabel("time [s]")
    ylabel("x [m]")
    ylim([-1,1])
    hold on

    figure
    axe3 = axes;
    title("xdot-Time "+string(runs{i,4}))
    xlabel("time [s]")
    ylabel("xdot [m/s]")
    ylim([-1,1])
    hold on

    plot(axe1,runs{i,1}(:,1),runs{i,1}(:,2))
    plot(axe2,runs{i,2},runs{i,1}(:,1))
    plot(axe3,runs{i,2},runs{i,1}(:,2))
%     title(axe1,runs{i,4})
%     title(axe2,runs{i,4})
%     title(axe3,runs{i,4})
end
```

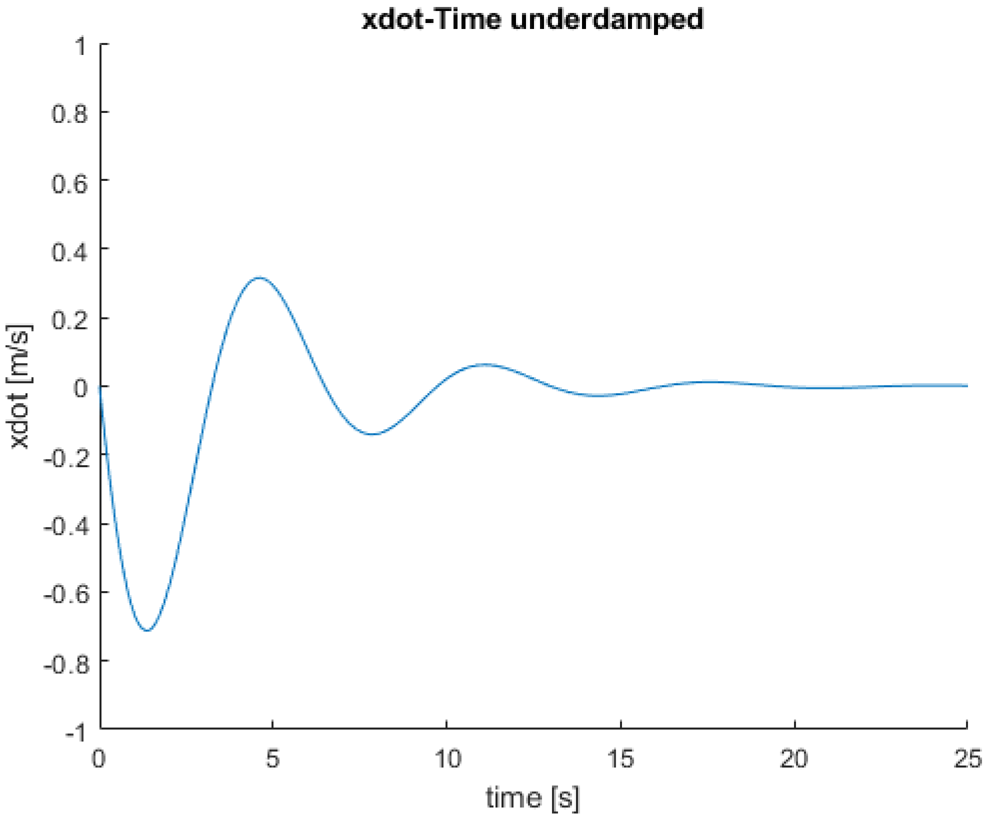


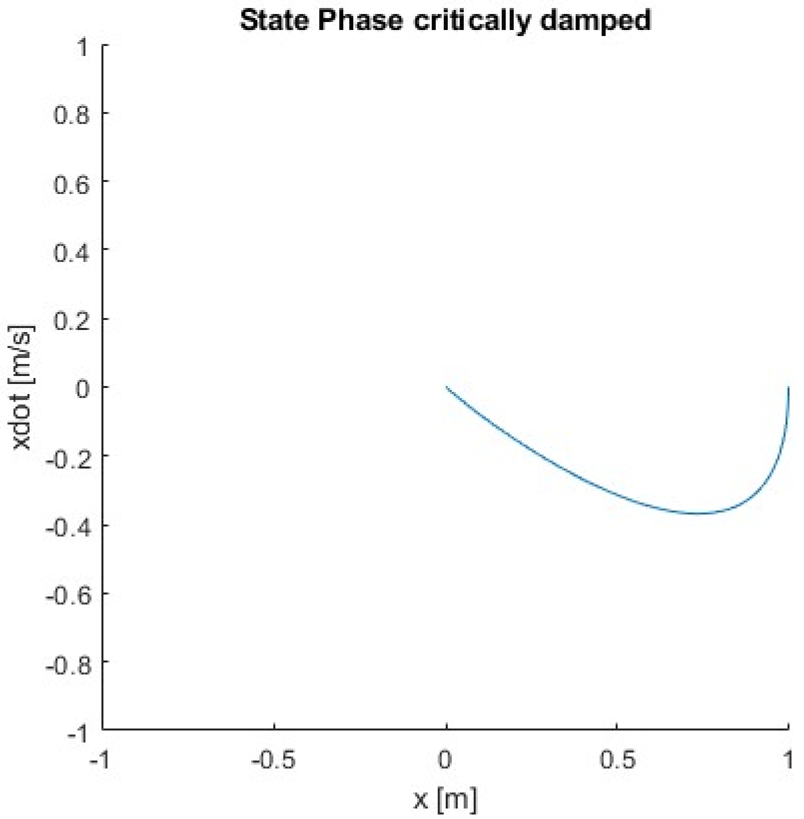

x-Time All undamped
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
x [m]
0
5
10
15
20
25
time [s]

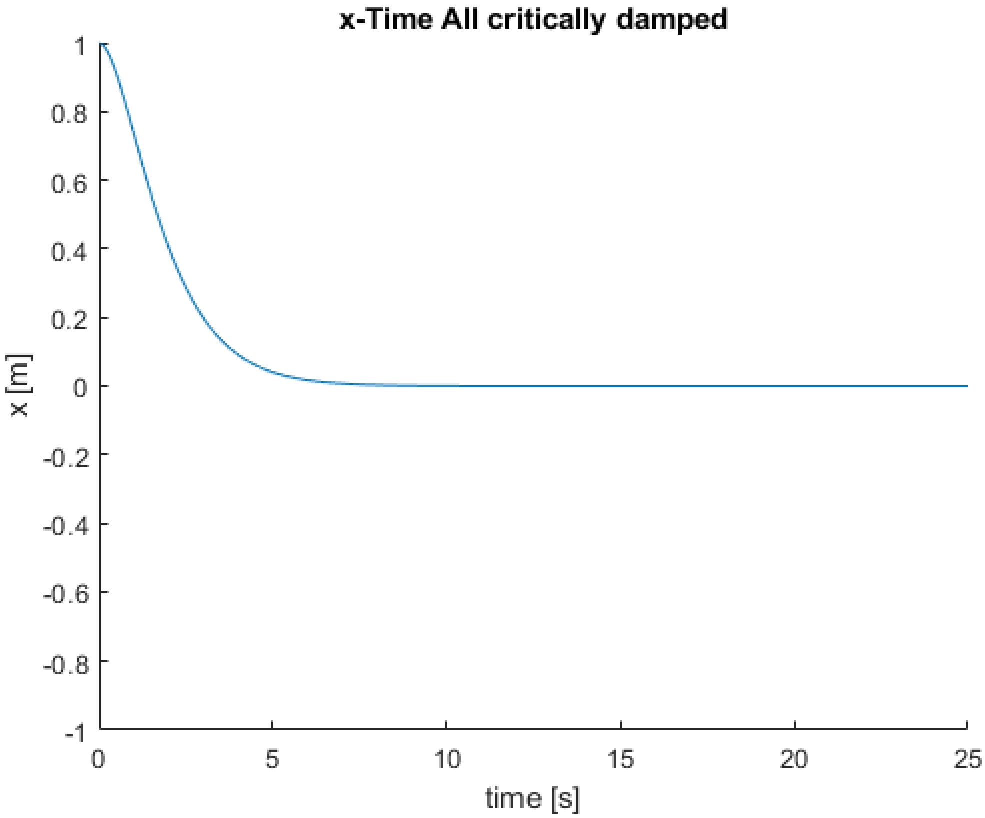

xdot-Time undamped
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
xdot [m/s]
0
5
10
15
20
25
time [s]

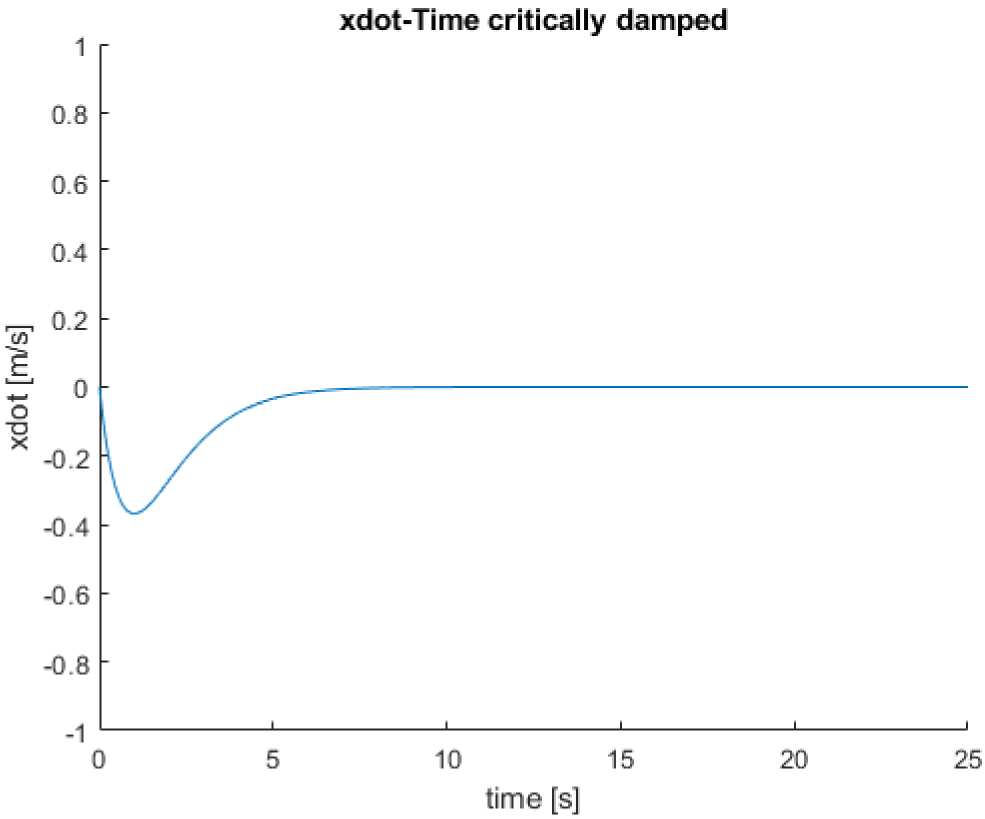

State Phase underdamped
xdot [m/s]
x [m]
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
1

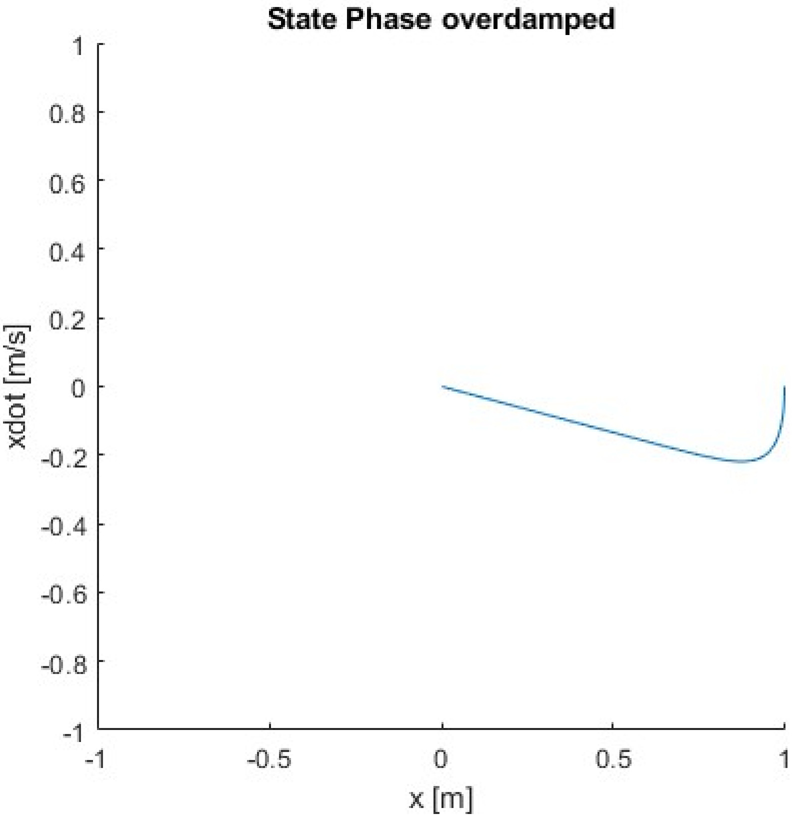

x-Time All underdamped
x [m]
time [s]
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
5
10
15
20
25

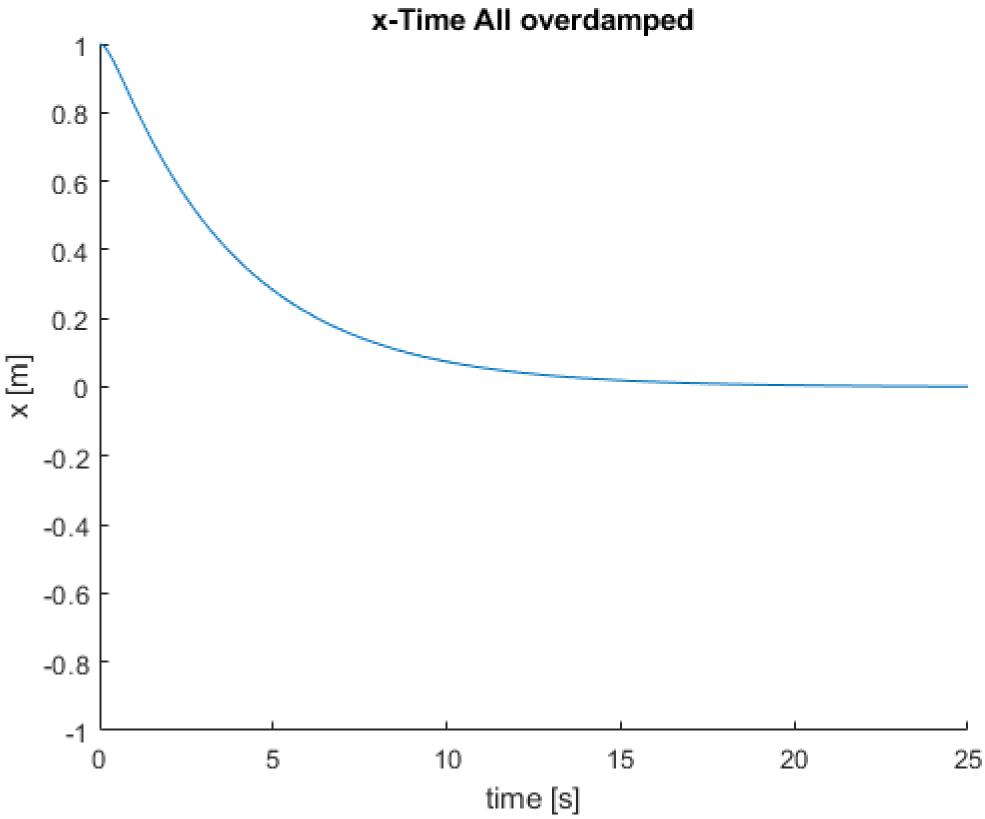

xdot-Time underdamped
xdot [m/s]
time [s]
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
5
10
15
20
25

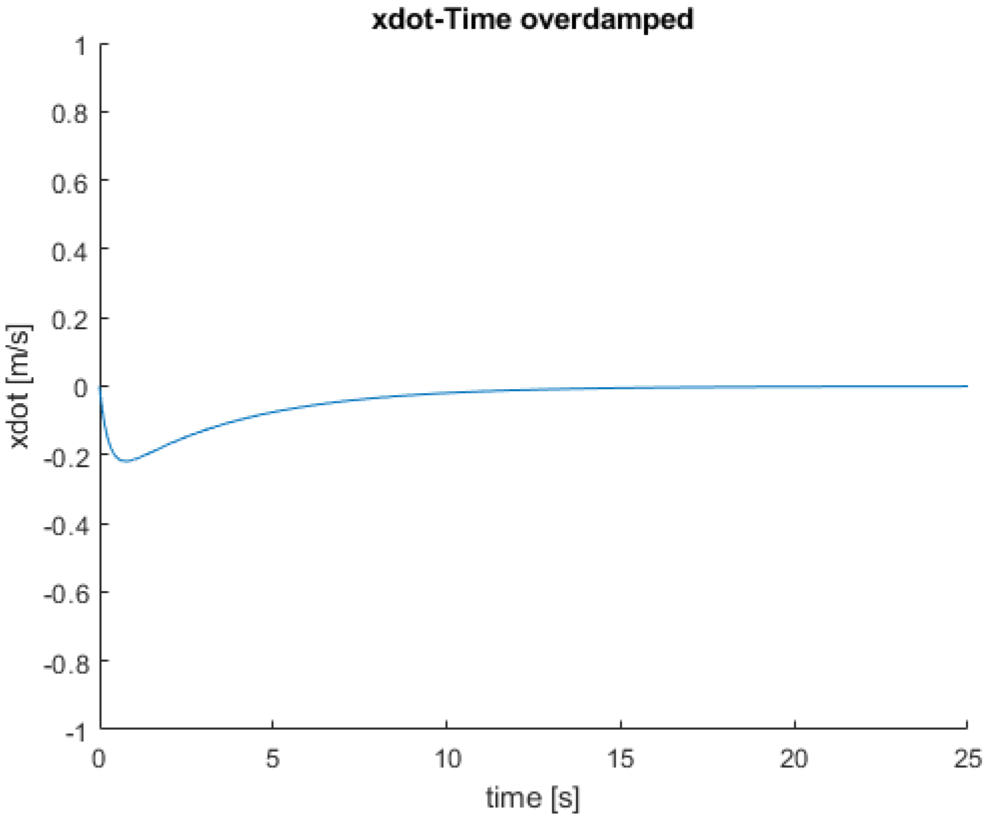

State Phase critically damped
xdot [m/s]
x [m]
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
1


x-Time All critically damped
x [m]
time [s]
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
5
10
15
20
25


xdot-Time critically damped
xdot [m/s]
time [s]
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
5
10
15
20
25


State Phase overdamped
xdot [m/s]
x [m]
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
1

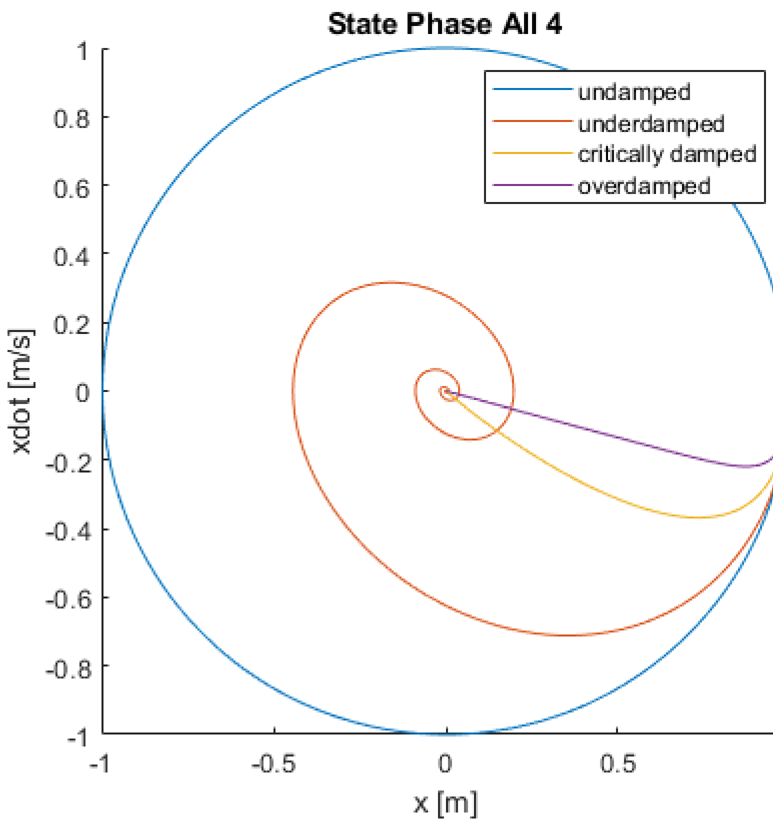

x-Time All overdamped
x [m]
time [s]
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
5
10
15
20
25

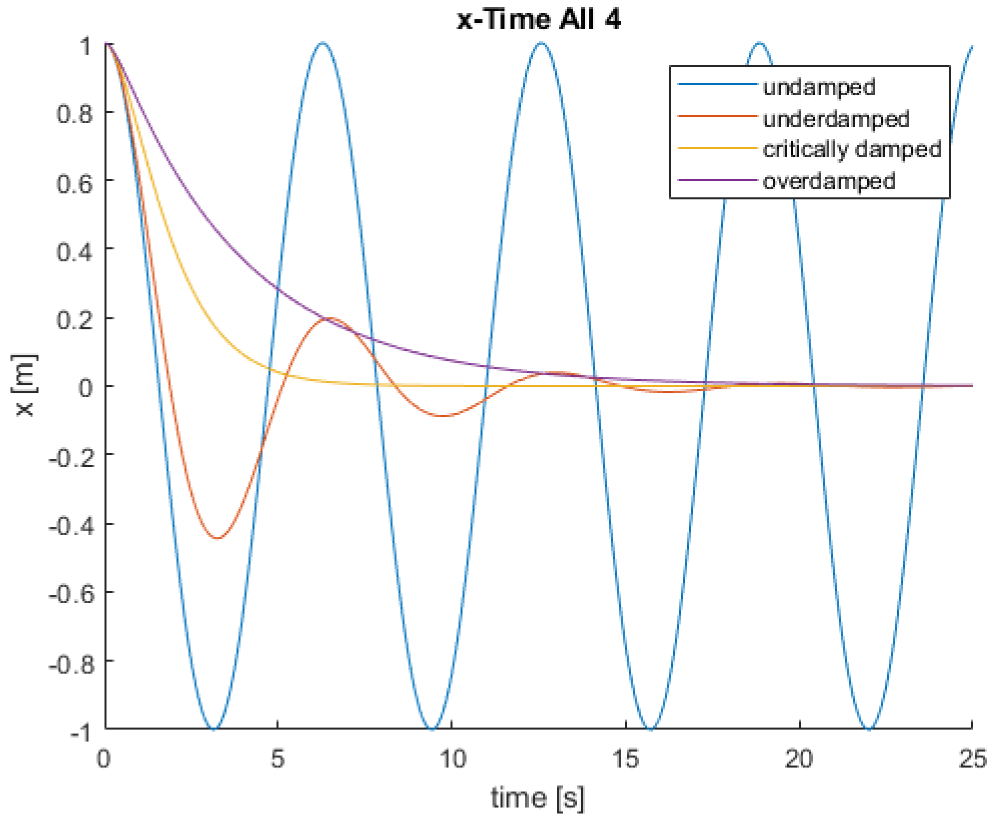


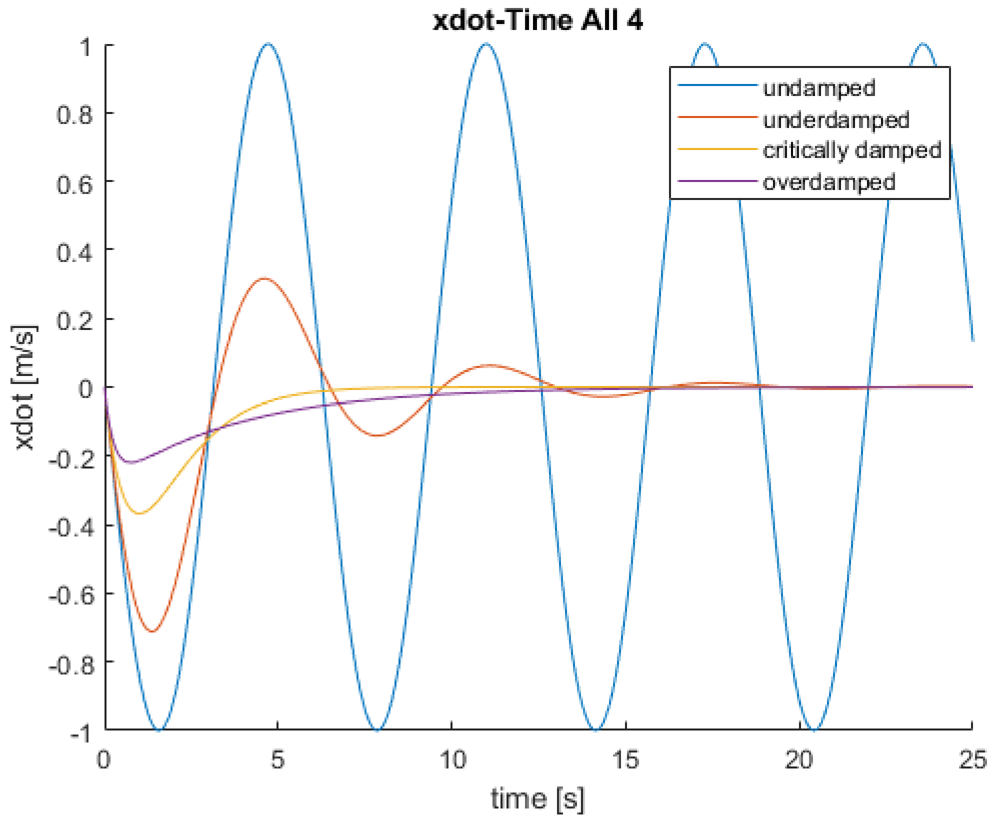
plot all 4

```
figure
axe1 = axes;
title("State Phase All 4")
xlabel("x [m]")
ylabel("xdot [m/s]")
axis('equal')
ylim([-1,1])
xlim([-1,1])
hold on


figure
axe2 = axes;
title("x-Time All 4")
xlabel("time [s]")
ylabel("x [m]")
ylim([-1,1])
hold on

figure
axe3 = axes;
title("xdot-Time All 4")
xlabel("time [s]")
```

```
ylabel("xdot [m/s]")
ylim([-1,1])
hold on

for i = 1:n
    plot(axe1,runs{i,1}(:,1),runs{i,1}(:,2))
    plot(axe2,runs{i,2},runs{i,1}(:,1))
    plot(axe3,runs{i,2},runs{i,1}(:,2))
end

legend(axe1,runs{:,4})
legend(axe2,runs{:,4})
legend(axe3,runs{:,4})
```




x-Time All 4
x [m]
time [s]
undamped
underdamped
critically damped
overdamped
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
5
10
15
20
25


xdot-Time All 4
xdot [m/s]
time [s]
undamped
underdamped
critically damped
overdamped
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
5
10
15
20
25

functions

```
function Xdot = eom(t,X,z,w)
% d/m = 2wz
% sqrt(k/m) = w
Xdot1 = X(2);
%      Xdot2 = -(d/m)*X(2)-(k/m)*X(1);
Xdot2 = -(2*w*z)*X(2)-(w^2)*X(1);
Xdot = [Xdot1;Xdot2];
end
```

Published with MATLAB® R2022a

11. From the x-time graph you can tell that the critically damped scenario is the fasted to converge on the equilibrium point. This may be useful for applications like car shock absorbers.