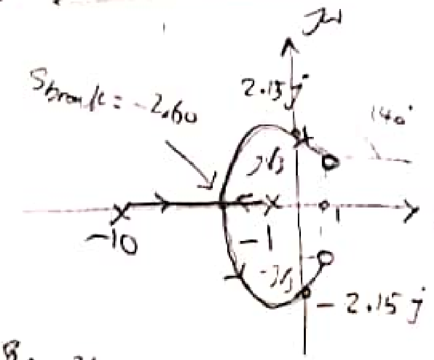
سوال ۱: $k>0$ 400101204 محمد باب دینی تمرین سری 5 سیستم های کنترل خطی

$$\text{I)}\quad G(s)=\frac{(s-1)^2+\sqrt{3}^2}{(s+1)(s+10)(s+30)} \rightarrow s^3+41s^2+340s+300$$



$$0=\frac{d}{ds}G(s) \Rightarrow -s^4+4s^3+410s^2+272s-1860=0 \rightarrow S_{break}\in\{-17.8,\ -2.6,\ 1.3,\ 22.5\}$$

$$\Longrightarrow S_{break}=-2.60$$

زاویه ورود به صفر

$$\theta+90-\tan^{-1}\left(\frac{\sqrt{3}}{2}\right)-\tan^{-1}\left(\frac{\sqrt{3}}{11}\right)=-180 \rightarrow \theta=-220 \overset{360}{\equiv} 139.84$$

$$\longrightarrow \sigma_a=\frac{-1-10-30-1-1}{1}=\boxed{-43}$$

$$1+KG(s):\ s^3+s^2(41+k)+s(340-2k)+(300+4k) \quad \rightarrow -41\le k\le 153.06$$

s^3	1	$340-2k$	
s^2	$41+k$	$300+4k$	
s^1	$\frac{(300+4k)-(41+k)(340-2k)}{-(41+k)}$	0	
s^0	$+300+4k$		

$$\begin{cases} k+41>0 \rightarrow k>-41 \\ 300+4k>0 \rightarrow k>-75 \\ 300+4k<(41+k)(340-2k) \rightarrow \underset{-44.56}{\frac{217-\sqrt{150209}}{4}}<k<\underset{153.06}{\frac{217+\sqrt{150209}}{4}} \\ 300+4k=(41+k)(340-2k) \rightarrow k=\frac{127}{2}\pm\frac{\sqrt{43409}}{2} \rightarrow 167.67,\ -40.67 \end{cases}$$

$$\longrightarrow a(s)=(41+k)s^2+(300+4k)\Big|_{k=\frac{127}{2}\pm\frac{\sqrt{43409}}{2}}=0 \longrightarrow s=\sqrt{\frac{-300-4k}{41+k}}$$

$$\left.\begin{array}{l} \text{if } k=-40.67 \longrightarrow s=\pm 20.4j \\ \text{if } k=167.67 \longrightarrow s=\pm 2.1578j \end{array}\right\} \longrightarrow$$

به علت مختلط بودن ریشه 20.4j، غیرقابل قبول است:

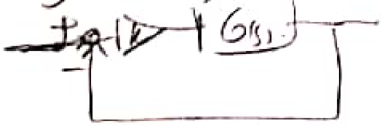
$s=\pm 2.1578j$

$$\ast\quad n-m=1 \longrightarrow \Phi_l=\frac{(2l+1)\pi}{n-m}\in\{\pi\}$$
$$0\le l\le n-m-1$$

- یک شاخه مجانب با زاویه 180° داریم.

- نقاط breaking out را از $0=\frac{d}{ds}G(s)$ به دست می آوریم که برابر $S=-2.60$ است.

- اگر بخواهیم که سیستم unstable شود، کافی است در جدول راث در ستون اول تغییر علامت داشته باشیم، پس برای هر k خارج بازه $[-41, 153.06]$ سیستم unstable است یعنی داریم



$\mathbb{R}-[-41,153.06]$ = unstable range of k

- برای صفر بالا زاویه برابر 140° درجه و برای صفر پایین -140° درجه است.

- محل برخورد مجانب ها $\sigma_a=-43$

ادامه سؤال ۱:

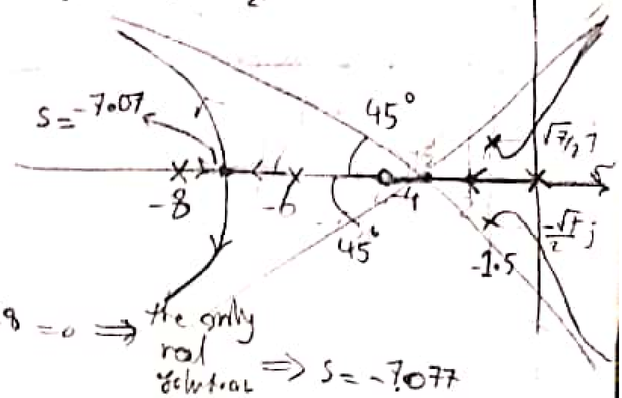
(II) $G(s) = \dfrac{s+4}{s(s+6)(s+8)(s^2+3s+4)}$ $(k>0)$

$\longrightarrow s^5 + 17s^4 + 94s^3 + 200s^2 + 192s$

$n-m=4 \rightarrow \phi_l = \dfrac{(2l+1)\pi}{n-m} \in [45°, 135°, -135°, -45°]$

$0 \le l \le 3$

$\sigma_A = \dfrac{-0-6-8-1.5-1.5-(-4)}{4} = \dfrac{-13}{4} = -3.25$



$\dfrac{d}{ds}G(s) = 0 \Rightarrow -4s^5 + 71s^4 + 460s^3 + 1328s^2 + 1600s + 768 = 0 \Rightarrow$ the only real solution $\Rightarrow s = -7.077$ breaking point

زاویه خروج از قطب $\Rightarrow -\theta - 90 + \tan^{-1}\left(\dfrac{\sqrt{7}/2}{4-1.5}\right) + \tan^{-1}\left(\dfrac{\sqrt{7}/2}{6-1.5}\right) - \tan^{-1}\left(\dfrac{\sqrt{7}/2}{8-1.5}\right) - 135 = -180$

$\Longrightarrow \theta_p = -43.43°$ $\Longrightarrow \theta'_p = +43.43°$

$1+kG(s) \Rightarrow s^5 + 17s^4 + 94s^3 + 200s^2 + (192+k)s + 4k$

s^5	1	94	$192+k$
s^4	17	200	$4k$
s^3	82.23	$\dfrac{4k-3264-17k}{-17}$	0
s^2	$\dfrac{13k-13183.05}{-82.23}$	$4k$	0
s	$f(k)$	0	0
s^0	$4k$	0	0

$f(k) = \dfrac{-0.764k^2 - 1497.07k + 194704}{k - 1014.08}$

شروط لازم برای پایداری

$$\begin{cases} 4k > 0 \longrightarrow k > 0 \\ \dfrac{13k - 13183.05}{-82.23} > 0 \longrightarrow k < 1014.08 \\ f(k) > 0 \longrightarrow (-\infty, -2080.11) \cup (122.40, 1014.08) \end{cases}$$

از ترکیب شروط بالا داریم:

$\boxed{122.404 \le k \le 1014.08}$

$S = -7.07 \Rightarrow$ breaking point -

$\pm 45°, \pm 135° \leftarrow$ زوایای مجانب ها -

$\phi_p = \pm 43.43° \Rightarrow$ angle of departure -

$\mathbb{R} - [122.404, 1014.08]$ unstability Range for k -

- سیستم نمی‌تواند critically damped باشد.

in this section we will prove that the real axis intersect of asymptotes: :2 نیاش

let $1+KG(s)H(s) = 1+\frac{kb_0}{a_0}\frac{\prod(s-z_i)}{\prod(s-p_i)} = 1+k\frac{b_0}{a_0}\times\frac{s^m-(\sum z_i)s^{m-1}+(\sum_{i,j} z_i z_j)s^{m-2}+\dots}{s^n-(\sum p_i)s^{n-1}+(\sum_{i,j} p_i p_j)s^{n-2}+\dots}$

$$\lim_{s\to\infty} 1+KG(s)H(s) \approx 1+\lim_{s\to\infty}\frac{kb_0}{a_0}\times\frac{s^m-s^{m-1}(\sum_{i=1}^{r} z_i)}{s^n-s^{n-1}(\sum_{i=1}^{n} p_i)} = \frac{kb_0}{a_0}\lim_{s\to\infty}\frac{1-s^{-1}(\sum_{i=1}^{r} z_i)}{s^{n-m}-s^{n-m-1}(\sum_{i=1}^{n} p_i)}$$

let us also assume that $n>m$ (there are $n-m$ branches that will diverge to infinity)

lemma: $-s^{-1}\left(\sum_{i=1}^{r} z_i\right) \approx \frac{1}{1+s^{-1}(\sum_{i=1}^{n} z_i)}$

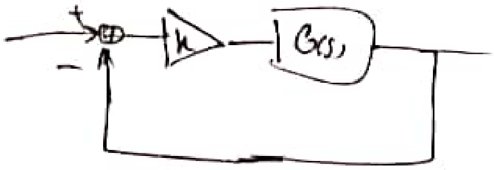
$$\Rightarrow \frac{kb_0}{a_0}\times\frac{1}{(s^{n-m}-s^{n-m-1}\sum_{i=1}^{n} p_i)(1+s^{-1}\sum_{i=1}^{r} z_i)} = k\frac{b_0}{a_0}\times\frac{1}{s^{n-m}-s^{n-m-1}\left(\sum_{i=1}^{n} p_i-\sum_{i=1}^{r} z_i\right)+s^{n-m-2}\sum_{i,j} p_i z_j}$$

$$\lim_{s\to\infty} kG(s)H(s) \approx \frac{kb_0}{a_0}\lim_{s\to\infty}\frac{1}{\underbrace{s^{n-m}}_{q}-s^{n-m-1}\underbrace{\left(\sum_{i=1}^{n} p_i-\sum_{j=1}^{r} z_j\right)}_{\alpha}} = \lim_{s\to\infty}\frac{kb_0}{a_0}\times\frac{1}{s^q-\alpha s^{q-1}} = \frac{kb_0}{a_0}\times\frac{1}{(s-\sigma)^q}$$

lemma: $\lim_{s\to\infty}(s-\sigma)^q \approx s^q-q\sigma s^{q-1} \longrightarrow \boxed{\sigma=\frac{\alpha}{q}=\frac{\sum_{i=1}^{n} p_i-\sum_{j=1}^{r} z_j}{n-m}}$

→ here $\frac{kb_0}{a_0}\times\frac{1}{(s-\sigma)^q}$ represents a set of vectors that intersects the real axis at $s=-\sigma$ that radiate outwards with angles $\theta_l=\pm\frac{(2l+1)\pi}{n-m}$

سوال 3:



$$\begin{cases} H(s) = \dfrac{kG(s)}{1+kG(s)} = \dfrac{k(s+2)}{s^2+(k+2)s+(2k+1)} \\ \forall\ 0 \le k \le 4 \longrightarrow \sqrt{k^2-4k} = j\sqrt{4k-k^2} \\ k^2-4k \le 0 \end{cases}$$

$$s_{1,2} = -\frac{k+2}{2} \pm \frac{j}{2}\sqrt{4k-k^2} = x+jy$$

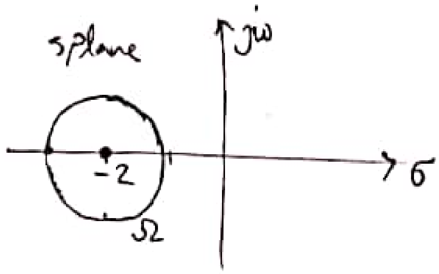
حال x و y که در صفحه مختلط $s=x+jy$ قرار دارند را از پارامتر یک $\gamma(t)$ در نظر می‌گیریم $0 \le t \le 4$

و نشان می‌دهیم که برای $0<t<4$ ، $\gamma_+(t) = \left(-\frac{t+2}{2}, +\frac{\sqrt{4t-t^2}}{2}\right)$ یک نیم دایره داریم . برای حالت

$\gamma_-(t) = \left(-\frac{t+2}{2}, -\frac{\sqrt{4t-t^2}}{2}\right)$ نیز به همین صورت ثابت می‌شود .

$$x(t) = -\frac{t}{2} - 1 \rightarrow t = -2x-2 \rightarrow y^2 = \frac{4t-t^2}{4} = \frac{-4(2x+2)-(2x+2)^2}{4}$$

$$\longrightarrow y^2 + 4x + x^2 = -3 \longrightarrow (x+2)^2 + y^2 = 1$$



دایره به مرکز $O(-2,0)$ و شعاع $r=1$

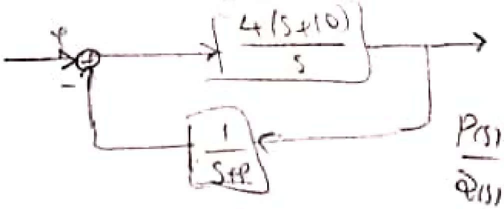
$$\forall t \in [0,4]: \gamma_\pm(t) = \left(-\frac{t+2}{2}, \pm\sqrt{4t-t^2}\right)$$

$\Rightarrow$ پس از نتیجه قطب‌های $H(s)$ ، با تغییر k از 0 به 4 روی دایره Ω حرکت می‌کنند

$$\ell\left(O\binom{-2}{0}, r=1\right)$$

سوال 5:

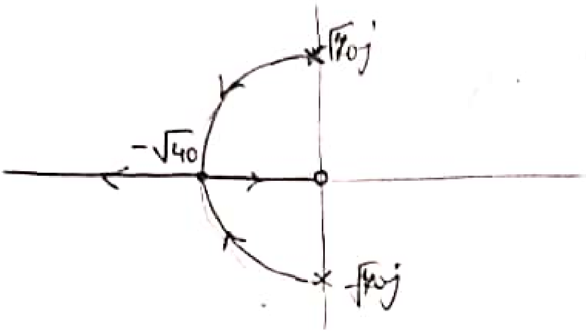
$p \in \mathbb{R}^+ \cup \{0\}$



$$\frac{P(s)}{Q(s)} = H(s) = \frac{G(s)}{1+M(s)G(s)} = \frac{\frac{4(s+10)}{s}}{1+\frac{4(s+10)}{s(s+p)}} = \frac{4(s+10)(s+p)}{s^2+s(4+p)+40}$$

$$\begin{cases} Q(s) = s^2+(40+p)s+40 = 1+\dfrac{(40+p)s}{s^2+40} = 1+K\dfrac{s}{s^2+40} \\ \text{let } K = p+40 \end{cases}$$

حال با نمودار مکان ریشه $\frac{s}{s^2+40}$ را رسم می‌کنیم



$$\frac{d}{ds}\left(\frac{s}{s^2+40}\right) = \frac{2s^2-s^2-40}{(s^2+40)^2} = \frac{s^2-40}{(s^2-40)^2} = 0$$

$\rightarrow s = \pm\sqrt{40}$

LCS hw5 Software assignment

Dr.Behzad Ahi

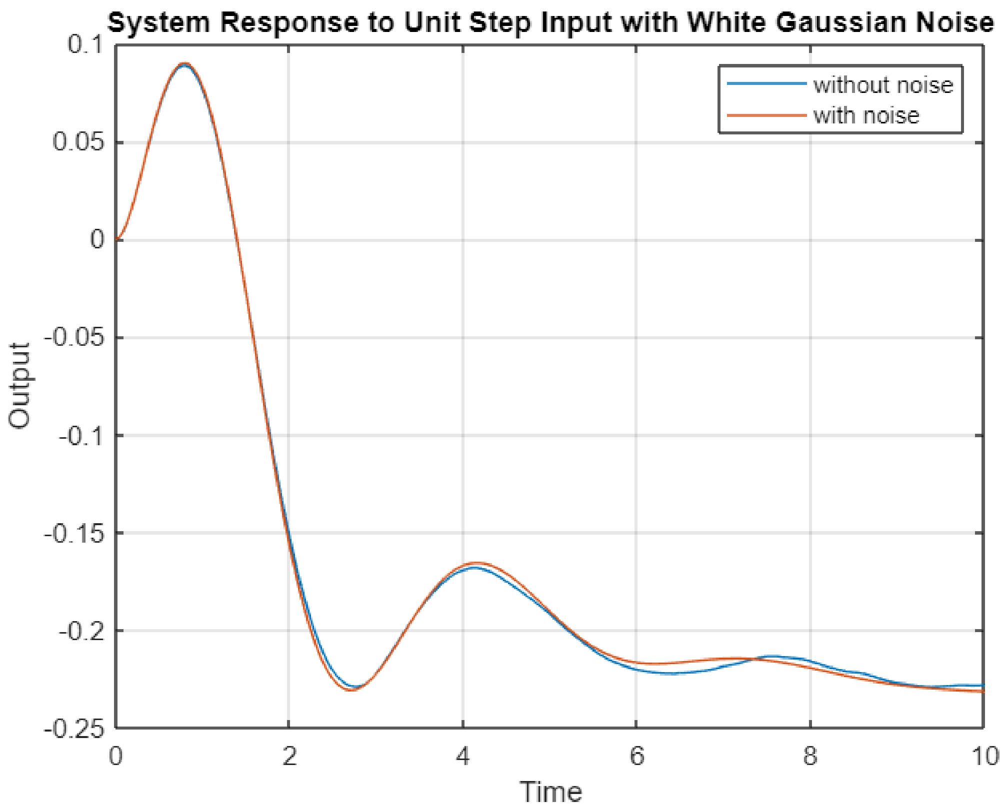
MohammadParsa Dini - 400101204

Problem 6:

part 1: Input noise:

```
clc; clear; close all;
% Define the parameters
k = 1;
b1 = 3;
b2 = 1;
wn = 2;
tau1 = 0.5;
tau2 = 5;
zeta = 0.3;

num = k * conv([b1, 1], [1, -b2]);
den = conv([tau1, 1], [tau2, 1]);
den = conv(den, [1, 2*zeta*wn, wn^2]);

% Create transfer function G(s)
G = tf(num, den);

% Define the time vector
t = linspace(0, 10, 1000); % Change the time range and number of points as needed

% Generate a unit step input
u = ones(size(t));

% Generate white Gaussian noise with mean 0 and standard deviation 0.1
noise = 0.1 * randn(size(t));

% Add noise to the input
u_with_noise = u + noise;

% Simulate the system's response to the input with noise
[y, ~, ~] = lsim(G, u_with_noise, t);
[yy,v,vv] = lsim(G, u, t);

% Plot the output response
figure;
plot(t, y);
hold on
plot(t, yy);
xlabel('Time');
ylabel('Output');
title('System Response to Unit Step Input with White Gaussian Noise');
legend('without noise','with noise')
```

```
grid on;
```

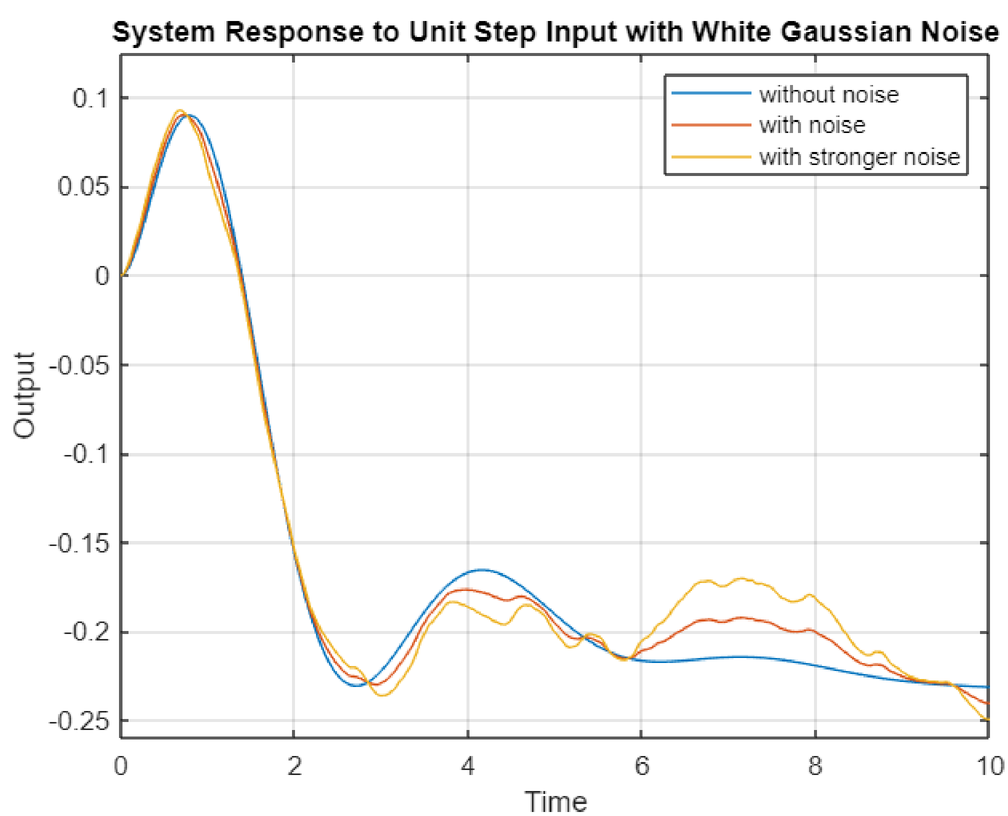


As you can see, there is not much difference, since the SNR of input was hogh, if we add a WGN with more energy we might see

much more deviations in that case:

```
clc; clear; close all;
% Define the parameters
k = 1;
b1 = 3;
b2 = 1;
wn = 2;
tau1 = 0.5;
tau2 = 5;
zeta = 0.3;

num = k * conv([b1, 1], [1, -b2]);
den = conv([tau1, 1], [tau2, 1]);
den = conv(den, [1, 2*zeta*wn, wn^2]);

% Create transfer function G(s)
G = tf(num, den);

% Define the time vector
t = linspace(0, 10, 1000); % Change the time range and number of points as needed

% Generate a unit step input
```

```
u = ones(size(t));

% Generate white Gaussian noise with mean 0 and standard deviation 0.1
noise = 0.5 * randn(size(t));

% Add noise to the input
u_with_noise = u + noise;
u_with_noise_ = u + 2*noise;

% Simulate the system's response to the input with noise
[y, ~, ~] = lsim(G, u_with_noise, t);
[yy,v,vv] = lsim(G, u, t);
[yyy, vvv, gg] = lsim(G, u_with_noise_, t);

% Plot the output response
figure;
plot(t, yy);
hold on
plot(t, y);
hold on;
plot(t, yyy);
xlabel('Time');
ylabel('Output');
title('System Response to Unit Step Input with White Gaussian Noise');
legend('without noise','with noise','with stronger noise');
ylim([-.26 0.125]);
grid on;
```

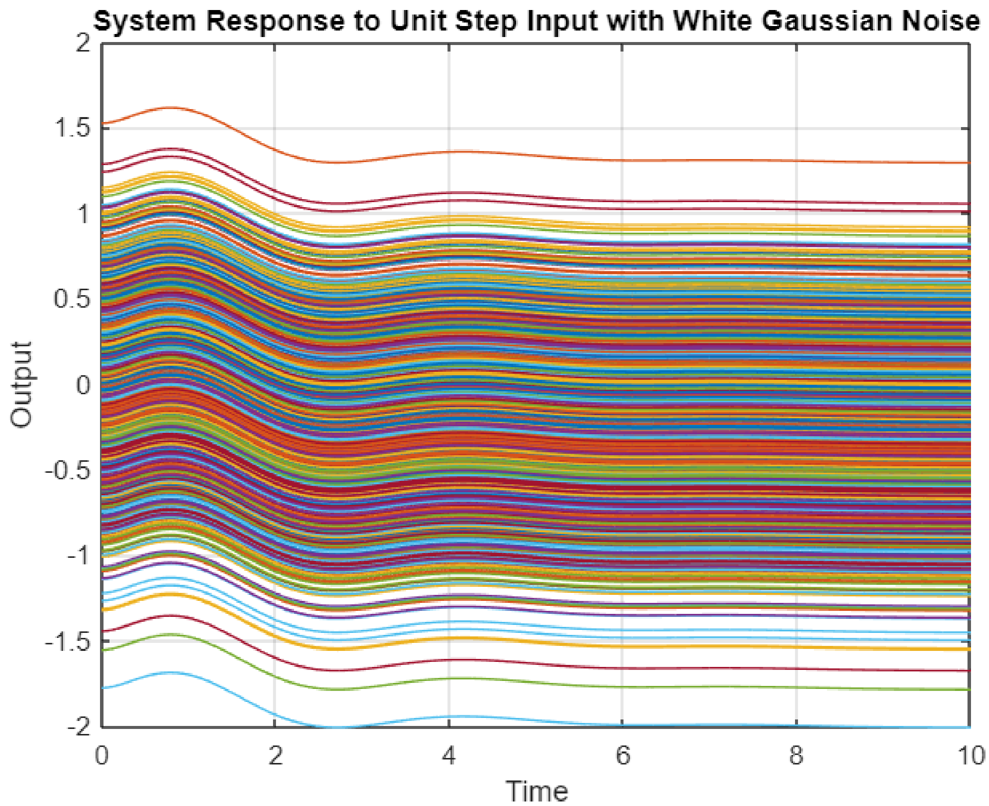


As we can see, the difference between the ideal output and noisy output will rise as time passes. Furthermore, in contrast to the ideal case, the noisy output is not smoothely changing and it has much more high frequency.

Problem 6:

part 2: Output noise

```
clc; clear; close all;
% Define the parameters
k = 1;
b1 = 3;
b2 = 1;
wn = 2;
tau1 = 0.5;
tau2 = 5;
zeta = 0.3;

num = k * conv([b1, 1], [1, -b2]);
den = conv([tau1, 1], [tau2, 1]);
den = conv(den, [1, 2*zeta*wn, wn^2]);

% Create transfer function G(s)
G = tf(num, den);

% Define the time vector
t = linspace(0, 10, 1000); % Change the time range and number of points as needed

% Generate a unit step input
u = ones(size(t));

% Generate white Gaussian noise with mean 0 and standard deviation 0.1
noise = 0.1 * randn(size(t));

% Simulate the system's response to the input with noise
[y, ~, ~] = lsim(G, u, t);

y1 = y + noise;
y2 = y + 5*noise;
% Plot the output response
figure;
plot(t, y);
hold on
plot(t, y1 - y);
hold on;
plot(t, y2);
xlabel('Time');
ylabel('Output');
title('System Response to Unit Step Input with White Gaussian Noise');
%legend('without noise','with noise','with stronger noise');
grid on;
```



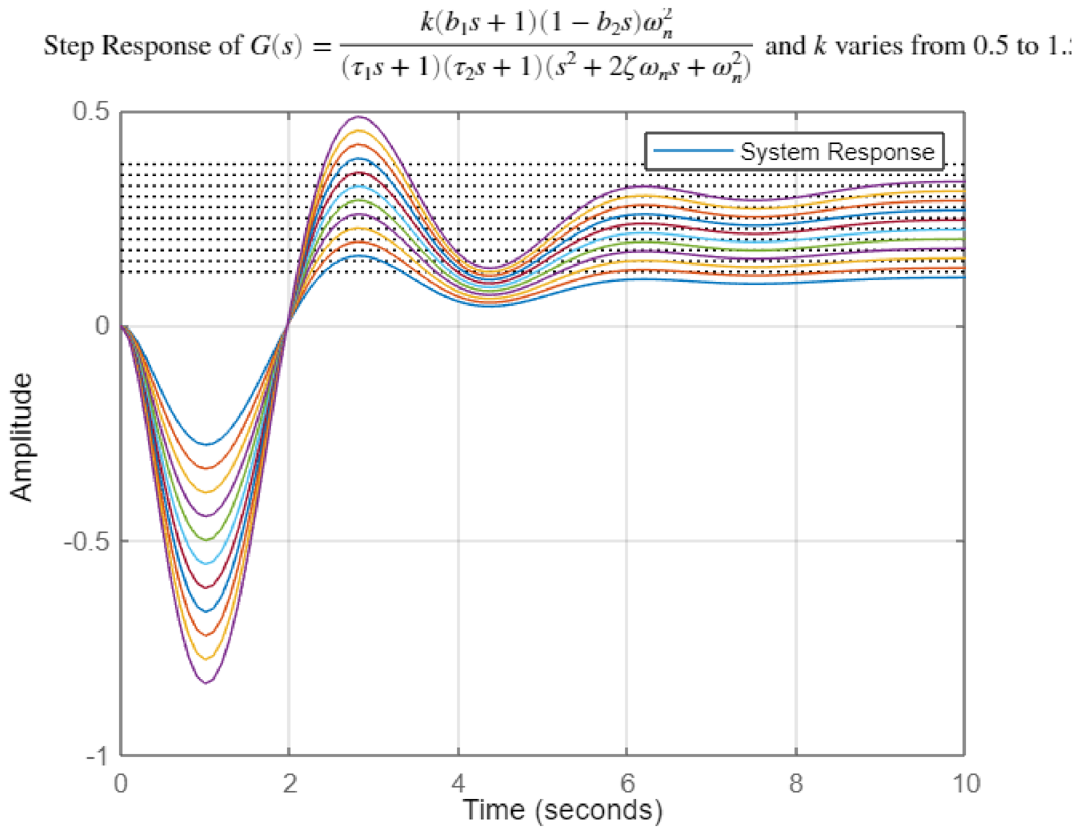
Problem 6:

part C:

Here we tuned k from .5 to 1.5 or equivalently with 50 percent deviation in its ideal case:

```
clc; clear; close all;
syms s
% Defining the parameters of the transfer function
param = 0.5:0.1:1.5;
k = 1;
b1 = 3;
b2 = 1;
wn = 2;
tau1 = 0.5;
tau2 = 5;
zeta = 0.3;
% loop for deviating k
for i = 1:length(param)
    hold on
    num = param(i)*(b1*s + 1)*(1 - b2 *s*wn^2);
    den = (tau2*s+1)*(tau1*s+1)*(s^2 + 2*zeta*wn*s + wn^2);
    H_sys = tf(sym2poly(num), sym2poly(den));
    t = 0:0.1:10;
    step(H_sys, t);
    xlabel('Time');
    ylabel('Amplitude');
```

```
    title(['Step Response of $G(s) = \frac{k(b_1 s + 1)(1 - b_2 s)\omega_n^2}' ...
        '{( \tau_1 s + 1)(\tau_2 s + 1)(s^2 + 2\zeta\omega_n s + \omega_n^2)}$ and $k$ ' ...
        'varies from 0.5 to 1.5'], ...
        'interpreter', 'latex');

    grid on;
    legend('System Response');
end
```



As we can see, as k rises, the amplitude of the output will rise as well.

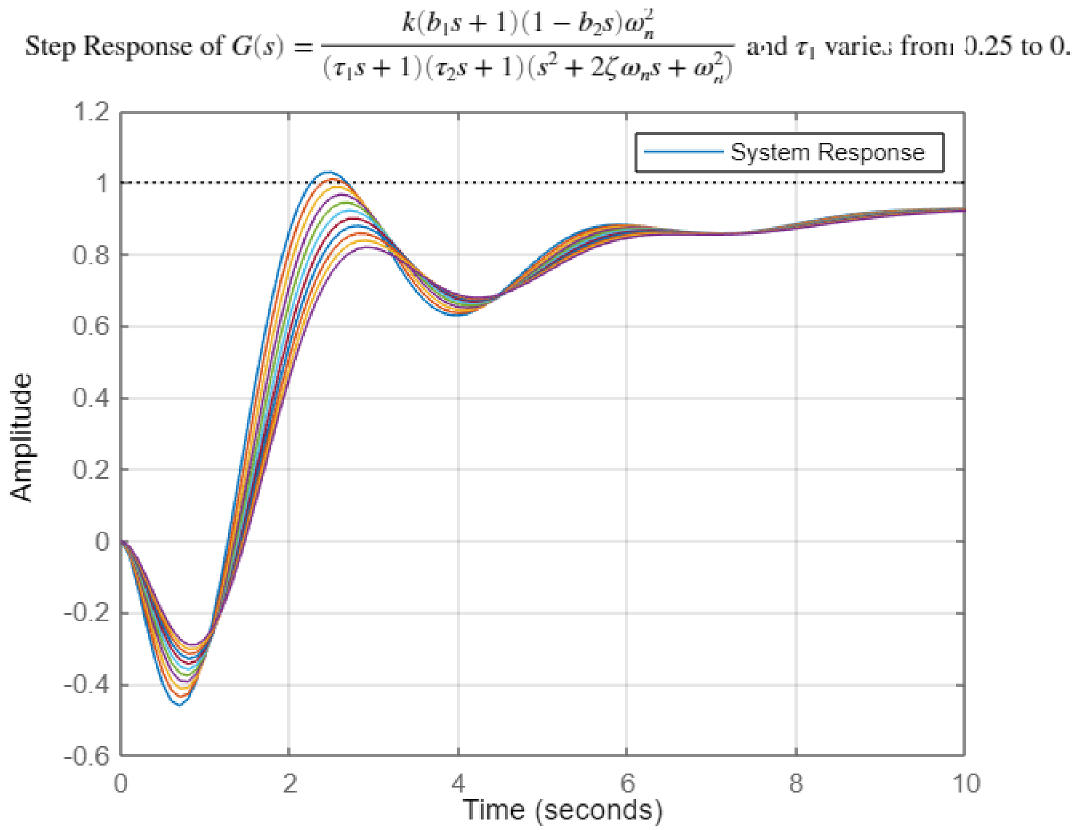
<u>Problem 6:</u>

<u>part D:</u>

Here we tuned τ_1 from 0.25 to 0.75 or equivalently with 50 percent deviation in its ideal case:

```
clc; clear; close all;
syms s
% Defining the parameters of the transfer function
param = 0.25:0.05:0.75;
k = 1;
b1 = 3;
b2 = 1;
wn = 2;
tau1 = 0.5;
tau2 = 5;
zeta = 0.3;
```

```matlab
% loop for deviating tau1
for i = 1:length(param)
    t = 0:0.1:10;
    hold on
    num = k*(b1*s + 1)*(1 -b2 *s)*wn^2;
    den = (tau2*s+1)*(param(i)*s+1)*(s^2 + 2*zeta*wn*s + wn^2);
    H_sys = tf(sym2poly(num), sym2poly(den));
    step(H_sys, t);
    xlabel('Time');
    ylabel('Amplitude');
    title(['Step Response of $G(s) = \frac{k(b_1 s + 1)(1 - b_2 s)\omega_n^2}' ...
        '{( \tau_1 s + 1)(\tau_2 s + 1)(s^2 + 2\zeta\omega_n s + \omega_n^2)}$ and $\tau_1$ '
        'varies from 0.25 to 0.75'], ...
        'interpreter', 'latex');
    grid on;
    legend('System Response');
end
```



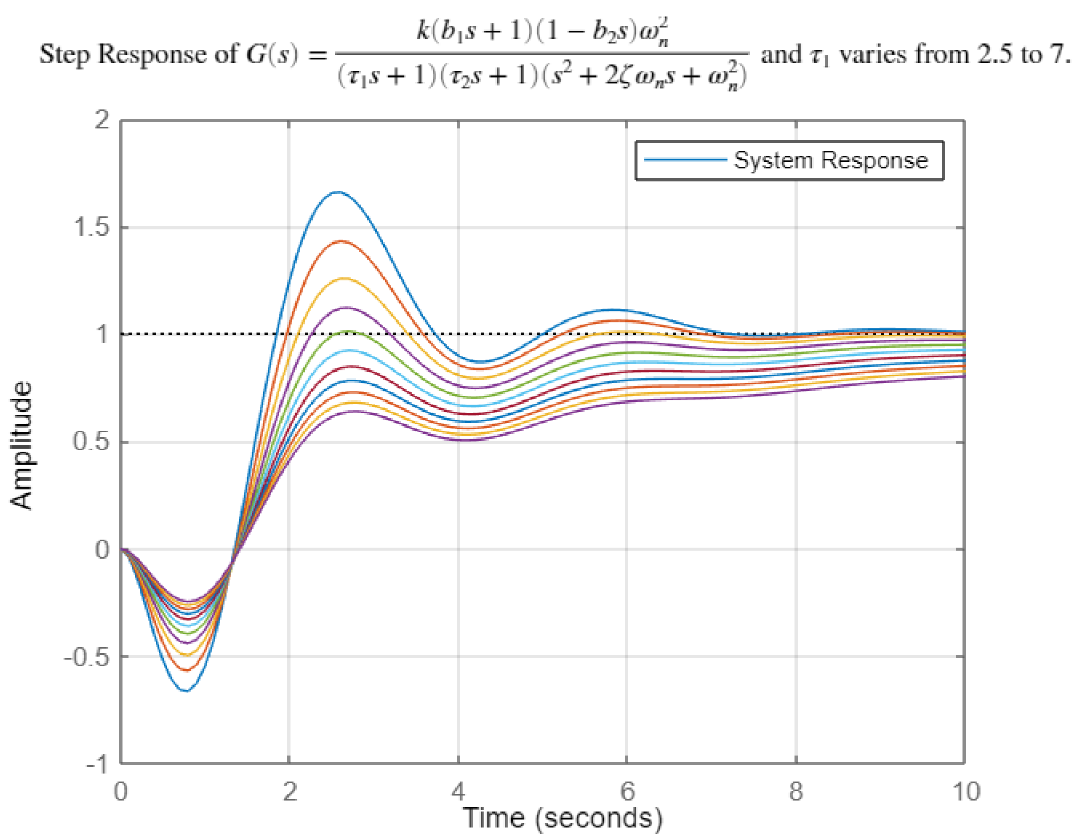
As τ_1 increases, the pole $\frac{1}{\tau_1}$ will be closer to $j\omega$ axis and hence, the amplitude will be lower. Furthermore, the setting time will increase.

Problem 6:

part E:

Here we tuned τ_2 from 2.5 to 7.5 or equivalently with 50 percent deviation in its ideal case:

```
clc; clear; close all;
syms s
% Defining the parameters of the transfer function
param = 2.5:0.5:7.5;
k = 1;
b1 = 3;
b2 = 1;
wn = 2;
tau1 = 0.5;
tau2 = 5;
zeta = 0.3;
% loop for deviating tau2
for i = 1:length(param)
    t = 0:0.1:10;
    hold on
    num = k*(b1*s + 1)*(1 - b2 *s)*wn^2;
    den = (param(i)*s+1)*(tau1*s+1)*(s^2 + 2*zeta*wn*s + wn^2);
    H_sys = tf(sym2poly(num), sym2poly(den));
    step(H_sys, t);
    xlabel('Time');
    ylabel('Amplitude');
    title(['Step Response of $G(s) = \frac{k(b_1 s + 1)(1 - b_2 s)\omega_n^2}' ...
        '{( \tau_1 s + 1)(\tau_2 s + 1)(s^2 + 2\zeta\omega_n s + \omega_n^2)}$ and $\tau_1$ ' 
        'varies from 2.5 to 7.5'], ...
        'interpreter', 'latex');
    grid on;
    legend('System Response');
end
```



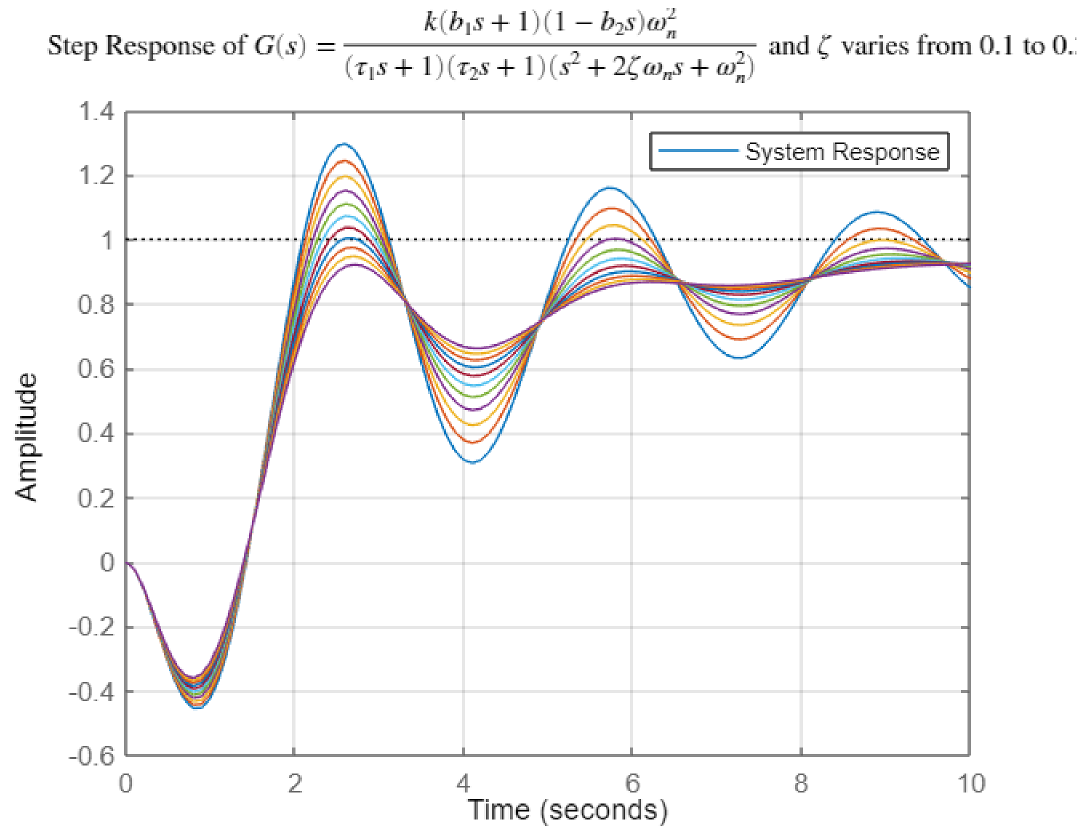
The same argument will be applied to this part. The amplitude will have less overshoot and the setting time will rise drastically.

<u>Problem 6:</u>

<u>part F:</u>

Here we tuned ζ from 0.1 to 0.3 or equivalently with 50 percent deviation in its ideal case:

```
clc; clear; close all;
syms s
% Defining the parameters of the transfer function
param = 0.1:0.02:0.3;
k = 1;
b1 = 3;
b2 = 1;
wn = 2;
tau1 = 0.5;
tau2 = 5;
zeta = 0.3;
% loop for deviating zeta
for i = 1:length(param)
    t = 0:0.1:10;
    hold on
    num = k*(b1*s + 1)*(1 - b2 *s)*wn^2;
    den = (tau2*s+1)*(tau1*s+1)*(s^2 + 2*param(i)*wn*s + wn^2);
    H_sys = tf(sym2poly(num), sym2poly(den));
    step(H_sys, t);
    xlabel('Time');
    ylabel('Amplitude');
    title(['Step Response of $G(s) = \frac{k(b_1 s + 1)(1 - b_2 s)\omega_n^2}' ...
        '{( \tau_1 s + 1)(\tau_2 s + 1)(s^2 + 2\zeta\omega_n s + \omega_n^2)}$ and $\zeta$ ' ..
        'varies from 0.1 to 0.3'], ...
        'interpreter', 'latex');
    grid on;
    legend('System Response');
end
```



By increasing ζ, the overshoot will shrink since $overshoot = f_v \times e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}}$ (we assumed this function will behave as a 2nd order function). and furthermore, the setting time will be lowered.

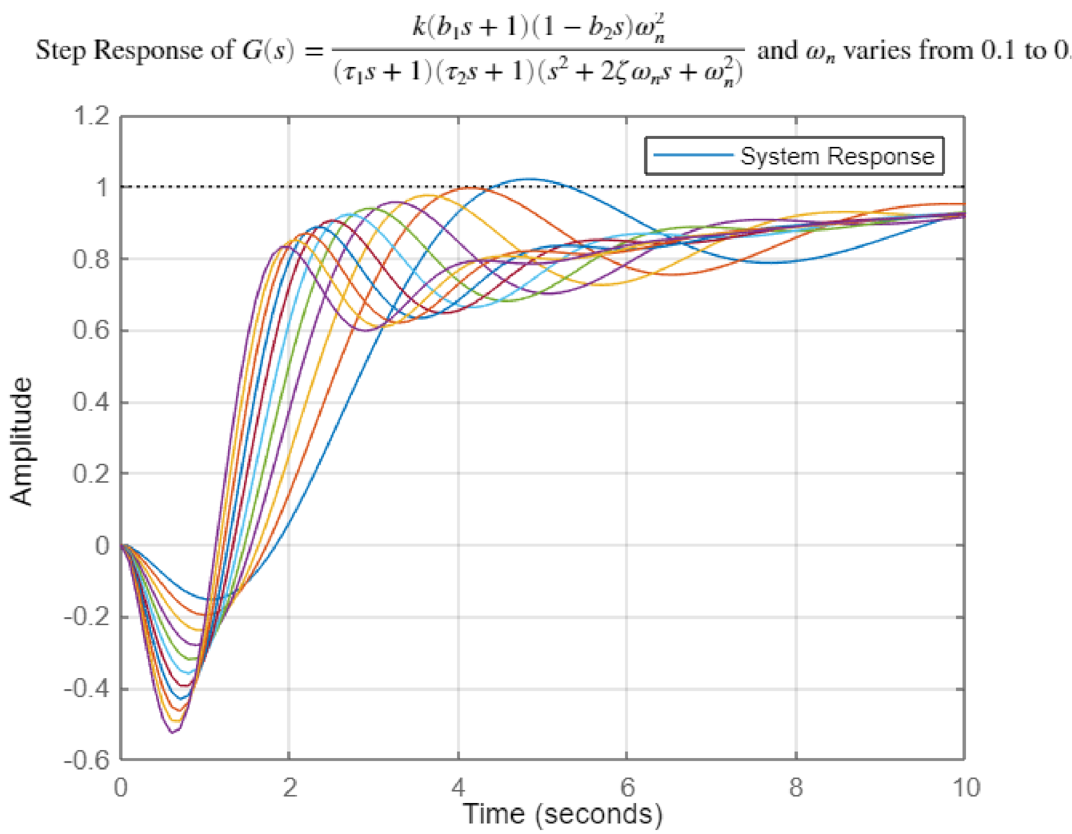
Problem 6:

part G:

Here we tuned ω_n from 1 to 3 or equivalently with 50 percent deviation in its ideal case:

```
clc; clear; close all;
syms s
% Defining the parameters of the transfer function
param = 1:0.2:3;
k = 1;
b1 = 3;
b2 = 1;
wn = 2;
tau1 = 0.5;
tau2 = 5;
zeta = 0.3;
% loop for deviating wn
for i = 1:length(param)
    hold on
    num = k*(b1*s + 1)*(1 - b2 *s)*param(i)^2;
    den = (tau2*s+1)*(tau1*s+1)*(s^2 + 2*zeta*param(i)*s + param(i)^2);
```

```
    H_sys = tf(sym2poly(num), sym2poly(den));
    t = 0:0.1:10;
    step(H_sys, t);
    xlabel('Time');
    ylabel('Amplitude');
    title(['Step Response of $G(s) = \frac{k(b_1 s + 1)(1 - b_2 s)\omega_n^2}' ...
        '{( \tau_1 s + 1)(\tau_2 s + 1)(s^2 + 2\zeta\omega_n s + \omega_n^2)}$ and $\omega_n$
        'varies from 0.1 to 0.3'], ...
        'interpreter', 'latex');
    grid on;
    legend('System Response');
end
```



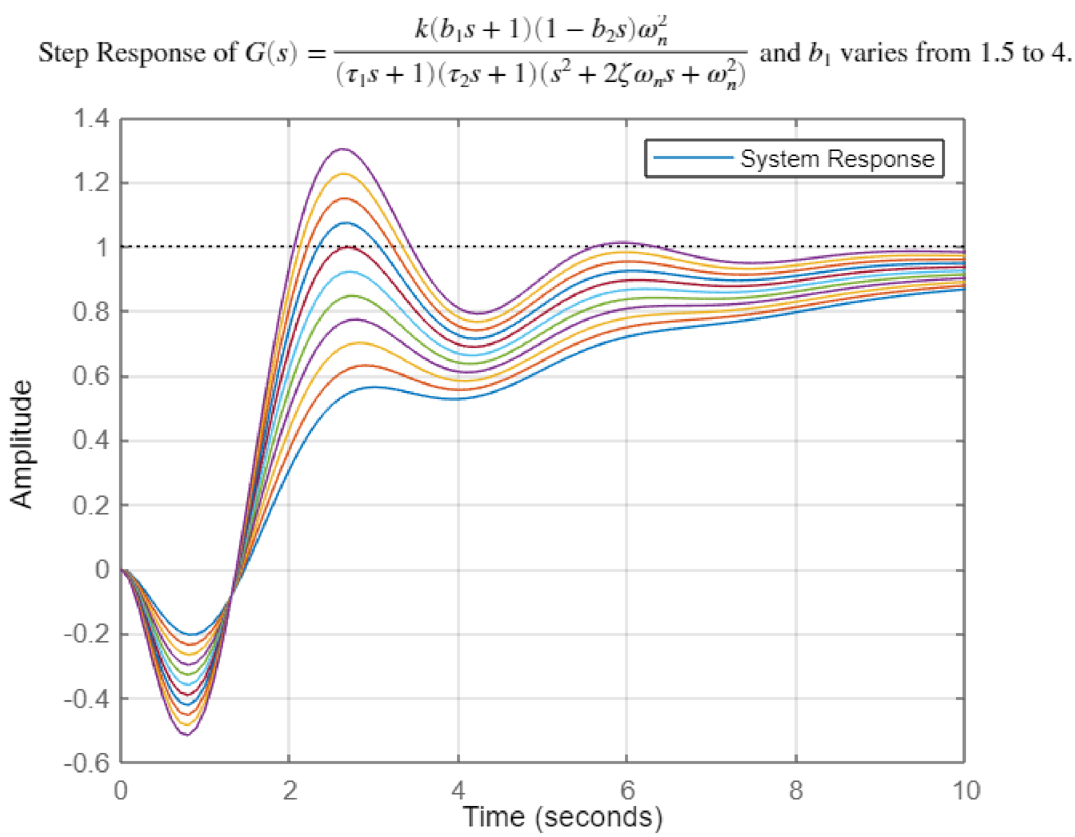
<u>Problem 6:</u>

<u>part G:</u>

Here we tuned b_1 from 1 to 3 or equivalently with 50 percent deviation in its ideal case:

```
clc; clear; close all;
syms s
% Defining the parameters of the transfer function
param = 1.5:0.3:4.5;
k = 1;
b1 = 3;
b2 = 1;
wn = 2;
tau1 = 0.5;
```

```
tau2 = 5;
zeta = 0.3;
% loop for deviating b1
for i = 1:length(param)
    hold on
    num = k*(param(i)*s + 1)*(1 - b2 *s)*wn^2;
    den = (tau2*s+1)*(tau1*s+1)*(s^2 + 2*zeta*wn*s + wn^2);
    H_sys = tf(sym2poly(num), sym2poly(den));
    t = 0:0.1:10;
    step(H_sys, t);
    xlabel('Time');
    ylabel('Amplitude');
    title(['Step Response of $G(s) = \frac{k(b_1 s + 1)(1 - b_2 s)\omega_n^2}' ...
        '{( \tau_1 s + 1)(\tau_2 s + 1)(s^2 + 2\zeta\omega_n s + \omega_n^2)}$ and $b_1$ ' ...
        'varies from 1.5 to 4.5'], ...
        'interpreter', 'latex');
    grid on;
    legend('System Response');
end
```



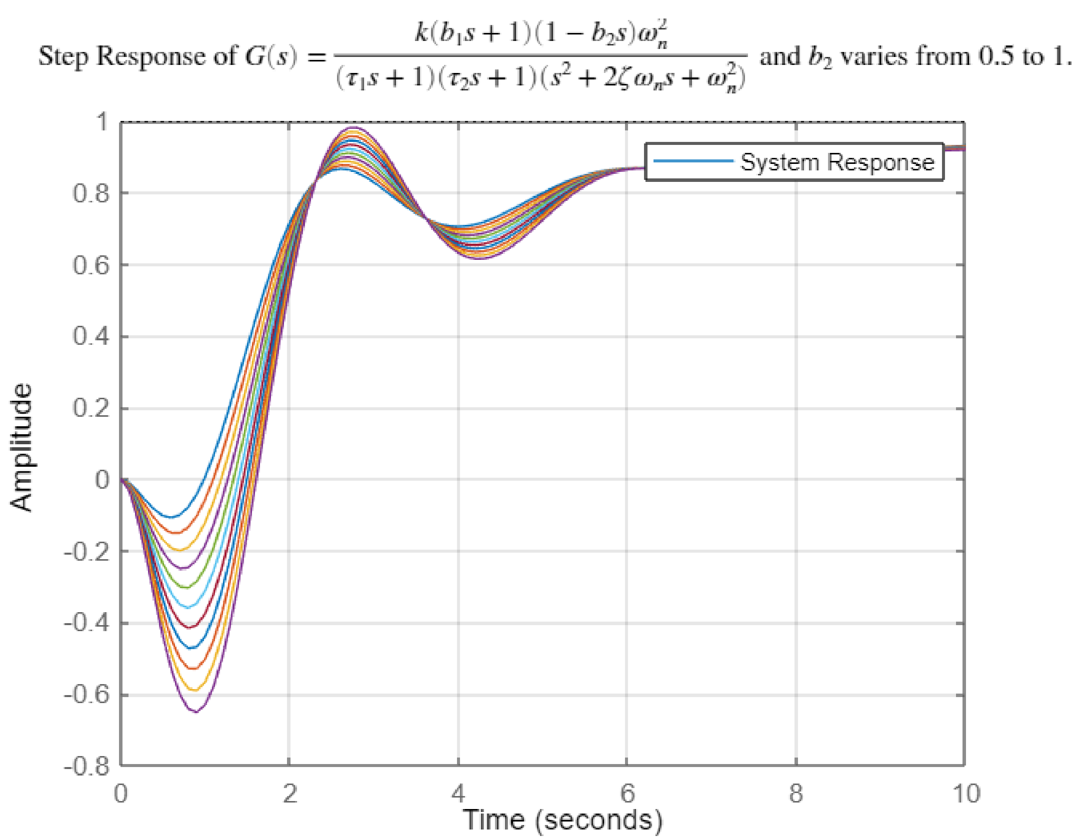
Problem 6:

part H:

Here we tuned b_2 from 1 to 3 or equivalently with 50 percent deviation in its ideal case:

```
clc; clear; close all;
syms s
```

```
% Defining the parameters of the transfer function
param = 0.5:0.1:1.5;
k = 1;
b1 = 3;
b2 = 1;
wn = 2;
tau1 = 0.5;
tau2 = 5;
zeta = 0.3;
% loop for deviating b2
for i = 1:length(param)
    hold on
    num = k*(b1*s + 1)*(1 - param(i) *s)*wn^2;
    den = (tau2*s+1)*(tau1*s+1)*(s^2 + 2*zeta*wn*s + wn^2);
    H_sys = tf(sym2poly(num), sym2poly(den));
    t = 0:0.1:10;
    step(H_sys, t);
    xlabel('Time');
    ylabel('Amplitude');
    title(['Step Response of $G(s) = \frac{k(b_1 s + 1)(1 - b_2 s)\omega_n^2}' ...
        '{( \tau_1 s + 1)(\tau_2 s + 1)(s^2 + 2\zeta\omega_n s + \omega_n^2)}$ and $b_2$ ' ...
        'varies from 0.5 to 1.5'], ...
        'interpreter', 'latex');
    grid on;
    legend('System Response');
end
```



Problem 7:

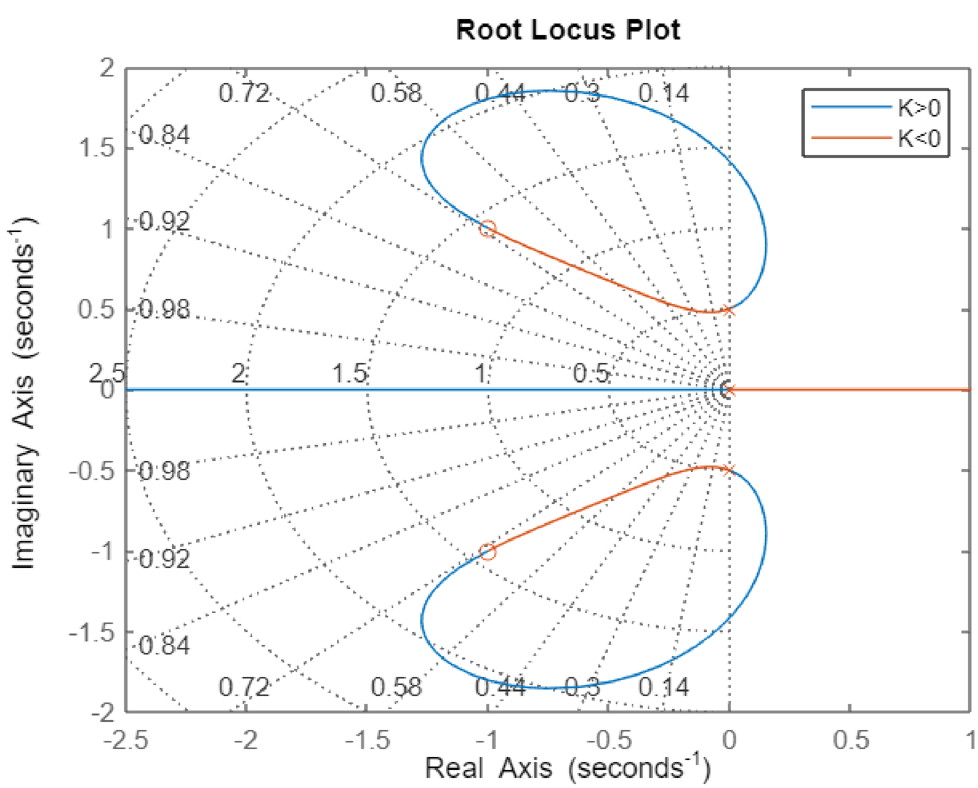
System 1:

Part A,B:

```
clc; clear; close all;

s = tf('s');
numer1 = s^2 + 2*s + 2;
denominat1 =  s^3 + 0.25*s;
sys1 = numer1/denominat1
```

```
sys1 =

  s^2 + 2 s + 2
  -------------
  s^3 + 0.25 s

Continuous-time transfer function.
```

```
figure;
rlocus(sys1,-sys1);
legend('K>0','K<0');
title('Root Locus Plot');
grid on
```



And here using the `find_breakaway_points(sys)` `function we took system's trtransferr function as an input and` we found the breakaway points by finding the roots of $\frac{d}{ds}G(s) = 0$:

Here is implementation of the desired function and also its output:

We also know that for finding the complementary rooy locus it suffices to take -G(s) as input:

```
% function [breakaway_points] = find_breakaway_points(sys)
%     syms s;
%     num = sys.Numerator;
%     den = sys.Denominator;
%     % Create symbolic expressions for the transfer function and its derivative
%     G = poly2sym(num, s) / poly2sym(den, s);
%     G_prime = diff(abs(G), s);
%
%     % Find the critical points where the derivative is zero
%     critical_points = vpasolve(G_prime == 0, s);
%
%     % Evaluate the critical points to determine breakaway points
%     breakaway_points = [];
%     for i = 1:length(critical_points)
%         if isreal(critical_points(i)) && abs(critical_points(i)) ~= Inf
%             breakaway_points = [breakaway_points; double(critical_points(i))];
%         end
%     end
% end
break_aways =  find_breakaway_points(sys1)
```

```
break_aways =

     []
```

Thus, there are no braking points in this system, since the list of `break_aways` is null.

System 2:

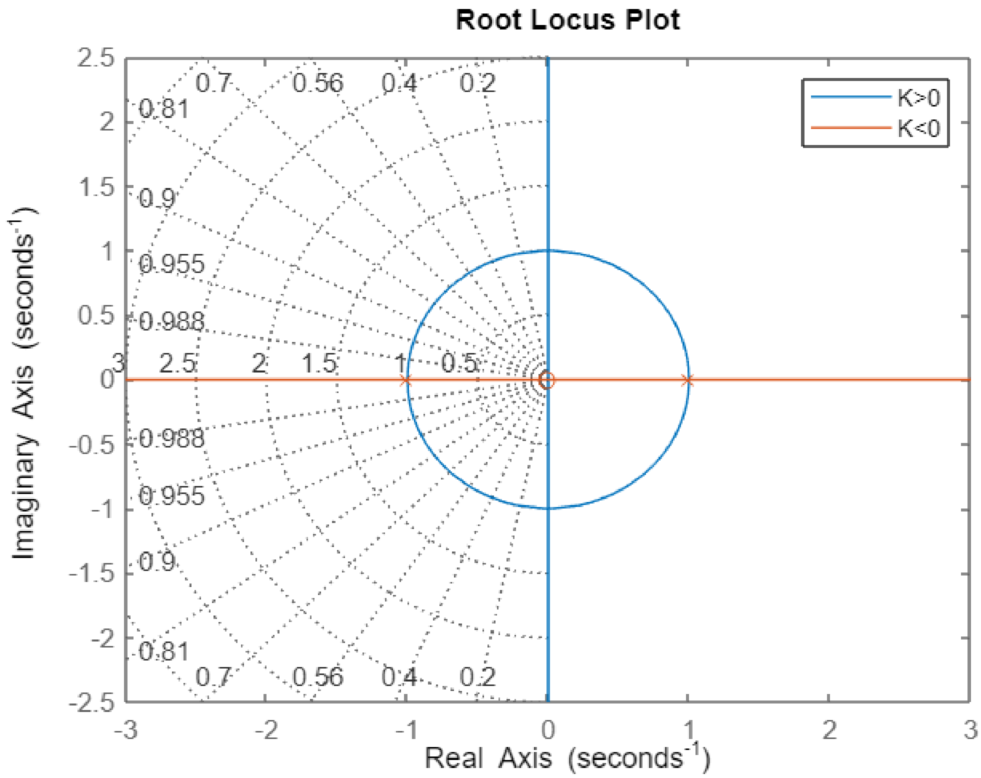
Part A,B:

```
clc; clear; close all;

s = tf('s');
numer2 = s^2;
denominat2 =  (s^2 - 1)^2;
sys2 = numer2/denominat2
```

```
sys2 =

         s^2
  ---------------
  s^4 - 2 s^2 + 1

Continuous-time transfer function.
```

```
figure;
rlocus(sys2,-sys2);
```

```
legend('K>0','K<0');
title('Root Locus Plot');
grid on
```



Part C:

System 1:

Here using the manipulator function we will output:

1)the points which belong to root-locus

2) The points which belong to complementary root-locus

3) The candidate points which neither belong to root-locus, nor to its complementary plot

```
clc; clear; close all;
syms s
G(s) = (s^2 + 2*s + 2)/(s *(s^2 + 0.25));
[k_pos ,k_neg, others] = Manipulator(G,s)
```

```
k_pos =

Empty sym: 0-by-1


k_neg =
```

```
Empty sym: 0-by-1

others =
```

$$\begin{pmatrix} -2.0291289456597803634180201034811 - 1.3357918919869738626969712269717\,i \\ -2.0291289456597803634180201034811 + 1.3357918919869738626969712269717\,i \\ 0.029128945659780363418020103481059 - 0.28960800211921472630369875126637\,i \\ 0.029128945659780363418020103481059 + 0.28960800211921472630369875126637\,i \end{pmatrix}$$

System 2:

```
clc; clear; close all;
syms s
G(s) = (s^2 / ((s^2-1)^2));
[k_pos ,k_neg, others] = Manipulator(G,s)
```

```
k_pos =

Empty sym: 0-by-1

k_neg =
```

$$\begin{pmatrix} -1.0\,i \\ 1.0\,i \end{pmatrix}$$

```
others =

Empty sym: 0-by-1
```

which in both cases alligns with what we have got in the first place.

```
function [breakaway_points] = find_breakaway_points(sys)
    syms s;
    num = sys.Numerator;
    den = sys.Denominator;
    % Create symbolic expressions for the transfer function and its derivative
    G = poly2sym(num, s) / poly2sym(den, s);
    G_prime = diff(abs(G), s);

    % Find the critical points where the derivative is zero
    critical_points = vpasolve(G_prime == 0, s);

    % Evaluate the critical points to determine breakaway points
    breakaway_points = [];
    for i = 1:length(critical_points)
        if isreal(critical_points(i)) && abs(critical_points(i)) ~= Inf
            breakaway_points = [breakaway_points; double(critical_points(i))];
        end
    end
end


function [k_pos,k_neg,others] = Manipulator(H_sys , s)
```

```
% we know that the breakaway points can be found
% using solving d/ds G(s) =0
sol = vpasolve(diff(H_sys , s) == 0,s);
point = H_sys(sol);
reals = (imag(point) == 0);
% first we will separate the complex roots
% since they must belong to the reral axis
others = sol(~reals);
idga = imag(point) == 0 & real(point) > 0;
idgc = imag(point) == 0 & real(point) < 0;
k_pos = sol(idga);
k_neg = sol(idgc);
end
```