Problem I

Part I

$T(s) = \frac{16}{s^2 + 3s + 16}$ $\quad W_n^2 = 16 \rightarrow \omega_n = 4$ $\quad 2\zeta\omega_n = 3 \rightarrow \zeta = \frac{3}{2\omega_n} = 0.375$

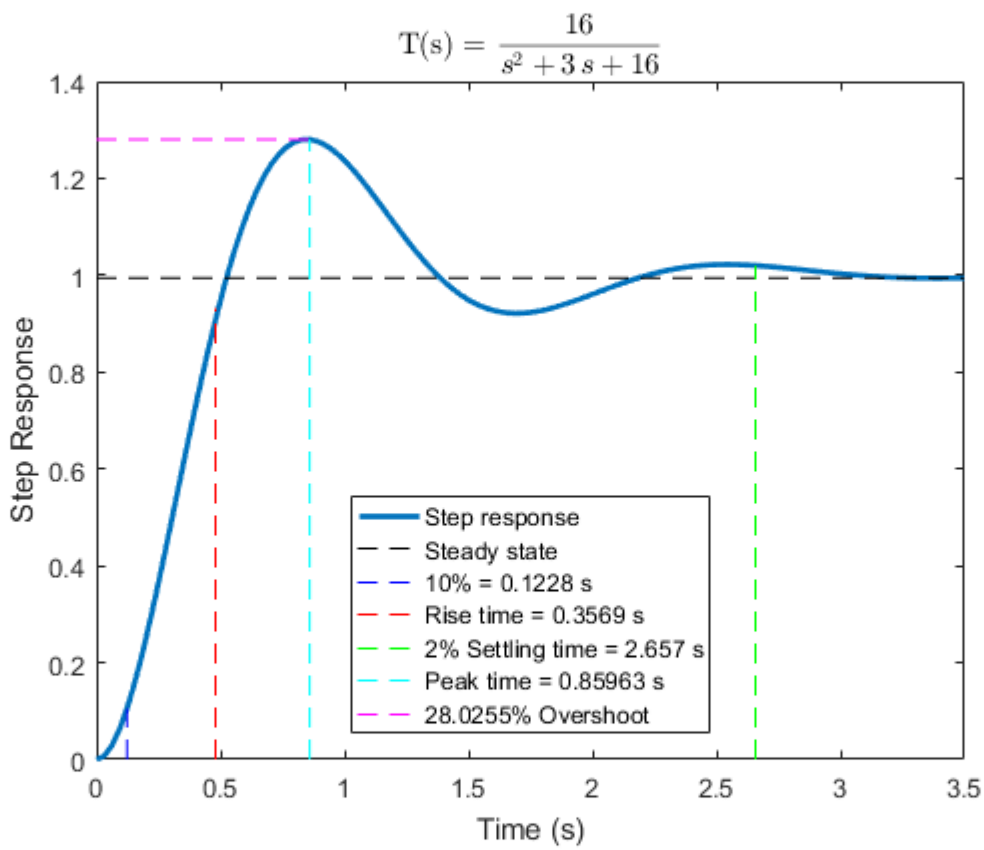
$T_r = \frac{1.36}{4} = 0.34s$ (Matlab 0.35s) ✓

$2\%\ T_s = \frac{4.0}{\zeta\omega_n} = \frac{4.0}{1.5} = 2.67s$ (Matlab 2.66s) ✓

$T_p = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}} = \frac{\pi}{4\sqrt{1-0.375^2}} = 0.847s$ (Matlab 0.856s) ✓

$\%OS = 100\, e^{-\left(\frac{\pi\zeta}{\sqrt{1-\zeta^2}}\right)} = 100\, e^{-\left(\frac{\pi\, 0.375}{\sqrt{1-0.375^2}}\right)} = 28.06\%$ (Matlab 28.02%) ✓

```
% Author:   Bradley Anderson
% Date:     Oct-21 2017
% Name:     ME 430 Computer Assignment 1, Problem 1, Part 1
clear, clf, clc

numerator = 16;
demoninator = [1 3 16];
settleValue = 0.02;

transferFunctionPlot(numerator, demoninator, settleValue);
```



Published with MATLAB® R2016a

Part 2

$$T(s) = \frac{0.04}{s^2 + 0.02s + 0.04} \qquad \omega_n^2 = 0.04 \rightarrow \omega_n = 0.2 \qquad 2\zeta\omega_n = 0.02 \rightarrow \zeta = \frac{0.02}{2\omega_n} = 0.05$$

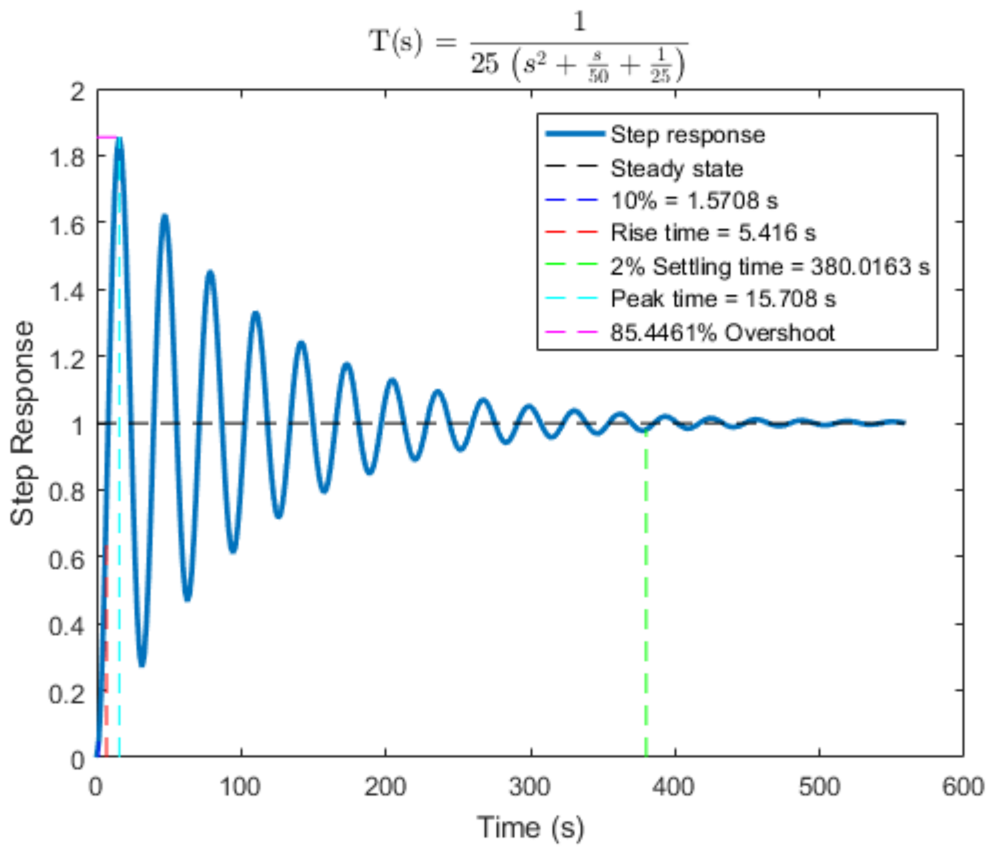
$$T_r = \frac{1.36}{0.2} = 6.8s$$ (Matlab 5.4s) ... meh

$$2\%\ T_s = \frac{4.0}{\zeta\omega_n} = \frac{4.0}{0.01} = 400s$$ (Matlab 380s) ✓

$$T_p = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}} = \frac{\pi}{0.2\sqrt{1-0.05^2}} = 15.73s$$ (Matlab 15.71s) ✓

$$\%\ OS = 100\, e^{-\left(\frac{\pi\zeta}{\sqrt{1-\zeta^2}}\right)} = 100^{-\left(\frac{\pi 0.05}{\sqrt{1-0.05^2}}\right)} = 85.446\ \%$$ (Matlab 85.446 %) ✓✓

```
% Author:    Bradley Anderson
% Date:      Oct-21 2017
% Name:      ME 430 Computer Assignment 1, Problem 1, Part 2
clear, clf, clc

numerator = 0.04;
demoninator = [1 0.02 0.04];
settleValue = 0.02;

[S, AxH] = transferFunctionPlot(numerator, demoninator, settleValue);

legend(AxH, 'Location', 'northeast')
```



Published with MATLAB® R2016a

Part 3

$T(s) = \frac{s+2.1}{(s+2)(s^2+s+5)}$ ... pole-zero cancellation $T(s) \approx \frac{1}{s^2+s+5}$

$\omega_n^2 = 5 \rightarrow \omega_n = \sqrt{5}$ $\quad 2\zeta\omega_n = 1 \rightarrow \zeta = \frac{1}{2\omega_n} = \frac{\sqrt{5}}{10} \approx 0.224$

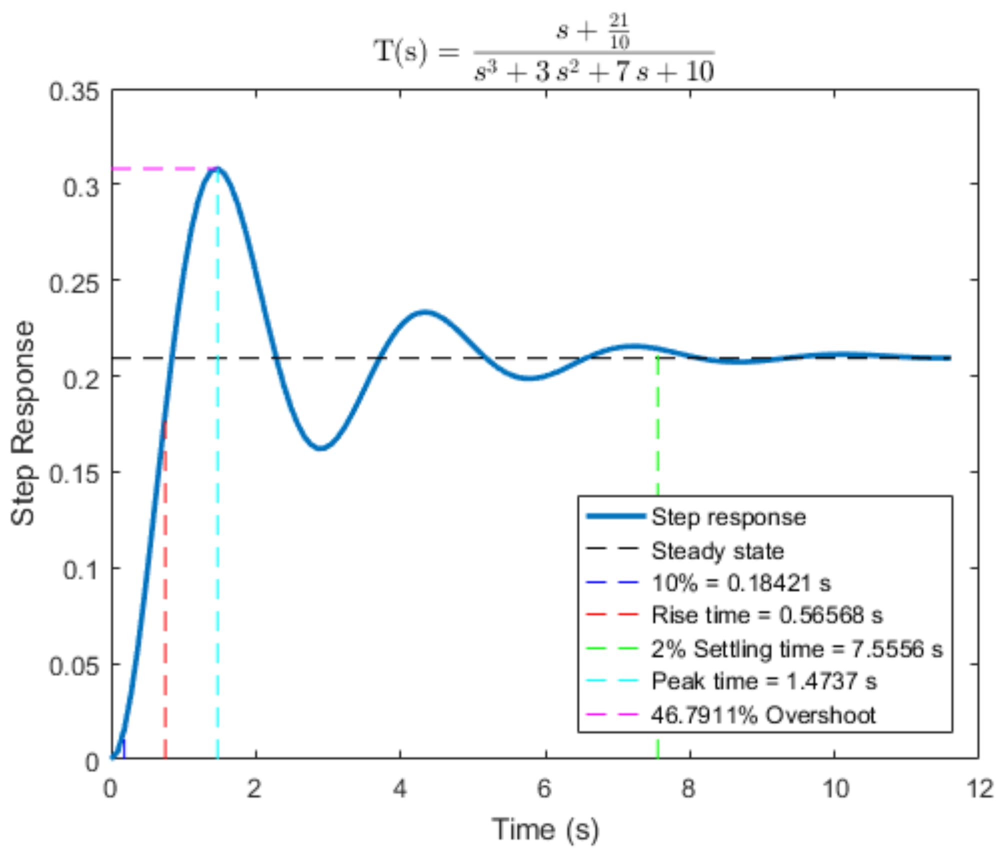
$T_r = \frac{1.36}{\omega_n} = 0.608s$ (Matlab $0.566s$) ✓

2% $T_s = \frac{4.0}{\zeta\omega_n} = \frac{4.0}{\frac{1}{2}} = 8.0s$ (Matlab $7.56s$) ✓

$T_p = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}} = \frac{\pi}{\sqrt{5}\sqrt{1-.224^2}} = 1.44s$ (Matlab $1.47s$) ✓

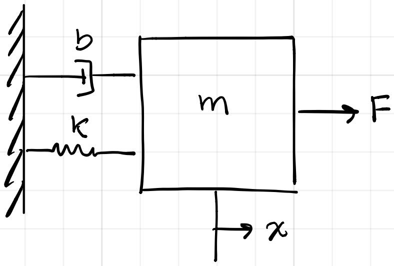
$\%OS = 100\, e^{-\left(\frac{\pi\zeta}{\sqrt{1-\zeta^2}}\right)} = 48.64\%$ (Matlab 46.79%) ✓

```
% Author:    Bradley Anderson
% Date:      Oct-21 2017
% Name:      ME 430 Computer Assignment 1, Problem 1, Part 3
clear, clf, clc

%Settling Time
settleValue = 0.02;

%Transfer Function Coefficients
num = [1 2.1];
dem1 = [1 2];
dem2 = [1 1 5];

Ts = series(tf(num, dem1),tf(1, dem2));

[numerator, demoninator] = tfdata(Ts);

[S, AxH] = transferFunctionPlot(numerator, demoninator, settleValue);

legend(AxH, 'Location', 'southeast')
```



Published with MATLAB® R2016a

Problem 2



$$\Sigma F: \quad ma = F - kx - bv$$

$$mx'' = F - kx - bx'$$

$$mx'' + bx' + kx = F$$

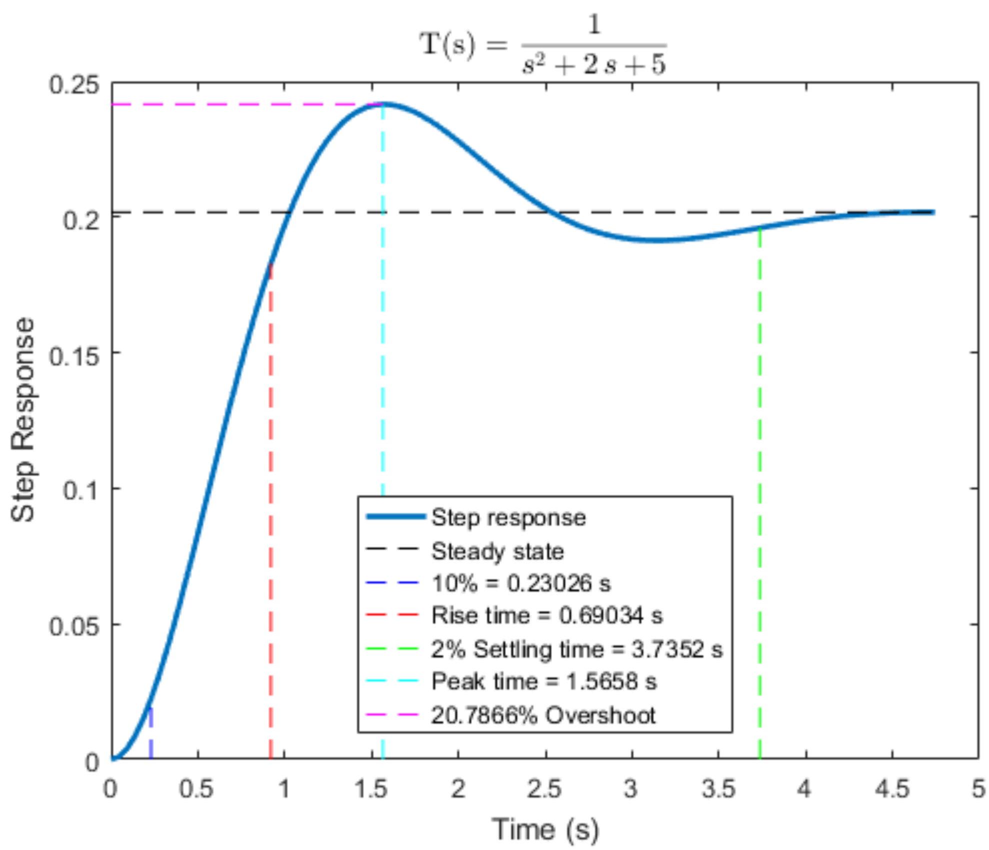
$$x'' + \frac{b}{m}x' + \frac{k}{m}x = F/m$$

$$Xs^2 + Xs\frac{b}{m} + X\frac{k}{m} = F(s)$$

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

$$\frac{X(s)}{F(s)} = \frac{1}{s^2 + \frac{b}{m}s + \frac{k}{m}} \qquad T(s) = \frac{1}{s^2 + 2s + 5}$$

```
% Author:    Bradley Anderson
% Date:      Oct-21 2017
% Name:      ME 430 Computer Assignment 1, Problem 2
clear, clf, clc

numerator = 1;
demoninator = [1 2 5];
settleValue = 0.02;

[S, AxH] = transferFunctionPlot(numerator, demoninator, settleValue);
```



Published with MATLAB® R2016a

```
% Author:   Bradley Anderson
% Date:     Oct-21 2017
% Name:     transferFunctionPlot
% Purpose:  Takes a numerator array, demoninator array, and settling
%           time. Creates a transfer function from parameters and
%           plots a step response. Overlays plot with rise time,
%           settling time of given ST, peak time, and percent
%           overshoot. Returns stepinfo and axes handle.

function [S, AxH] = transferFunctionPlot(numerator, demoninator, ST)

Ts = tf(numerator, demoninator);
[Y, T] = step(Ts);

% Format Title
% Takes coefficeint arrays and converts them into symbols that can be
% interpretted by latex
syms s
numSymPoly = sym(numerator);
denSymPoly = sym(demoninator);
numSym = poly2sym(numSymPoly, s);
denSyn = poly2sym(denSymPoly, s);
tfSym = numSym/denSyn;
tfTitle = latex(tfSym);

% Get info about the transfer function
S = stepinfo(Ts, 'SettlingTimeThreshold', ST);

% Plot formatting
AxH = axes;
func = plot(AxH,T, Y, 'LineWidth',2);
hold on
xlabel(AxH, 'Time (s)');
ylabel(AxH, 'Step Response');
title(AxH, sprintf('T(s) = $$ %s $$', tfTitle), 'Interpreter','latex')

% Strady state of function
steadyState = plot(AxH, [min(T), max(T)],[Y(end) Y(end)],'--k');

% Calculate ten percent time and index of that value in the time array
tenPercent = (Y(end)-Y(1))*0.1;
[~, tpIndex] = min( abs(Y-tenPercent) );

% Calculate ninety percent time and index of that value in the time
 array
ninetyPercentTime = S.RiseTime + T(tpIndex);
[~, npIndex] = min( abs(T-ninetyPercentTime) );

% Find index of settling time in time array
[~, stIndex] = min( abs(T - S.SettlingTime) );

% Plot into variables in order to set legend later
```

```
riseTime0 = plot(AxH, [T(tpIndex) T(tpIndex)], [Y(1) Y(tpIndex)], '--
b');
riseTime1 = plot(AxH, [ninetyPercentTime ninetyPercentTime], [Y(1)
 Y(npIndex)], '--r');
settTime = plot(AxH, [S.SettlingTime S.SettlingTime], [Y(1)
 Y(stIndex)],'--g');
peakTime = plot(AxH, [S.PeakTime S.PeakTime], [Y(1) S.Peak],'--c');
overshoot = plot(AxH, [T(1) S.PeakTime], [S.Peak S.Peak],'--m');

% Legend requires proper order, not tuples
legend(AxH, [func steadyState riseTime0 riseTime1 settTime peakTime
 overshoot], ...
    {'Step response', 'Steady state', ...
    ['10% = ', num2str(T(tpIndex)), ' s'], ...
    ['Rise time = ', num2str(S.RiseTime), ' s'], ...
    [num2str(ST*100),'% Settling time = ' num2str(S.SettlingTime) '
 s'] ...
    ['Peak time = ', num2str(S.PeakTime), ' s'], ...
    [num2str(S.Overshoot), '% Overshoot']}, ...
    'Location', 'south')

end
```

```
Not enough input arguments.

Error in transferFunctionPlot (line 12)
Ts = tf(numerator, demoninator);
```

Published with MATLAB® R2016a