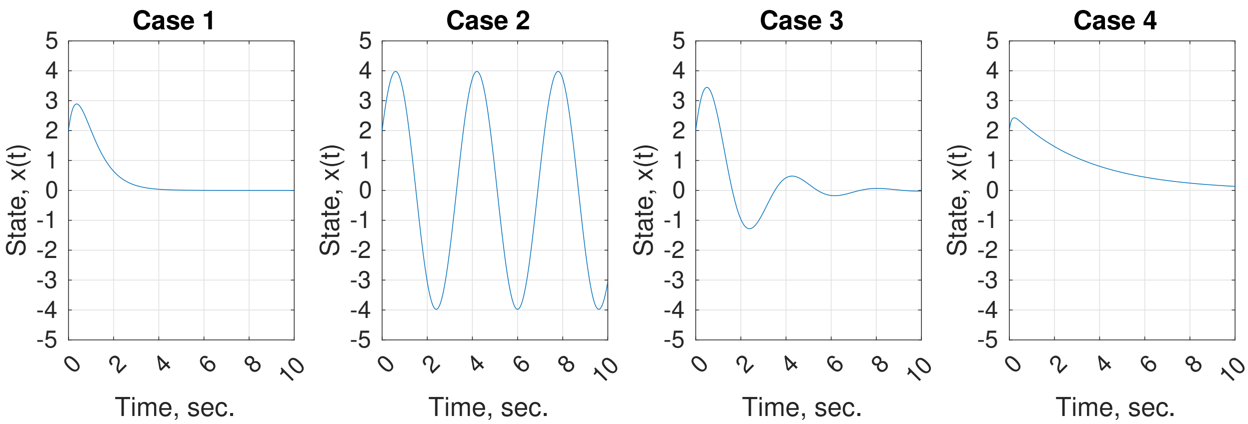
Homework 5

1 Problem



Match each of the resposes above to one of the following second-order system types:

- undamped
- underdamped
- critically damped
- overdamped

Solution

- undamped: Case 2
- underdamped: Case 3
- critically damped: Case 1
- overdamped: Case 4

2 Problem

Consider the following transfer function:

$$G(s) = \frac{X(s)}{U(s)} = \frac{100}{s^2 + 5s + 100} \, .$$

By comparing the characteristic equation in the denominator of $G(s)$ to that of a damped harmonic oscillator

$$G_{\text{DHO}}(s) = \frac{1}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

determine:

- the undamped natural frequency of the system (in both rad/s and Hz)
- the damping ratio of the system
- the damped natural frequency of the system (in both rad/s and Hz)

Solution

$$G(s) = \frac{100}{s^2+5s+100} \leftarrow \text{characteristic eq.}$$

Equate characteristic equation of G(s) to that of a damped harmonic oscillator

$$s^2+5s+100 = s^2+2\zeta\omega_n s+\omega_n^2$$

$$s: \quad 5 = 2\zeta\omega_n$$

$$\text{const} \quad 100 = \omega_n^2 \Rightarrow \boxed{\omega_n = 10} \ \frac{rad}{s} \text{ or } \boxed{1.59\ Hz}$$

$$\text{thus } \zeta = \frac{5}{2\cdot 10}$$

$$\boxed{\zeta = 0.25}$$

$$\omega_d = \omega_n\sqrt{1-\zeta^2}$$

$$= 10\sqrt{1-\frac{1}{16}}$$

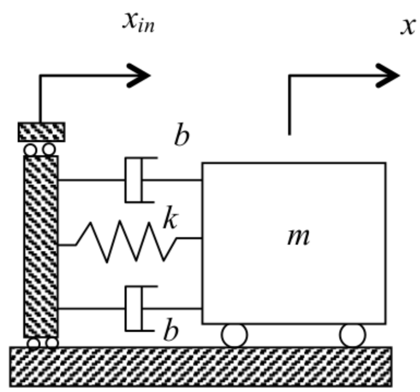
$$\frac{rad}{s}\cdot\frac{1\ cycle}{2\pi\ rad} = \frac{cylce}{s} = Hz$$

$$= 10\frac{\sqrt{15}}{\sqrt{16}}$$

$$= \frac{5\sqrt{15}}{2} \approx \boxed{9.6825\ rad/s} = \boxed{1.54\ Hz}$$

3 Problem

Consider the following model of a mechanical system:
The system has a mass m, a linear spring with stiffness k, and two identical dampers with damping constant b. The left wall generates an input motion $x_{in}(t)$ that causes the mass to



undergo a displacement $x(t)$ from its equilibrium position. The initial position and velocity are zero.

- Find the ODE describing the motion of the system by drawing a free-body diagram and applying Newton's 2nd Law. Your answer should be in terms of the following variables:

$$x_{in}, \dot{x}_{in}, x, \dot{x}, \ddot{x}, b, k, m$$

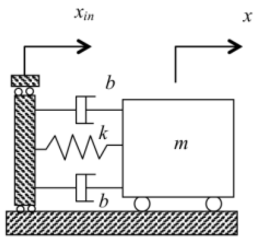
- Show that the transfer function is

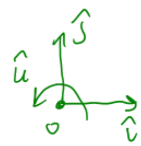
$$G(s) = \frac{X(s)}{X_{\text{in}}(s)} = \frac{\frac{2b}{m}s + \frac{k}{m}}{s^2 + \frac{2b}{m}s + \frac{k}{m}}$$

When taking the Laplace transform $\mathcal{L}[\dot{x}_{in}(t)]$ you may assume the initial (input) condition $x_{in}(0) = 0$.

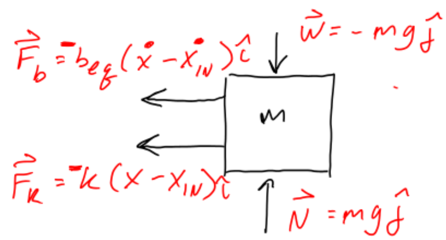
Solution

Since the two dampers are in parallel the equivalent damper is $b_{eq} = 2b$





Free body diagram



Newton's 2nd Law

$$\hat{i}: \quad \sum \vec{F} = -\underbrace{b_{eq}}_{=2b}(\dot{x} - \dot{x}_{IN}) - k(x - x)\hat{i} = m\ddot{x}$$

$$\ddot{x} + \frac{2b}{m}(\dot{x} - \dot{x}_{IN}) + \frac{k}{m}(x - x_{IN}) = 0$$

Laplace Transform

$$\mathcal{L}[\ddot{x}] + \frac{2b}{m}\mathcal{L}[\dot{x}] - \frac{2b}{m}\mathcal{L}[\dot{x}_{IN}(t)] + \frac{k}{m}\mathcal{L}[x] - \frac{k}{m}\mathcal{L}[x_{IN}(t)] = 0$$

Assume $x_{IN}(0) = 0$

Assume I.Cs are zero

$$s^2 X(s) + \frac{2b}{m} s X(s) - \frac{2b}{m} s X_{IN}(s) + \frac{k}{m} X(s) - \frac{k}{m} X_{IN}(s) = 0$$

$$X(s)\left(s^2 + \frac{2b}{m}s + \frac{k}{m}\right) - X_{IN}(s)\left(\frac{2b}{m}s + \frac{k}{m}\right) = 0$$

Transfer Func:

$$\frac{X(s)}{X_{IN}(s)} = \frac{\left(\frac{2b}{m}s + \frac{k}{m}\right)}{\left(s^2 + \frac{2b}{m}s + \frac{k}{m}\right)}$$

Plugging in $m = 1$, $b = 4$, $k = 97$

$$\boxed{\frac{X(s)}{X_{IN}(s)} = \frac{8s + 97}{s^2 + 8s + 97}}$$

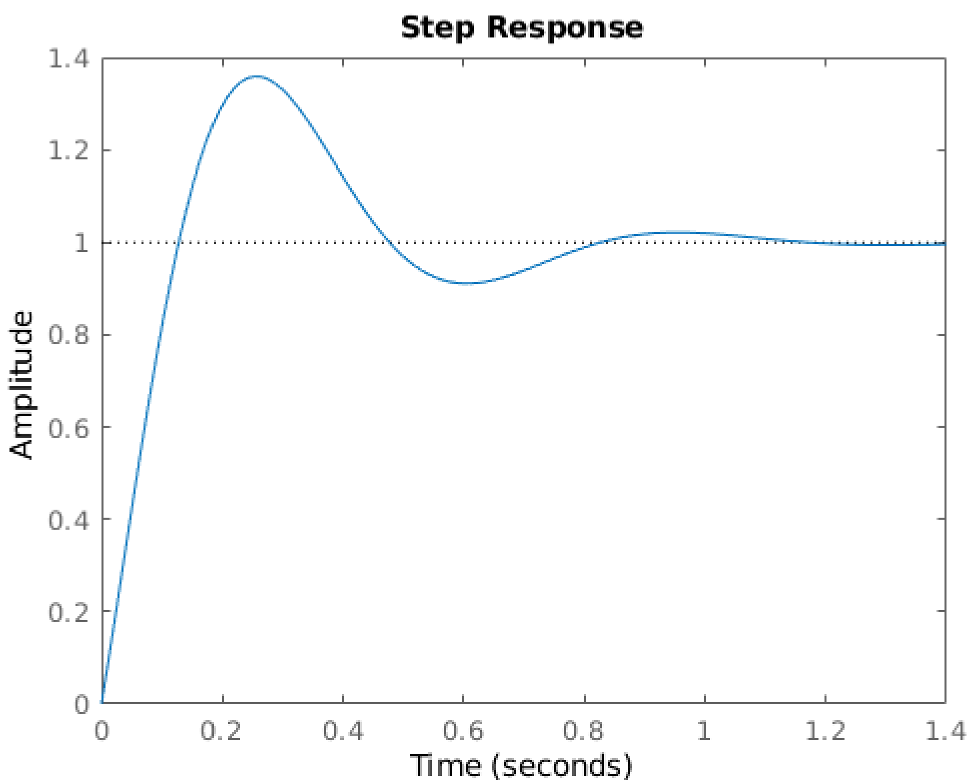
4 Problem

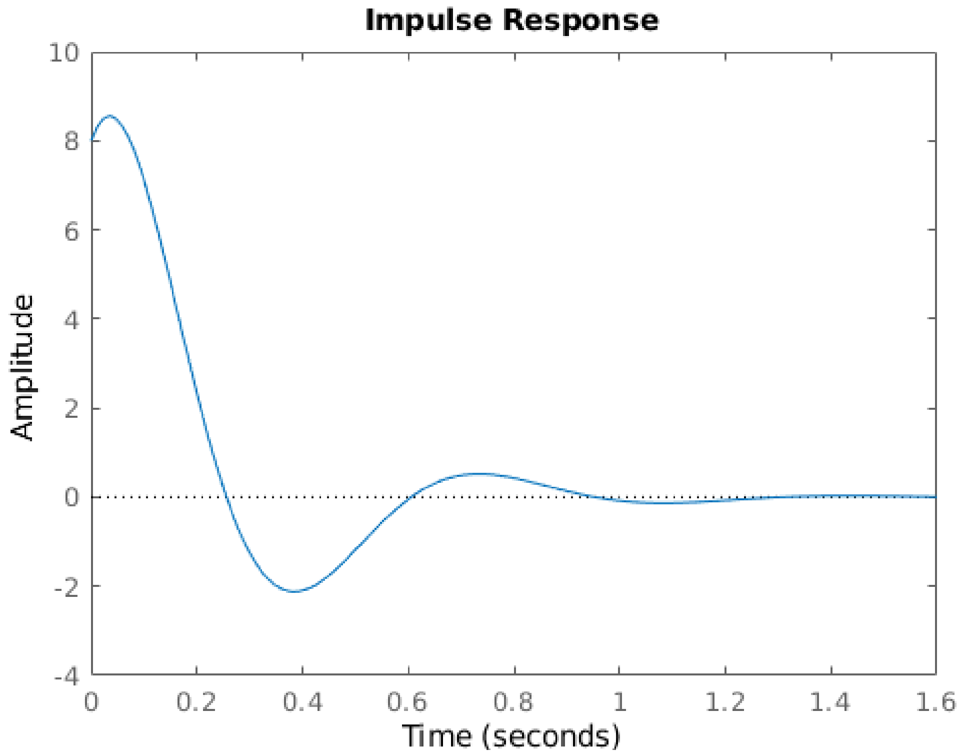
Define the transfer function from the previous problem in MATLAB using the `tf` command. The values of the constants are: $m = 1$ kg, $k = 97$ N/m, and $b = 4$ N-s/m.

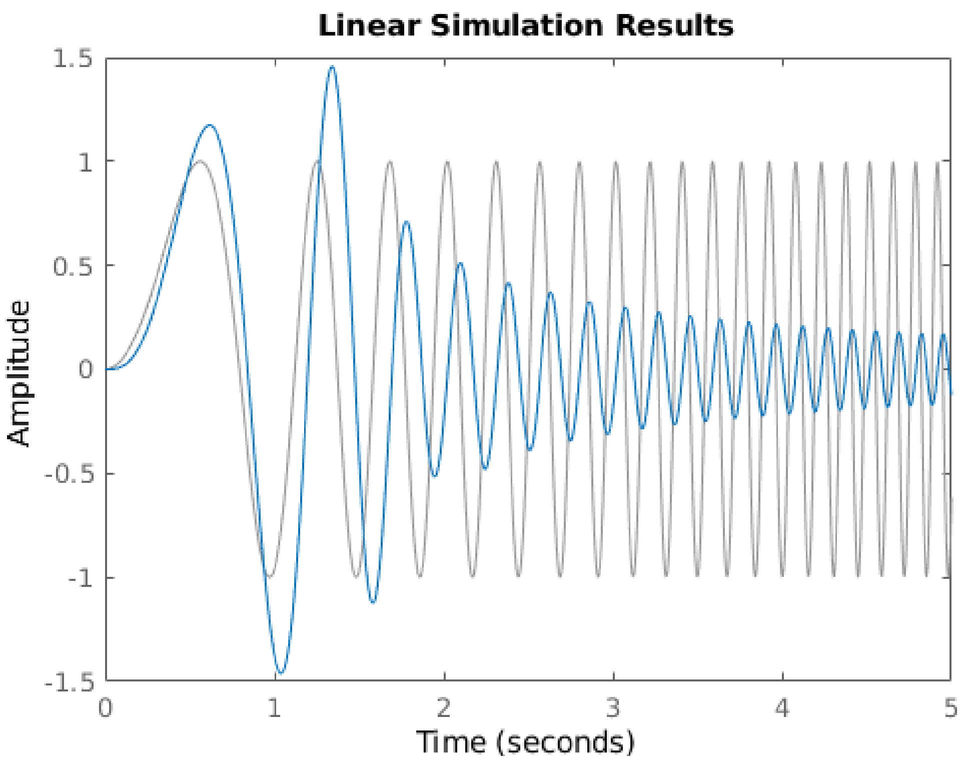
- Suppose the input is a unit step, $x_{in}(t) = u(t)$. Use your transfer function $G(s)$ to plot the response in MATLAB.
- Suppose the input is an impulse, $x_{in}(t) = \delta(t)$. Use your transfer function $G(s)$ to plot the response in MATLAB.
- Suppose the input is a chirp (a sinusoid of increasing frequency), $x_{in}(t) = \sin(5t^2)$. Use your transfer function $G(s)$ to plot the response in MATLAB from time 0 to 5 seconds. To ensure the input chirp is defined with sufficient resolution, use at least 1,000 time steps to represent the time interval requested.

Solution

```
clear; close all; clc; % prepare workspace

num = [8 97];
den = [1 8 97];
sys = tf(num,den);

figure;
step(sys)

figure;
impulse(sys)

figure;
t = linspace(0,5,1E3);
f = sin(5*t.^2);
lsim(sys,f,t)
```




Impulse Response
10
8
6
4
2
0
-2
-4
Amplitude
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
Time (seconds)


Linear Simulation Results
1.5
1
0.5
0
-0.5
-1
-1.5
Amplitude
0
1
2
3
4
5
Time (seconds)

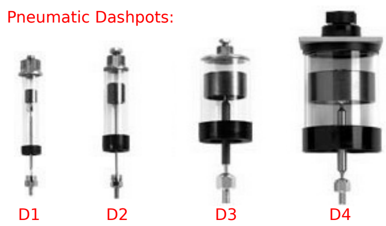
Published with MATLAB® R2020a

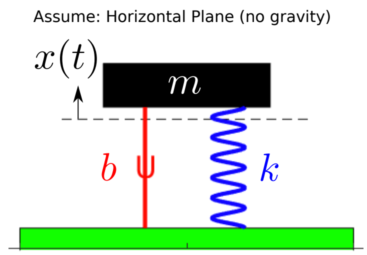
5 Problem

Consider the damped harmonic oscillator shown below and modeled by the system:

$$\ddot{x} + \left(\frac{b}{m}\right)\dot{x} + \left(\frac{k}{m}\right)x = 0 \tag{1}$$

with mass $m = 1$ kg, spring stiffness $k = 100$ N/m, and initial conditions $x(t_0) = 0.1$ m and $\dot{x}(t_0) = 0$, where $x(t)$ is the displacement of the mass from the nominal (unstretched) spring position. Four dashpots are being considered for this system: D1, D2, D3 and D4 have damping coefficients $b_1 = 3$, $b_2 = 10$, $b_3 = 20$, and $b_4 = 50$ N/(m/s), respectively.





You are tasked with selecting the correct dashpot that renders the system critically damped and to determine the response of the system over a two second interval from the initial condition. To perform your analysis, include the following:

- A *single plot with all five curves* showing the response of the system to the initial conditions for each dashpot over a two second interval, and for the case of no dashpot $b = 0$. Each curve should be plotted with a different color with time on the abscissa and the displacement $x(t)$ on the ordinate axis. Label your axes and include a legend.
- Determine the damping ratio (numerical value) for Case 0 (no dashpot) and for Cases 1-4 with dashpots D1, D2, D3, and D4. For each case specify whether it as critically damped, overdamped, undamped, or underdamped.
- Make your final recommendation on which dashpot to choose

MATLAB Hints (optional):

- To avoid repeating the same code for each of the five cases you are encouraged to encapsulate it in a function of the form below.

```
function xhist = simulateMassSpringDamper(tvals,x0,xdot0,k,m,b)

% simulate system (your code here)

end
```

This function can then be called five times in your main script to simulate each case (with a different input value for `b`).

```
b_vec = [0 3 10 20 50];
% Simulate (changing b value each time)
xhist0 = simulateMassSpringDamper(t,x0,xdot0,k,m,b_vec(1)); %
xhist1 = simulateMassSpringDamper(t,x0,xdot0,k,m,b_vec(2)); %
xhist2 = simulateMassSpringDamper(t,x0,xdot0,k,m,b_vec(3)); %
xhist3 = simulateMassSpringDamper(t,x0,xdot0,k,m,b_vec(4)); %
xhist4 = simulateMassSpringDamper(t,x0,xdot0,k,m,b_vec(5)); %
```

- You can animate/verify your solution by downloading the `playMassSpringDamper.m` file from Canvas, placing it in your working directory, and running the following commands once your simulation is complete:

  ```
  playMassSpringDamper(thist,xhist)
  ```

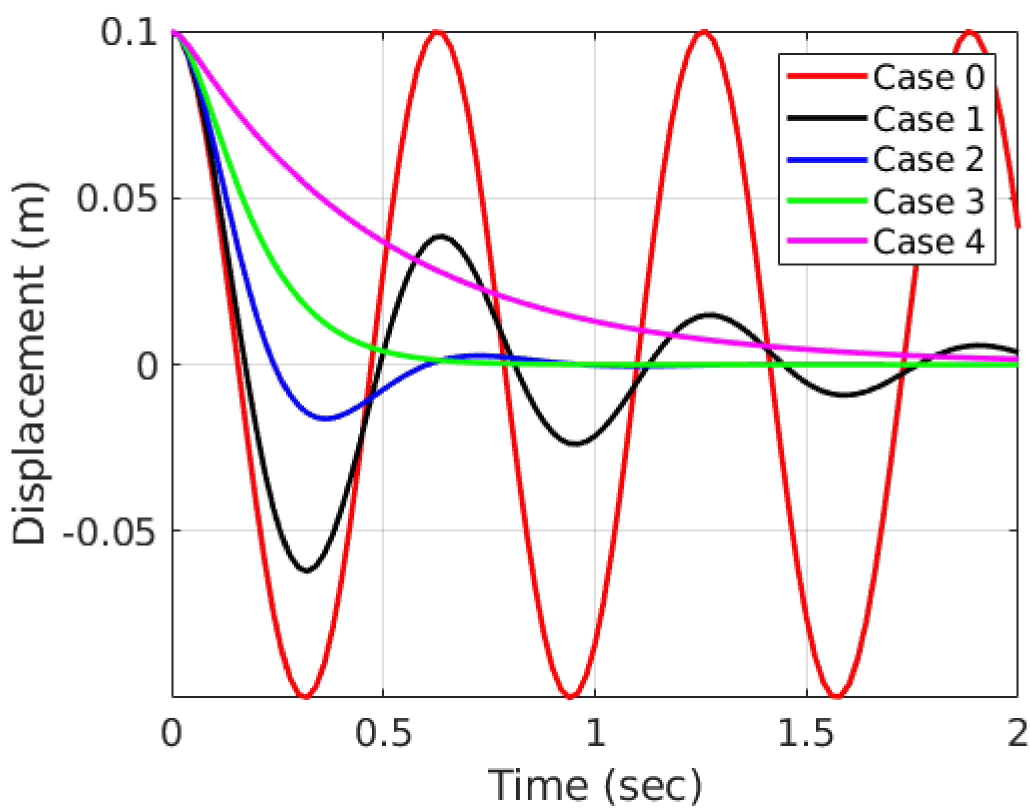
  The above command assumes `xhist` is a `1 x N` vector of displacements, and `thist` is a `1 x N` vector of corresponding times (from 0 to 2 seconds) where a value of around `N=100` was used to produce the animations posted on Canvas.

Solution

```
clear; close all; clc; % prepare workspace

% User Inputs
%  x''  + (b/m)x + (k/m)x'  = f
k = 100; % N/m
m = 1; % kg
x0 = 0.1; % x(0) IC, initial position
xdot0 = 0; % x'(0) IC, initial speed
t = [0:1/50:2]; % simulation time 50 frames per second
b_vec = [0 3 10 20 50];

% Simulate (changing c value each time)
xhist0 = simulateMassSpringDamper(t,x0,xdot0,k,m,b_vec(1)); %
xhist1 = simulateMassSpringDamper(t,x0,xdot0,k,m,b_vec(2)); %
xhist2 = simulateMassSpringDamper(t,x0,xdot0,k,m,b_vec(3)); %
xhist3 = simulateMassSpringDamper(t,x0,xdot0,k,m,b_vec(4)); %
xhist4 = simulateMassSpringDamper(t,x0,xdot0,k,m,b_vec(5)); %


% Calculate natural frequency
omega_n = sqrt(k/m)/(2*pi);
fprintf('The natural frequency and period of the system are %3.3f Hz
 and %3.3f seconds, respectively. \n', omega_n, 1/omega_n);

% Calculate damping ratio
zeta_vec = b_vec/(2*sqrt(k*m));
omegad_vec = omega_n*sqrt(1-zeta_vec.^2);

for i = 1:1:length(b_vec)
fprintf('The damping ratio for Case %d is : %3.3f sec \n', i-1,
 zeta_vec(i));
end
disp('Case 0 has no damping since the damping ratio = 0');
disp('Case 1 is underdamped damped since the damping ratio < 1');
disp('Case 2 is underdamped since the damping ratio <  1');
disp('Case 3 is critically damped since the damping ratio = 1');
disp('Case 4 is overdamped since the damping ratio >  1');


% Plot
figure;
plot(t,xhist0,'r','linewidth',2); hold on;
plot(t,xhist1,'k','linewidth',2)
plot(t,xhist2,'b','linewidth',2)
plot(t,xhist3,'g','linewidth',2)
plot(t,xhist4,'m','linewidth',2)
set(gca,'FontSize',14);
xlabel('Time (sec)')
ylabel('Displacement (m)')
legend('Case 0','Case 1','Case 2','Case 3','Case 4');
grid on;
hold on;
```

```
axis tight;

% % optional:
% close all;
% playMassSpringDamper(t,xhist5)
```

```
The natural frequency and period of the system are 1.592 Hz and 0.628
 seconds, respectively.
The damping ratio for Case 0 is : 0.000 sec
The damping ratio for Case 1 is : 0.150 sec
The damping ratio for Case 2 is : 0.500 sec
The damping ratio for Case 3 is : 1.000 sec
The damping ratio for Case 4 is : 2.500 sec
Case 0 has no damping since the damping ratio = 0
Case 1 is underdamped damped since the damping ratio < 1
Case 2 is underdamped since the damping ratio <  1
Case 3 is critically damped since the damping ratio = 1
Case 4 is overdamped since the damping ratio >  1
```



Published with MATLAB® R2020a

For each case of damper parameter the following function was called

```
function xhist = simulateMassSpringDamper(tvals,x0,xdot0,k,m,c)

syms X s x t f; % define symbolic variables
f = 0; % forcing function (could make this an input, if desired, but set to be zero
    for now)
F = laplace(f,t,s); % take laplace transform
X1 = s*X - x0; % laplace transform of x-dot
X2 = s^2*X - s*x0 - xdot0; % laplace transform of x-double-dot
Xsol = solve(X2 + c/m*X1 + k/m*X == F/m, X); % solve for X(s)
x = ilaplace(Xsol); % take inverse in MATLAB for x(t)
xhist = eval(subs(x,tvals)); % evaluate x(t) for tvals given

end
```