PES University, Bangalore

(Established under Karnataka Act No. 16 of 2013)

UE21EC241B- CONTROL SYSTEMS

CS- PROJECT

Session: Jan-May 2023

Branch: ELECTRONICS AND COMMUNICATION ENGINEERING

Semester &Section:4TH SEM , A SECTION

Sl No.	Name of the Student	SRN	Marks Allotted (Out of 5)
1.	AKASH RAVI BHAT	PES1UG21EC025	
2.	ABHISHEK A. SHETTY	PES1UG21EC008	
3.	AKASH C.K.	PES1UG21EC022	

Name of the Course Instructor : **Prof. THIPPESWAMY E.**

Signature of the Course Instructor

(with Date) : ________________________________

ANALYSIS OF THE ROLL ANGLE CONTROLLER

- Roll angle control (RAC) is required for the lateral stability of an aircraft. Lateral stability makes the aircraft more stable around the longitudinal axis.
- Roll angle control makes both the wings of the aircraft to be at the same level.
- If one of the wing dips below the other, then RAC tries to stabilize the system again

1. The objective of this experiment is to analysis and design of control systems specific to a physical system. Each student will be given a specific physical system, and experiments are to be conducted on that particular physical system. (The specific physical system will be given to a student by the respective Teacher or Student can select the physical system by themselves.)

a. The objective of this exercise is to obtain the open loop characteristics of the given transfer function of the physical system or plant.

Aim or the outcome of the Project.

Regulate the bank angle of airplane to zero degrees and maintain the wings level orientation in the presence of unpredictable external disturbances.

(i)Where are the poles and zeros of the open loop system? (Exclude the controller, if considered in your

CODE:-

```
clc;

clear all;

close all;

%Defining the open loop transfer function.

num =[36.6];

den=[1 9.2 15.4 0];

sys= tf(num, den)

% Find's the poles and zeros of open loop system

P =pole(sys)
```

```
z = zero(sys)
```

OUTPUT:-

```
sys =

          36.6
  ----------------------
  s^3 + 9.2 s^2 + 15.4 s

Continuous-time transfer function.
P = 3x1
          0
     -7.0000
     -2.2000

z =

  0x1 empty double column vector
```

Obtain the ‘pole-zero’ map for the open loop system.

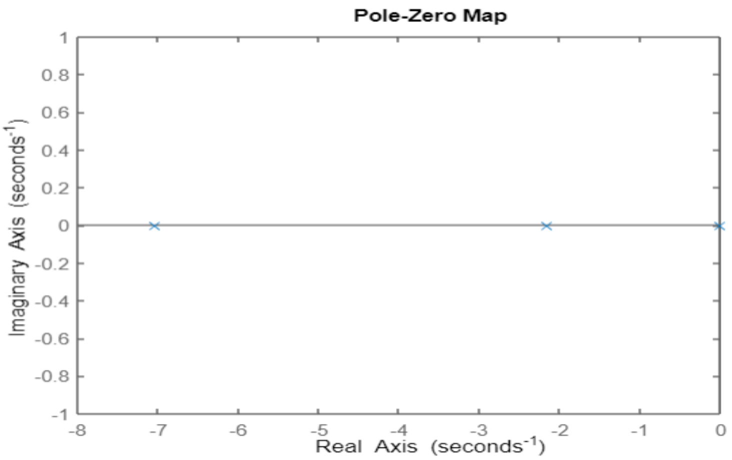
Code:

```
clc;

clear all;

close all;

%Defining the open loop transfer function

num= [36.6];

den = [1 9.2 15.4 0];

sys =tf (num, den)

% Find's the poles and zeros of open loop system

p = pole(sys)

z = zero(sys)
```

```
% Plot the pole-zero map

figure;

pzmap (sys);

title('Pole-Zero Map');
```

OUTPUT:-

```
sys =

            36.6
  ----------------------
  s^3 + 9.2 s^2 + 15.2 s

Continuous-time transfer function.
p = 3×1
            0
      -7.0413
      -2.1587

z =

  0×1 empty double column vector
```

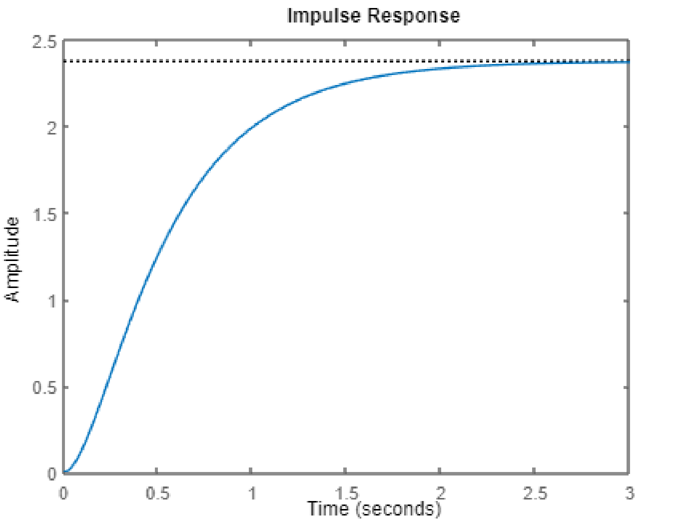


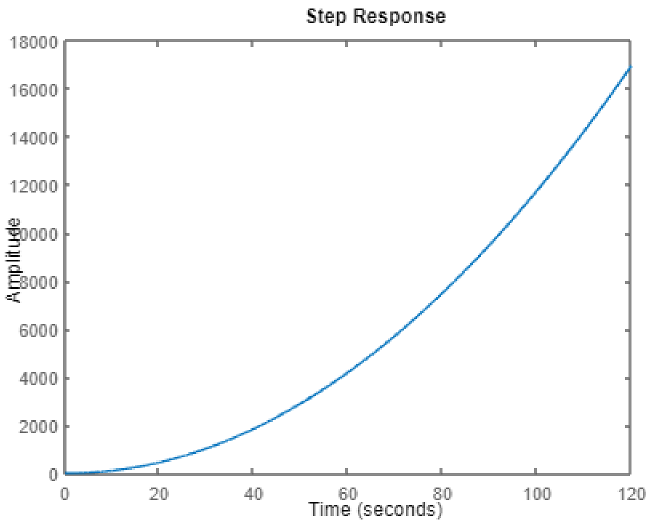
(ii) Apply different test signals, and observe the timedomain response. Discuss the results obtained from the viewpoint of pole-zero map.

MATLAB CODE

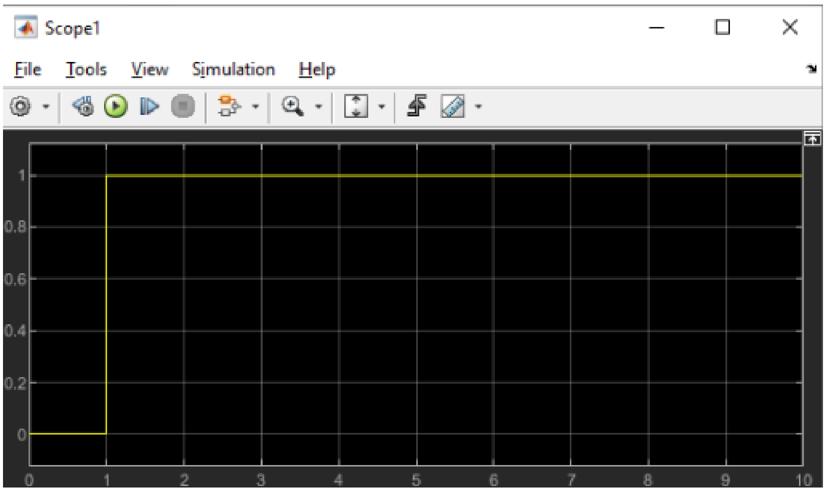
```
s = tf('s')

num = [ 0 0 36.6];

den = [1 9.2 15.4 0];

TF = tf(num,den)

figure

pzmap(TF)

figure

step(TF)

figure

impulse(TF)

figure

title('Ramp response')

step(TF/s) %ramp response
```

MATLAB OUTPUT

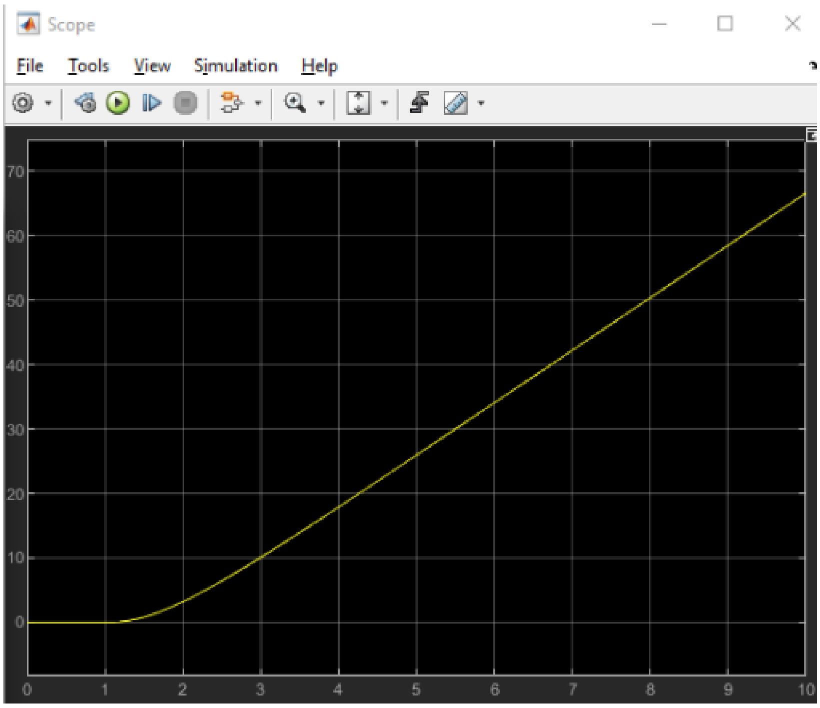

Impulse Response
Amplitude
Time (seconds)
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3


Step Response
Amplitude
Time (seconds)
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
0
20
40
60
80
100
120

Step input:



OUTPUT:-

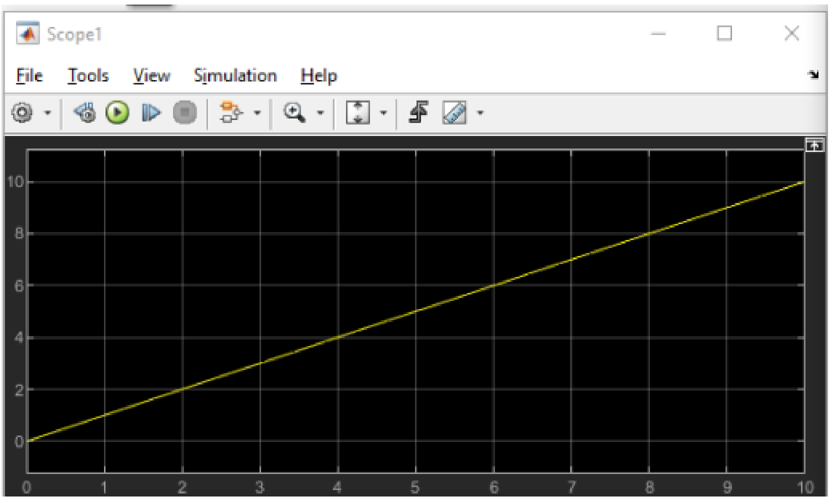


CODE:-

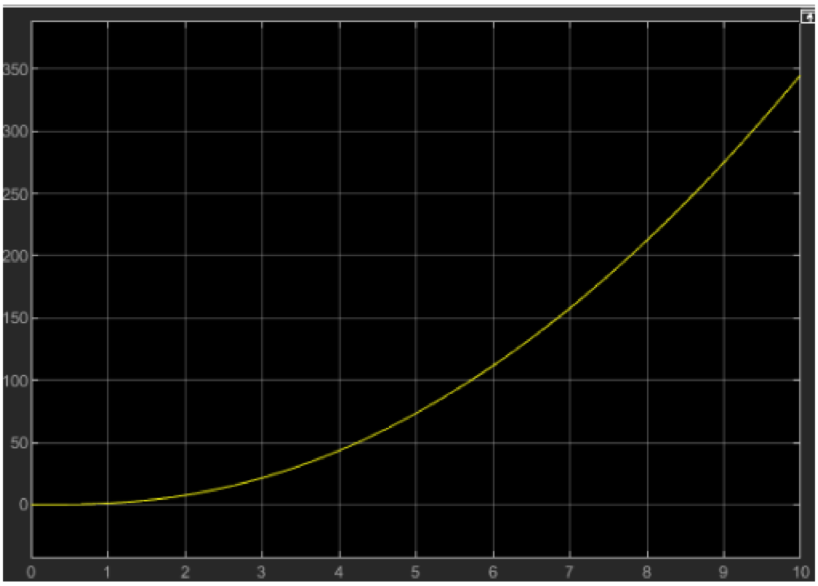
```
num= [36.6];
den = [1 9.2 15.4 0];
sys=tf (num, den)
stepinfo(sys)
```

```
sys =

           36.6
  ----------------------
  s^3 + 9.2 s^2 + 15.4 s

Continuous-time transfer function.
ans = struct with fields:
         RiseTime: NaN
    TransientTime: NaN
     SettlingTime: NaN
      SettlingMin: NaN
      SettlingMax: NaN
        Overshoot: NaN
       Undershoot: NaN
             Peak: Inf
         PeakTime: Inf
```

RAMP INPUT



OUTPUT:-



2. The objective of this exercise is to determine the range of a gain that assures closed loop stability. Assume that the given system is part of a unity negative feedback system, and there is a gain in cascade with the given system in the forward path. Conduct experiments similar to (project2-1) and determine the range of k for which the closed loop system is stable

Matlab code

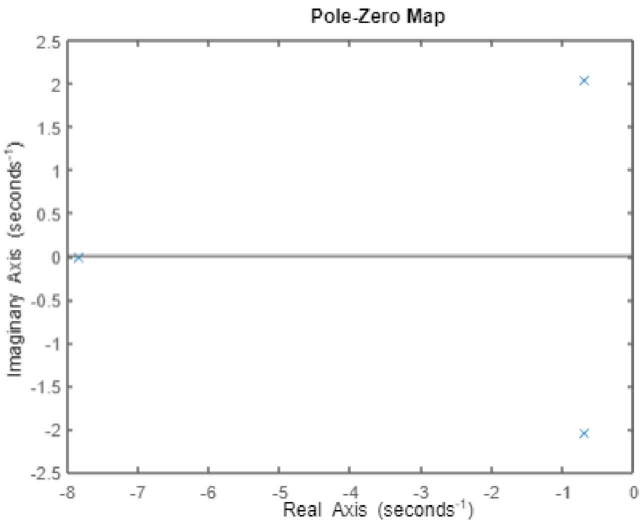
```
k = [1:1:5];


num = [0 0 0 36.6];
den = [1 9.2 15.4 0];


n1 = conv(num,k(1));
n2 = conv(num,k(2));
n3 = conv(num,k(3));
n4 = conv(num,k(4));
n5 = conv(num,k(5));
d1 = conv(den,1);
```

```
[num1,den1] = negfeedback(n1,1,d1,1);
[num2,den2] = negfeedback(n2,1,d1,1);
[num3,den3] = negfeedback(n3,1,d1,1);
[num4,den4] = negfeedback(n4,1,d1,1);
[num5,den5] = negfeedback(n5,1,d1,1);

tf1 = tf(num1,den1);
tf2 = tf(num2,den2);
tf3 = tf(num3,den3);
tf4 = tf(num4,den4);
tf5 = tf(num5,den5);

figure
pzplot(tf1)
figure
pzplot(tf2)
figure
pzplot(tf3)
figure
pzplot(tf4)
```

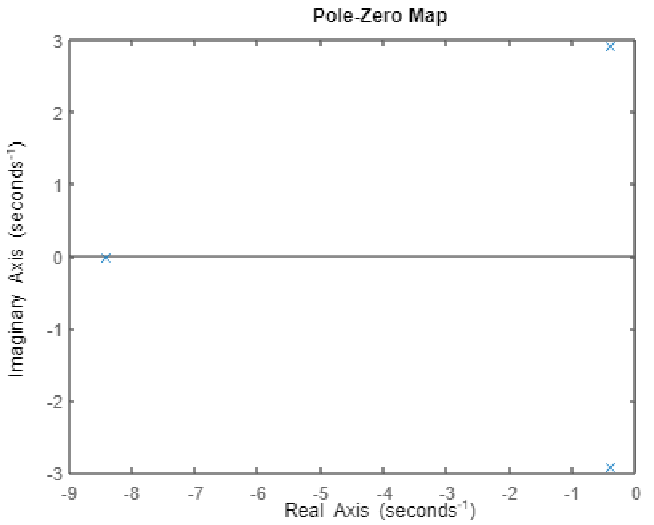
```
figure
pzplot(tf5)


function [num,den] = negfeedback(n1,n2,d1,d2)
num = conv(n1,d2);
den = conv(d1,d2)+conv(n1,n2);
end
```

output:-

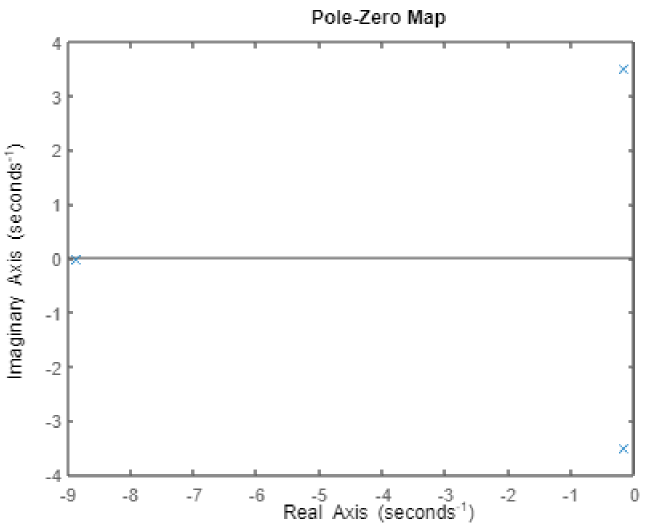
for k=1



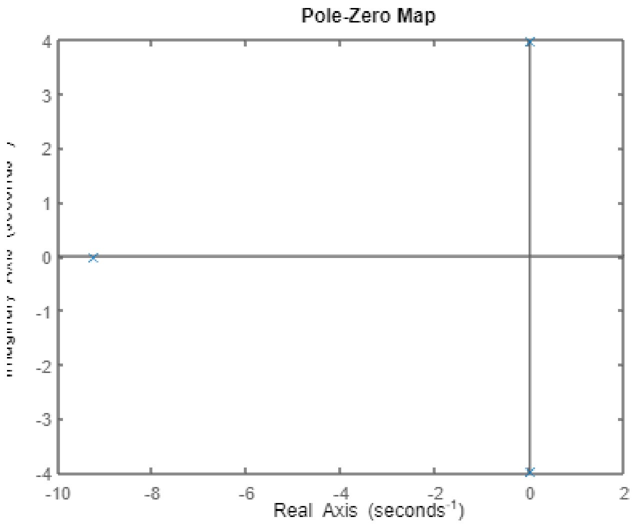
For k=2


Pole-Zero Map
Imaginary Axis (seconds⁻¹)
Real Axis (seconds⁻¹)
3
2
1
0
-1
-2
-3
-9
-8
-7
-6
-5
-4
-3
-2
-1
0

K=3


Pole-Zero Map
Imaginary Axis (seconds⁻¹)
Real Axis (seconds⁻¹)
4
3
2
1
0
-1
-2
-3
-4
-9
-8
-7
-6
-5
-4
-3
-2
-1
0

For k=4



3. The objective of the exercise to analyse the closed loop system behaviour with proportional controller of the system whose transfer function you were given earlier. a. Place a gain k in the forward path, and close the loop with negative unity feedback. Take different values for k. For each value of in this set, obtain the step response. What is the rise time, the settling time? Are there any oscillations? If so, what is the frequency of oscillation? Compare the response of the closed loop system to the open loop system. Compare the closed loop responses. Discuss the results. Can we increase k indefinitely? b. Obtain the root locus. Mark the earlier choices of k on the root locus. Discuss the results obtained from the root locus with reference to those obtained with k in the forward path in part 3(a).

MATLAB CODE

```
k = [1:1:5];
```

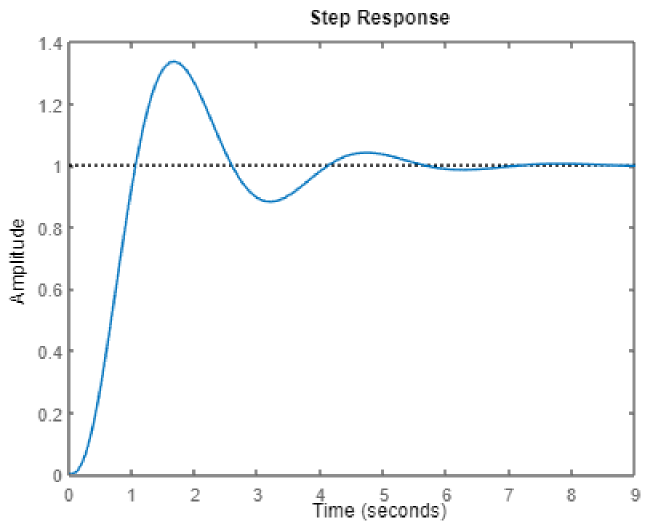
```
num = [0 0 0 36.6];
den = [1 9.2 15.4 0];


n1 = conv(num,k(1));
n2 = conv(num,k(2));
n3 = conv(num,k(3));
n4 = conv(num,k(4));
n5 = conv(num,k(5));
d1 = conv(den,1);


[num1,den1] = negfeedback(n1,1,d1,1);
[num2,den2] = negfeedback(n2,1,d1,1);
[num3,den3] = negfeedback(n3,1,d1,1);
[num4,den4] = negfeedback(n4,1,d1,1);
[num5,den5] = negfeedback(n5,1,d1,1);


tf1 = tf(num1,den1);
tf2 = tf(num2,den2);
tf3 = tf(num3,den3);
tf4 = tf(num4,den4);
tf5 = tf(num5,den5);


 figure
 step(tf1)
 stepinfo(tf1)

 figure
 step(tf2)
 stepinfo(tf2)


 figure
 step(tf3)
 stepinfo(tf3)


figure
step(tf4)
stepinfo(tf4)


tf11 = tf(n1,d1);
tf22 = tf(n2,d1);
tf33 = tf(n3,d1);
tf44 = tf(n4,d1);


figure
```

```
rlocus(tf11)
rlocus(tf22)
rlocus(tf33)
rlocus(tf44)
```

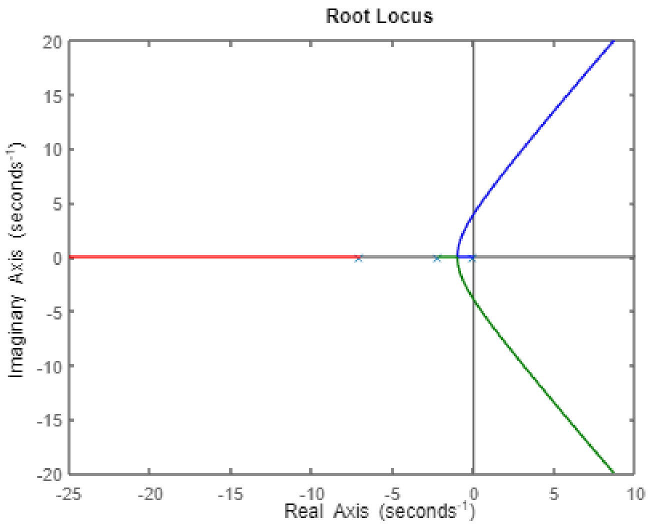
```
function [num,den] = negfeedback(n1,n2,d1,d2)
num = conv(n1,d2);
den = conv(d1,d2)+conv(n1,n2);
end
```

MATLAB OUTPUT

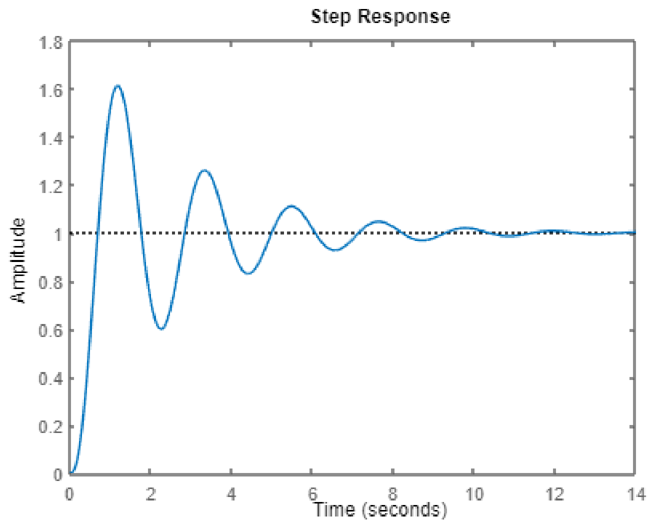
FOR K=1



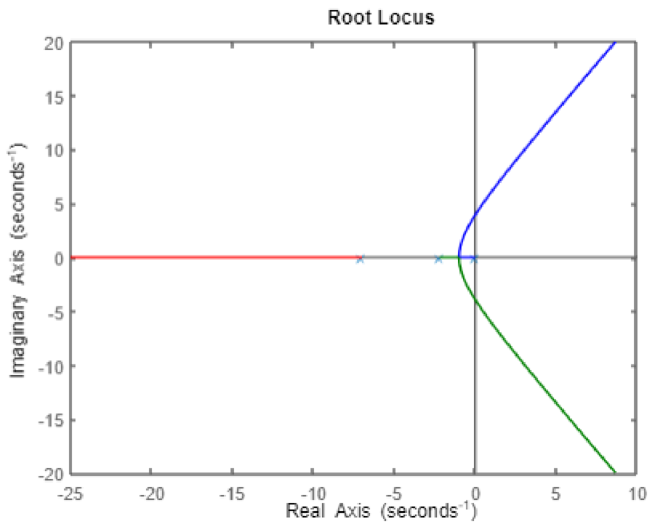

```
ans = struct with fields:
        RiseTime: 0.6592
   TransientTime: 5.2933
    SettlingTime: 5.2933
     SettlingMin: 0.8825
     SettlingMax: 1.3359
       Overshoot: 33.5871
      Undershoot: 0
            Peak: 1.3359
        PeakTime: 1.6810
```



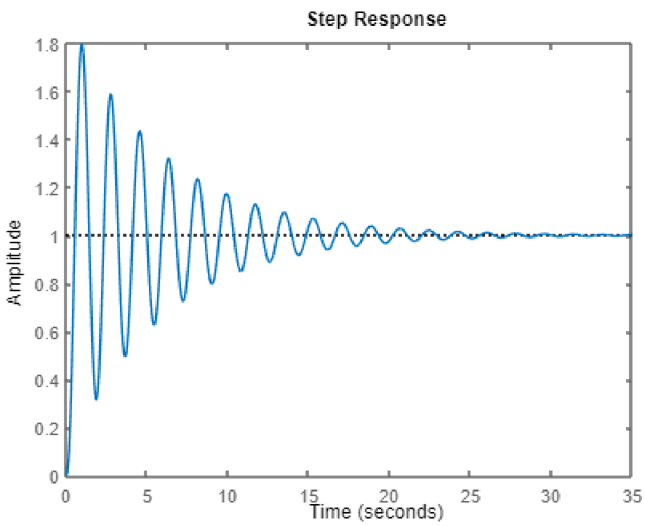
FOR K=2



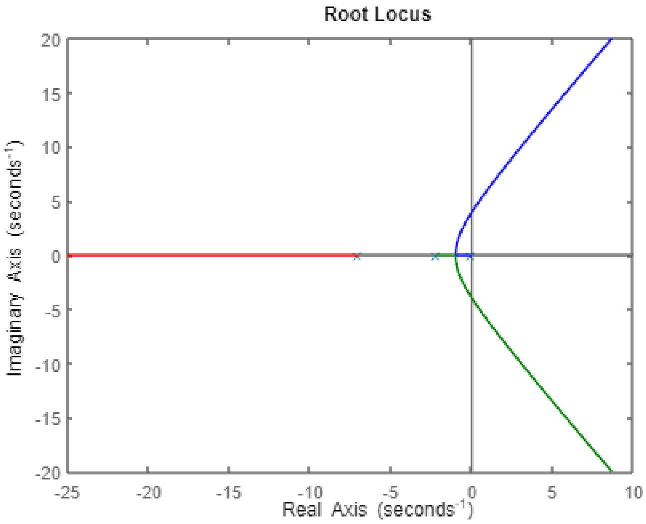

```
= struct with fields:
      RiseTime: 0.4193
 TransientTime: 9.8096
  SettlingTime: 9.8096
   SettlingMin: 0.6003
   SettlingMax: 1.6130
     Overshoot: 61.3017
    Undershoot: 0
          Peak: 1.6130
      PeakTime: 1.1946
```



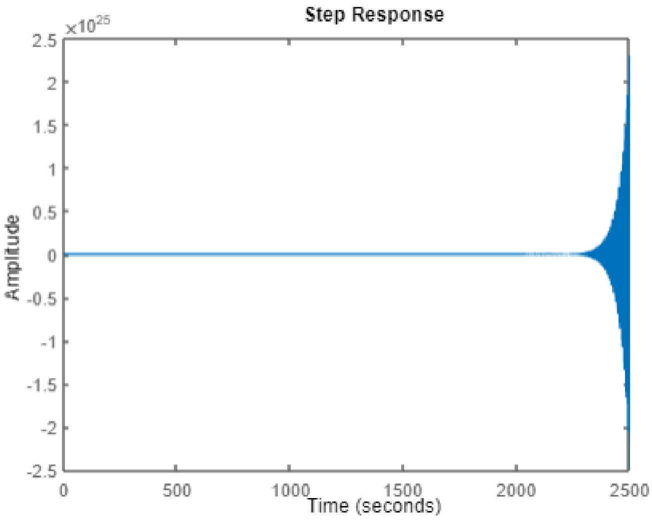
FOR K=3



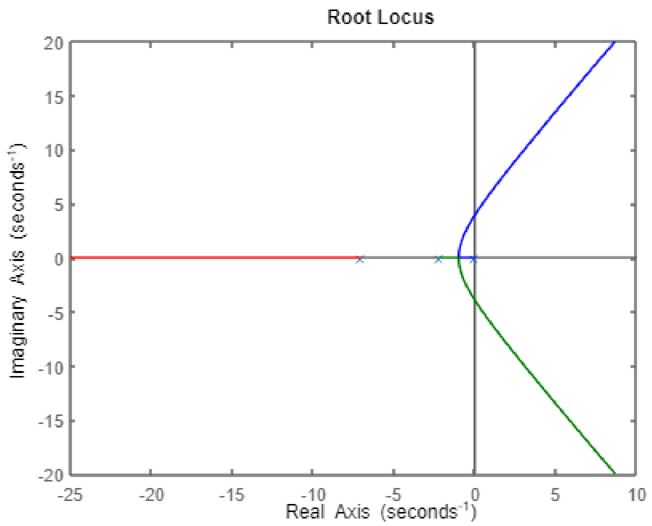

```
ans = struct with fields:
        RiseTime: 0.3330
   TransientTime: 22.5317
    SettlingTime: 22.5317
     SettlingMin: 0.3151
     SettlingMax: 1.7970
       Overshoot: 79.7043
      Undershoot: 0
            Peak: 1.7970
        PeakTime: 0.9979
```



FOR K=4




```
ans = struct with fields:
        RiseTime: NaN
   TransientTime: NaN
    SettlingTime: NaN
     SettlingMin: NaN
     SettlingMax: NaN
       Overshoot: NaN
      Undershoot: NaN
            Peak: Inf
        PeakTime: Inf
```



4. The objective of this exercise is to obtain the closed loop behaviour with proportional plus derivative controller of the system you were given earlier. Place a function K (s + z) in the forward path, and close the loop with negative unity feedback. Take different values for K and z. For each sets of (K,z) obtain the step response, and the rlocus. Compare the step response for each case, and compare with the case of putting only a gain K in the forward path. What is therefore the effect of adding a zero in the forward path? Are there any additional insight to be gained from the rlocus: Obtain the root locus for each case. Compare the three loci, and discuss the results.

MATLAB CODE

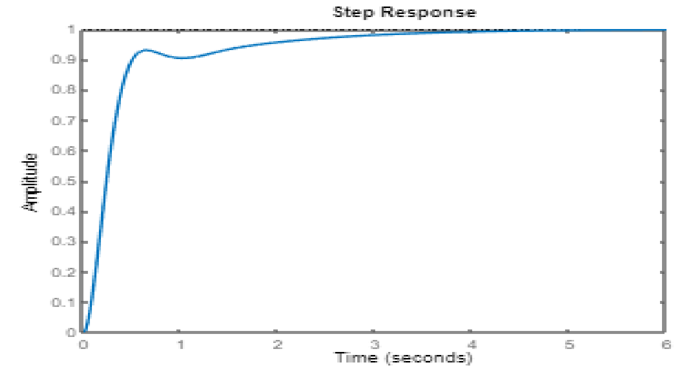
```
num = [0 0 0 36.6];
den = [0 1 9.2 15.4 0];
```

```
 k=1,z=1
 n1 = conv([1 1],num);
 [num1,den1] = negfeedback(n1,1,den,1);
 tf1 = tf(num1,den1);
tf11 = tf(n1,den);

 figure;
 step(tf1)
 stepinfo(tf1)
 figure;
 rlocus(tf11)


 k=2,z=1
 n2 = conv([2 2],num);
 [num2,den2] = negfeedback(n2,1,den,1);
 tf2 = tf(num2,den2);
 tf22 = tf(n2,den);

 figure;
 step(tf2)
 stepinfo(tf2)
 figure;
rlocus(tf22)

  k=1,z=2
 n3 = conv([1 2],num);
 [num3,den3] = negfeedback(n3,1,den,1);
 tf3 = tf(num3,den3);
 tf33 = tf(n3,den);

 figure;
 step(tf3)
 stepinfo(tf3)
 figure;
 rlocus(tf33)

 k=3,z=2
n4 = conv([3 6],num);
 [num4,den4] = negfeedback(n4,1,den,1);
 tf4 = tf(num4,den4);
 tf44 = tf(n4,den);

 figure;
 step(tf4)
 stepinfo(tf4)
figure;
 rlocus(tf44)

 k=5,z=7
 n5 = conv([5 35],num);
```

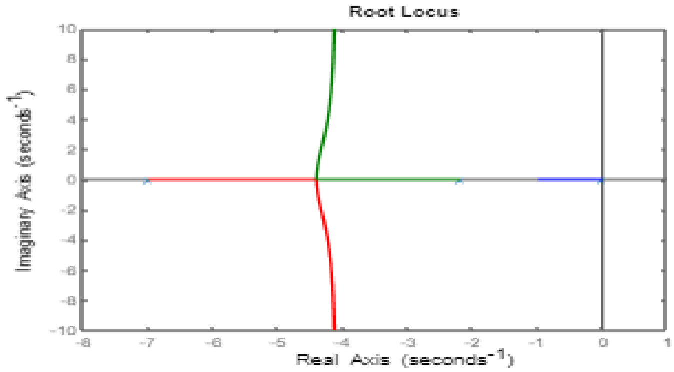
```
 [num5,den5] = negfeedback(n5,1,den,1);
 tf5 = tf(num5,den5);
 tf55 = tf(n5,den);

 figure;
 step(tf5)
 stepinfo(tf5)
 figure;
 rlocus(tf55)

  k=9,z=13
 n6 = conv([9 117],num);
 [num6,den6] = negfeedback(n6,1,den,1);
 tf6 = tf(num6,den6);
 tf66 = tf(n6,den);


 figure;
 step(tf6)
 stepinfo(tf6)
 figure;
 rlocus(tf66)

 k=2,z=13
n7 = conv([2 26],num);
[num7,den7] = negfeedback(n7,1,den,1);
tf7 = tf(num7,den7);
tf77 = tf(n7,den);


figure;
step(tf7)
stepinfo(tf7)
figure;
rlocus(tf77)
```

```
function [num,den] = negfeedback(n1,n2,d1,d2)
num = conv(n1,d2);
den = conv(d1,d2)+conv(n1,n2);
end
```

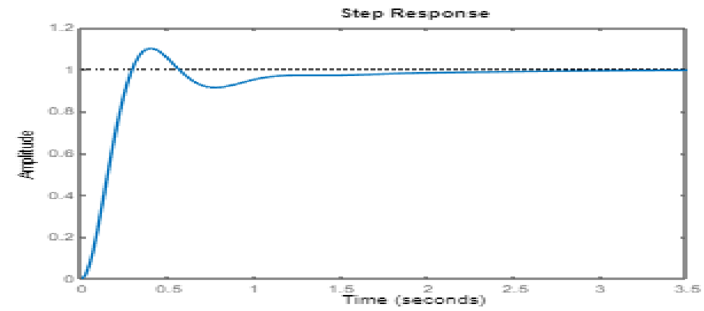
```
k = 1
z = 1
```



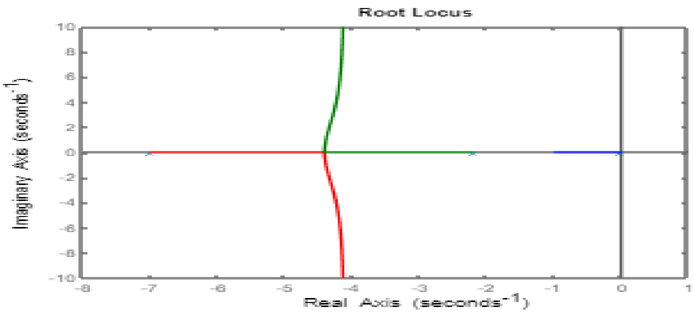

```
ans = struct with fields:
        RiseTime: 0.4233
   TransientTime: 2.9633
    SettlingTime: 2.9633
     SettlingMin: 0.9043
     SettlingMax: 0.9985
       Overshoot: 0
      Undershoot: 0
            Peak: 0.9985
        PeakTime: 6.1253
```



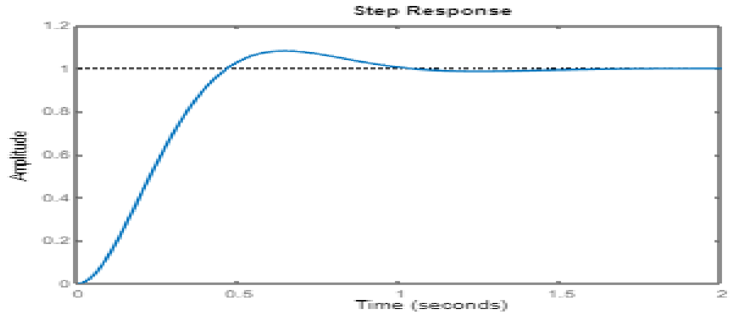
```
k = 2
z = 1
```



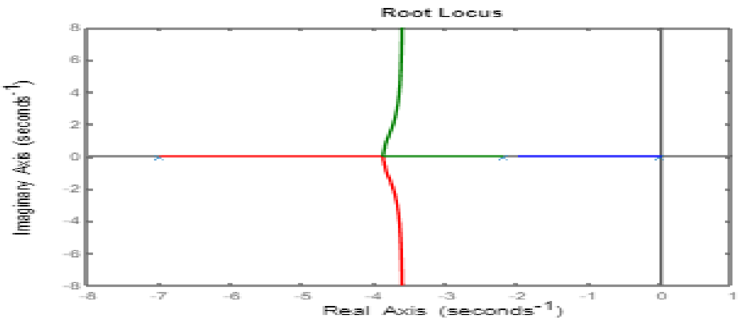

```
ans = struct with fields:
        RiseTime: 0.1966
   TransientTime: 1.8474
    SettlingTime: 1.8474
     SettlingMin: 0.9132
     SettlingMax: 1.1007
       Overshoot: 10.0727
      Undershoot: 0
            Peak: 1.1007
        PeakTime: 0.3996
```



```
k = 1
z = 2
```

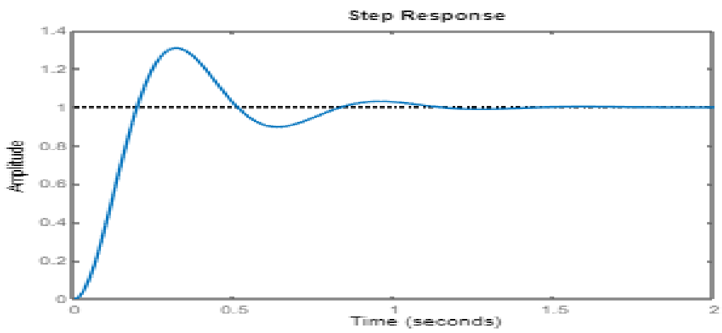



```
ans = struct with fields:
        RiseTime: 0.3091
   TransientTime: 0.9239
    SettlingTime: 0.9239
     SettlingMin: 0.9028
     SettlingMax: 1.0810
       Overshoot: 8.1000
      Undershoot: 0
            Peak: 1.0810
        PeakTime: 0.6459
```

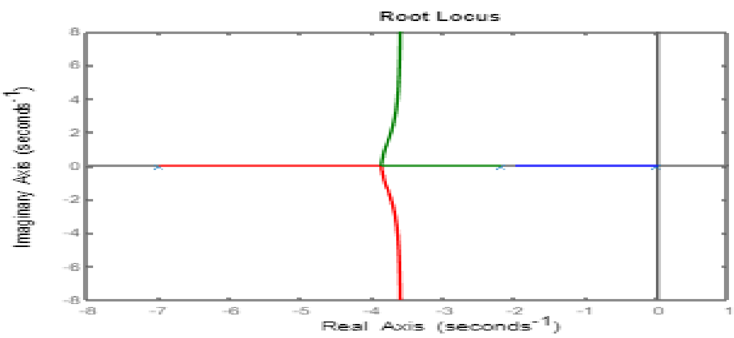


Real Axis (seconds $^{-1}$)

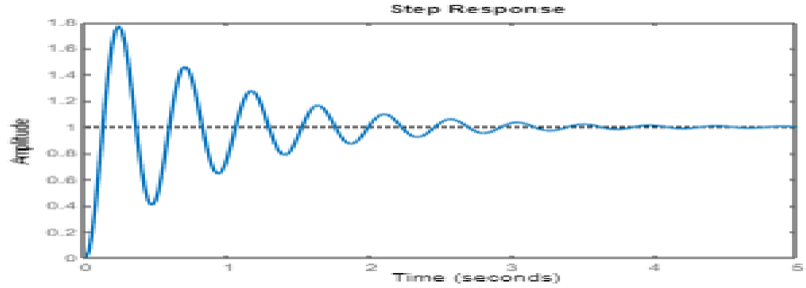
```
k = 3
z = 2
```



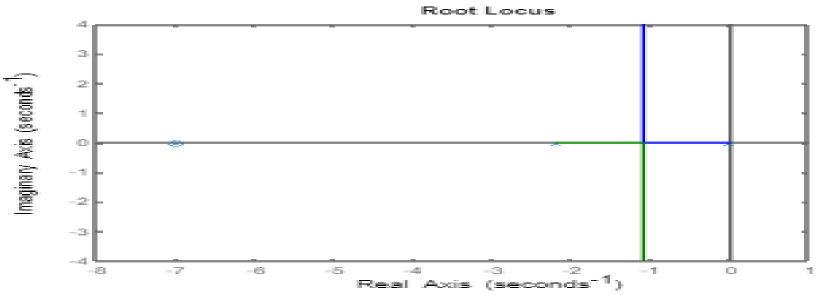

```
ans = struct with fields:
        RiseTime: 0.1324
   TransientTime: 1.0404
    SettlingTime: 1.0404
     SettlingMin: 0.8970
     SettlingMax: 1.3084
       Overshoot: 30.8438
      Undershoot: 0
            Peak: 1.3084
        PeakTime: 0.3189
```



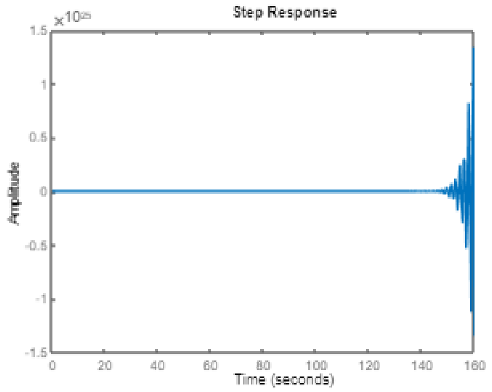
```
k = 5
z = 7
```



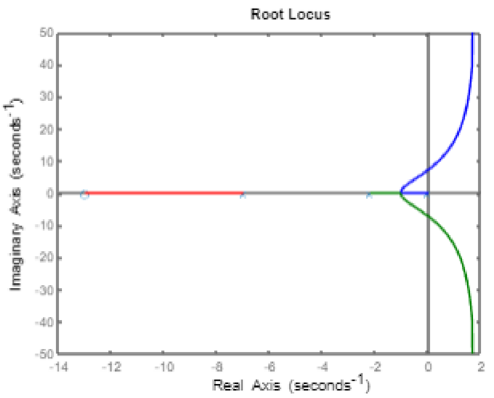

```
ans = struct with fields:
        RiseTime: 0.0821
   TransientTime: 3.5198
    SettlingTime: 3.5198
     SettlingMin: 0.4012
     SettlingMax: 1.7739
       Overshoot: 77.3863
      Undershoot: 0
            Peak: 1.7739
        PeakTime: 0.2322
```



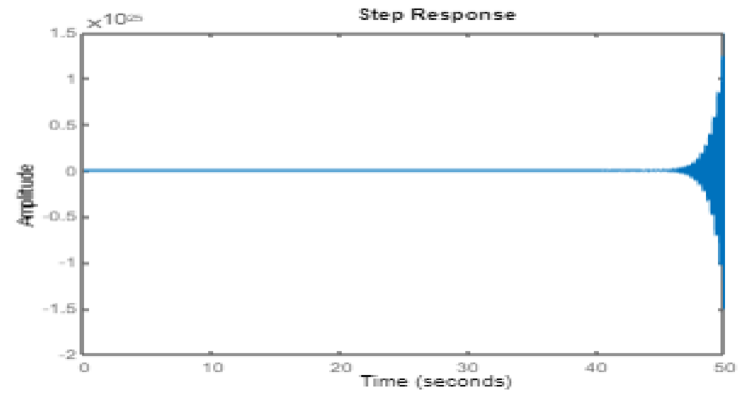

```
k = 2
z = 13
```



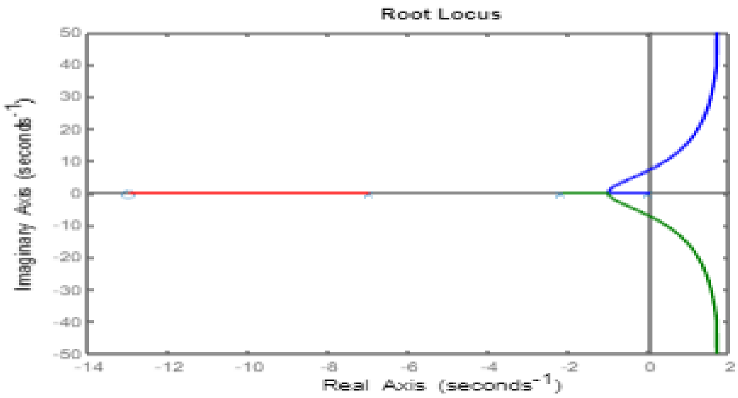

```
ans = struct with fields:
        RiseTime: NaN
   TransientTime: NaN
    SettlingTime: NaN
     SettlingMin: NaN
     SettlingMax: NaN
       Overshoot: NaN
      Undershoot: NaN
            Peak: Inf
        PeakTime: Inf
```



```
k = 9
z = 13
```




```
ans = struct with fields:
        RiseTime: NaN
   TransientTime: NaN
    SettlingTime: NaN
     SettlingMin: NaN
     SettlingMax: NaN
       Overshoot: NaN
      Undershoot: NaN
            Peak: Inf
        PeakTime: Inf
```



5. (a)The objective of this exercise is to obtain the closed loop behaviour with proportional plus integral controller of the system you were given earlier. (i) Place a function $((s + z)/s)$ in the forward path, and close the loop with negative unity feedback. Take different values for K and z, For each sets of (K, z)obtain the step response, and the root locus. Compare the step response for each case, and compare with the case of putting only a gain Kin the forward path and $(s + z)$. What is therefore the effect of adding a pole in the forward path? Are there any

additional insights to be gained from the root locus. Compare the three loci, and discuss the results. Hence (ii) infer the results if the following function $K1 + K2\ s$ + $K3s$ is placed in the forward path. Substantiate your answer for suitable choices of $K1$,2 and $K3$

MATLAB CODE

```
num = [0 0 0 36.6];
den = [1 9.2 15.4 0 0];


k=1,z=1
n1 = conv([1 1],num);
[num1,den1] = negfeedback(n1,1,den,1);
tf1 = tf(num1,den1);
tf11 = tf(n1,den);


figure;
step(tf1)
stepinfo(tf1)
figure;
rlocus(tf11)

k=2,z=1
n2 = conv([2 2],num);
[num2,den2] = negfeedback(n2,1,den,1);
tf2 = tf(num2,den2);
tf22 = tf(n2,den);


figure;
step(tf2)
stepinfo(tf2)
figure;
rlocus(tf22)


k=1,z=2
n3 = conv([1 2],num);
[num3,den3] = negfeedback(n3,1,den,1);
tf3 = tf(num3,den3);
tf33 = tf(n3,den);


figure;
```

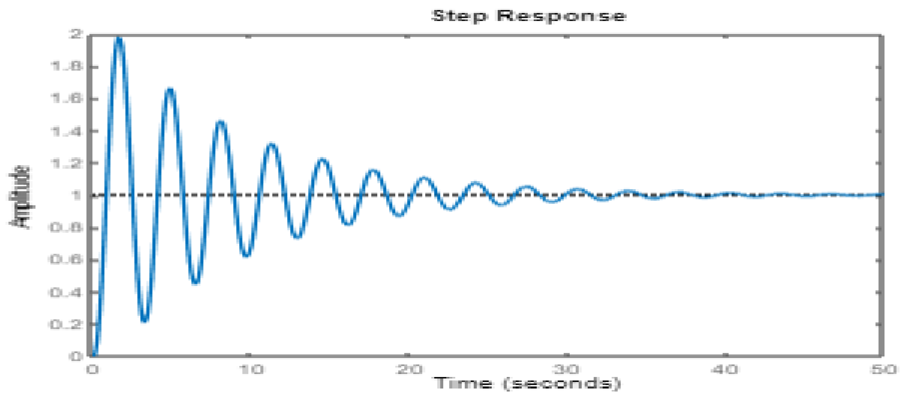
```
step(tf3)
stepinfo(tf3)
figure;
rlocus(tf33)


k=3,z=2
n4 = conv([3 6],num);
[num4,den4] = negfeedback(n4,1,den,1);
tf4 = tf(num4,den4);
tf44 = tf(n4,den);


figure;
step(tf4)
stepinfo(tf4)
figure;
rlocus(tf44)


k=5,z=7
n5 = conv([5 35],num);
[num5,den5] = negfeedback(n5,1,den,1);
tf5 = tf(num5,den5);
tf55 = tf(n5,den);


figure;
step(tf5)
stepinfo(tf5)
figure;
rlocus(tf55)


k=9,z=13
n6 = conv([9 117],num);
[num6,den6] = negfeedback(n6,1,den,1);
tf6 = tf(num6,den6);
tf66 = tf(n6,den);


figure;
step(tf6)
stepinfo(tf6)
figure;
rlocus(tf66)


k=2,z=13
n7 = conv([2 26],num);
[num7,den7] = negfeedback(n7,1,den,1);
tf7 = tf(num7,den7);
```

```
tf77 = tf(n7,den);


figure;
step(tf7)
stepinfo(tf7)
figure;
rlocus(tf77)
```

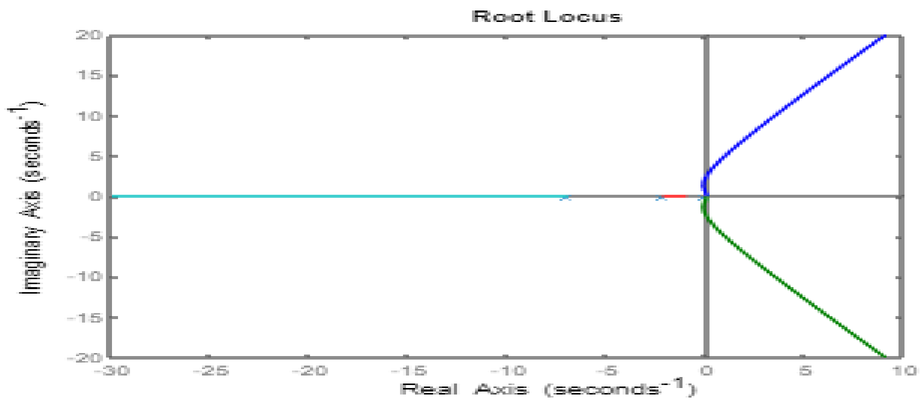
```
function [num,den] = negfeedback(n1,n2,d1,d2)
num = conv(n1,d2);
den = conv(d1,d2)+conv(n1,n2);
end
```

MATLAB OUTPUT:-

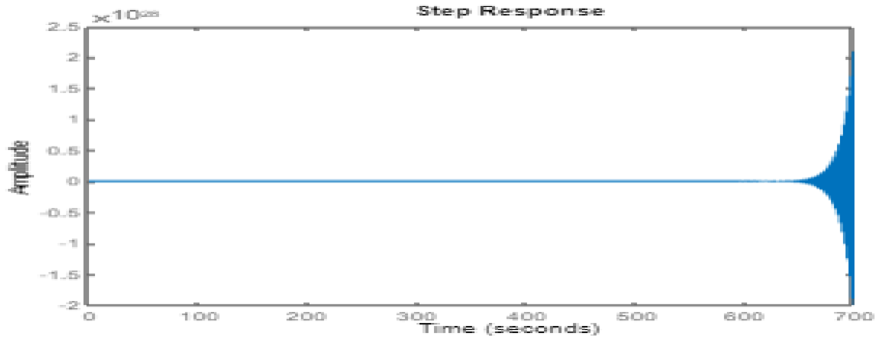
```
k = 1
z = 1
```



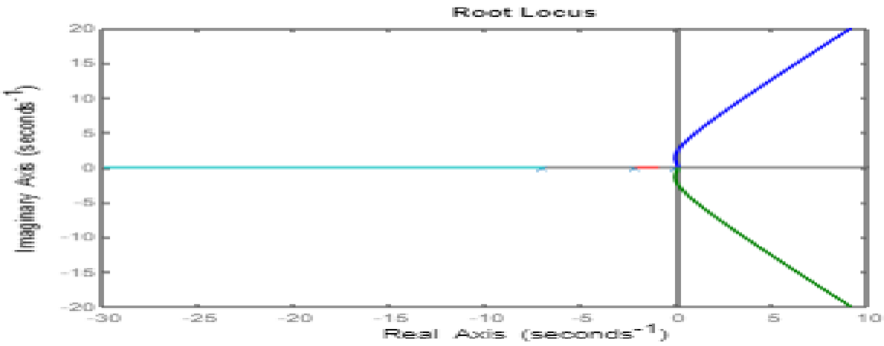

```
ans = struct with fields:
        RiseTime: 0.5224
   TransientTime: 35.5850
    SettlingTime: 35.5850
     SettlingMin: 0.2080
     SettlingMax: 1.9781
       Overshoot: 97.8142
      Undershoot: 0
            Peak: 1.9781
        PeakTime: 1.6454
```



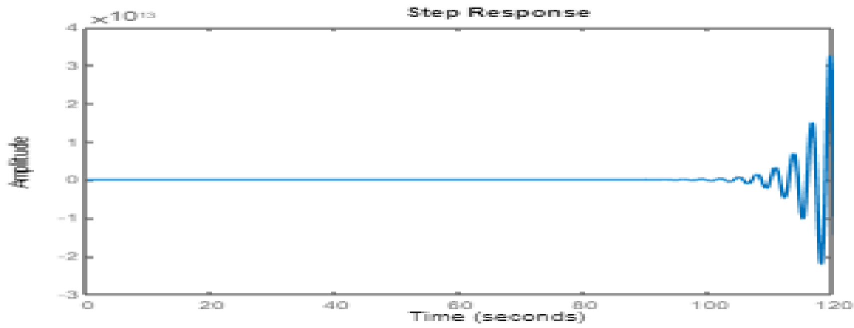
```
k = 2
z = 1
```



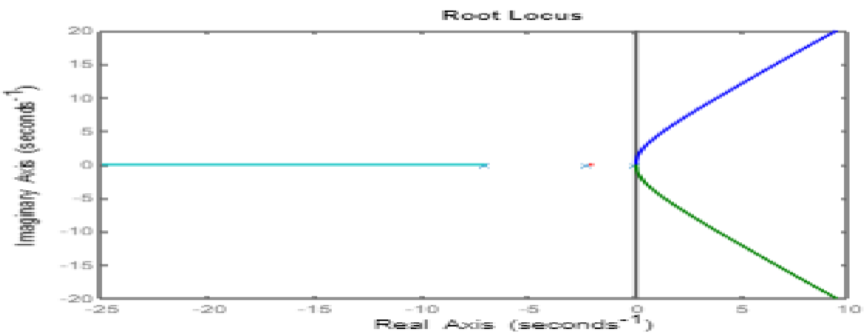

```
ans = struct with fields:
        RiseTime: NaN
   TransientTime: NaN
    SettlingTime: NaN
     SettlingMin: NaN
     SettlingMax: NaN
       Overshoot: NaN
      Undershoot: NaN
            Peak: Inf
        PeakTime: Inf
```



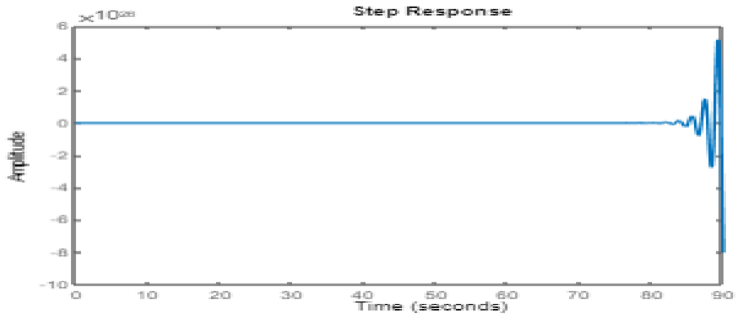

```
k = 1
z = 2
```



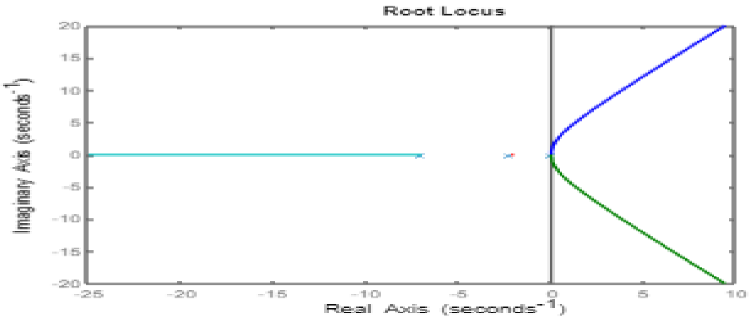

```
ans = struct with fields:
        RiseTime: NaN
   TransientTime: NaN
    SettlingTime: NaN
     SettlingMin: NaN
     SettlingMax: NaN
       Overshoot: NaN
      Undershoot: NaN
            Peak: Inf
        PeakTime: Inf
```



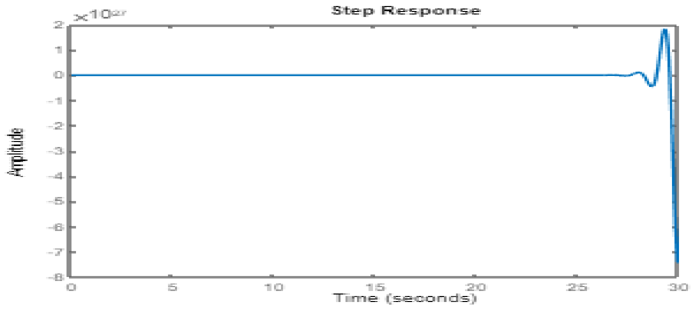
```
k = 3
z = 2
```



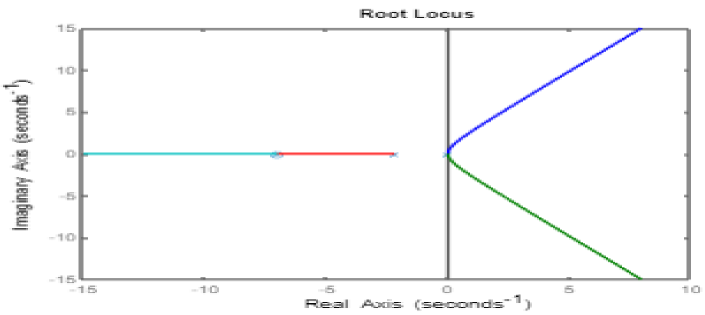

```
ans = struct with fields:
        RiseTime: NaN
   TransientTime: NaN
    SettlingTime: NaN
     SettlingMin: NaN
     SettlingMax: NaN
       Overshoot: NaN
      Undershoot: NaN
            Peak: Inf
        PeakTime: Inf
```



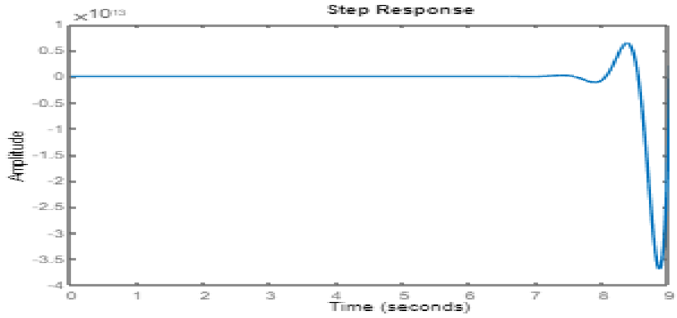

```
k = 5
z = 7
```



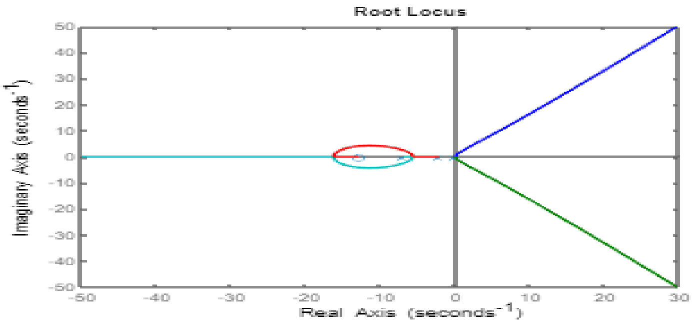

```
ans = struct with fields:
        RiseTime: NaN
   TransientTime: NaN
    SettlingTime: NaN
     SettlingMin: NaN
     SettlingMax: NaN
       Overshoot: NaN
      Undershoot: NaN
            Peak: Inf
        PeakTime: Inf
```



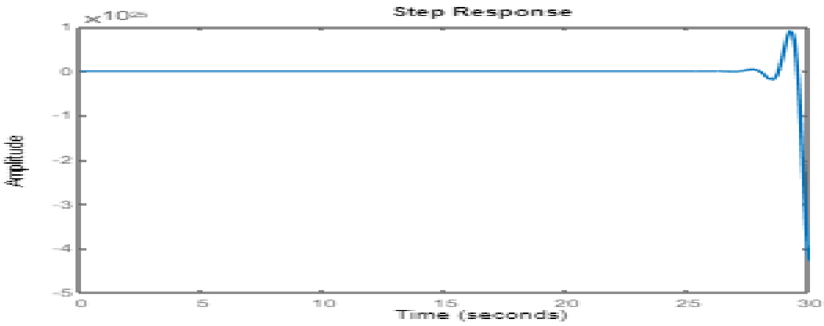
```
k = 9
z = 13
```



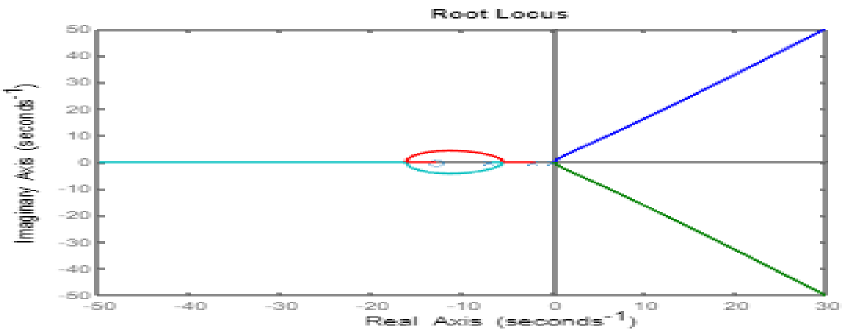

```
ans = struct with fields:
        RiseTime: NaN
   TransientTime: NaN
    SettlingTime: NaN
     SettlingMin: NaN
     SettlingMax: NaN
       Overshoot: NaN
      Undershoot: NaN
            Peak: Inf
        PeakTime: Inf
```




```
k = 2
z = 13
```




```
ans = struct with fields:
        RiseTime: NaN
   TransientTime: NaN
    SettlingTime: NaN
     SettlingMin: NaN
     SettlingMax: NaN
       Overshoot: NaN
      Undershoot: NaN
            Peak: Inf
        PeakTime: Inf
```



6. (b) The objective of this exercise is to analyse the system in frequencydomain. Obtain the Bode and Nyquist plot for the system you were given earlier. What are the gain and phase margins? What can you conclude about the stability of your system? The objective of this exercise is to obtain the closed loop behaviour with Lead or Lag compensator of the system you were given earlier. (i) Place a function $(\tau s+1)$ $\alpha(\alpha\tau s+1)$ in the forward path, and close the loop with negative unity feedback for Lag compensator and $K(\alpha\tau s+1)$ $\alpha(\tau s+1)$ for Lead compensator. For fixed K, take different values for α and τ, For each sets of (α, τ)obtain the Bode and Nyquist plots of the modified transfer function. Determine the phase and gain margins and also determine the step response of the closed loop transfer function. In addition, obtain the root locus plot. Discuss the results, and draw conclusions. Compare the results with P, PD in the forward path

```
num = [0 0 0 36.6];
den = [1 9.2 15.4 0];
tfm = tf(num,den);
figure
bode(tfm)
figure
nyquist(tfm)


%Lag compensator


num = [0 0 0 36.6];
den = [1 9.2 15.4 0];
```

```
K=1
alpha=2, tau=1
n1 = conv([1 1],num);
d1 = conv([4 2],den);
g1 = tf(n1,d1);
figure
bode(g1)
figure
nyquist(g1)
figure
rlocus(g1)
[n11,d11] = negfeedback(n1,1,d1,1);
tf1 = tf(n11,d11);
figure
step(tf1)
stepinfo(tf1)


alpha=3, tau=2
n2 = conv([2 1],num);
d2 = conv([18 3],den);
g2 = tf(n2,d2);
figure
bode(g2)
figure
nyquist(g2)
figure
rlocus(g2)
[n22,d22] = negfeedback(n2,1,d2,1);
tf2 = tf(n22,d22);
figure
step(tf2)
stepinfo(tf2)


alpha=4, tau=7
n3 = conv([7 1],num);
d3 = conv([112 4],den);
g3 = tf(n3,d3);
figure
bode(g3)
figure
nyquist(g3)
figure
rlocus(g3)
[n33,d33] = negfeedback(n3,1,d3,1);
tf3 = tf(n33,d33);
figure
step(tf3)
stepinfo(tf3)
```

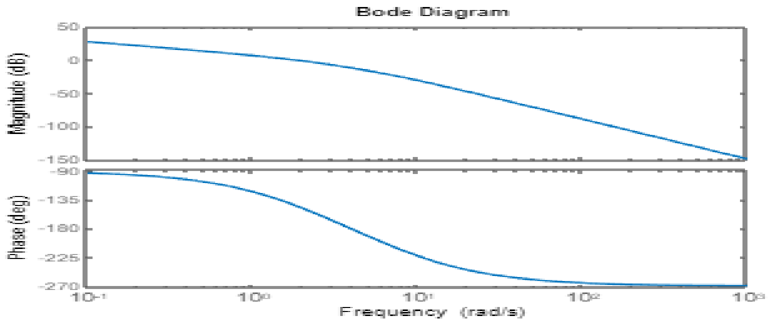
```
alpha=1, tau=7
n4 = conv([7 1],num);
d4 = conv([7 1],den);
g4 = tf(n4,d4);
figure
bode(g4)
figure
nyquist(g4)
figure
rlocus(g4)
[n44,d44] = negfeedback(n4,1,d4,1);
tf4 = tf(n44,d44);
figure
step(tf4)
stepinfo(tf4)


alpha=9, tau=5
n5 = conv([5 1],num);
d5 = conv([315 9],den);
g5 = tf(n5,d5);
figure
bode(g5)
figure
nyquist(g5)
figure
rlocus(g5)
[n55,d55] = negfeedback(n5,1,d5,1);
tf5 = tf(n55,d55);
figure
step(tf5)
stepinfo(tf5)


%Lead compensator


num = [0 0 0 36.6];
den = [1 9.2 15.4 0];


K=1
alpha=2, tau=1
n1 = conv([2 1],num);
d1 = conv([2 2],den);
g1 = tf(n1,d1);
figure
bode(g1)
figure
nyquist(g1)
```

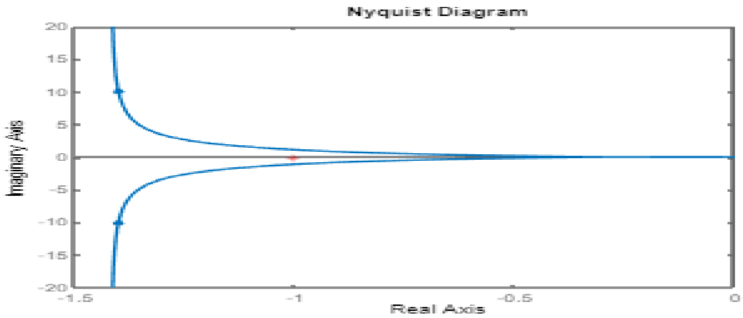
```
figure
rlocus(g1)
[n11,d11] = negfeedback(n1,1,d1,1);
tf1 = tf(n11,d11);
figure
step(tf1)
stepinfo(tf1)


alpha=3, tau=2
n2 = conv([6 1],num);
d2 = conv([6 3],den);
g2 = tf(n2,d2);
figure
bode(g2)
figure
nyquist(g2)
figure
rlocus(g2)
[n22,d22] = negfeedback(n2,1,d2,1);
tf2 = tf(n22,d22);
figure
step(tf2)
stepinfo(tf3)


alpha=4, tau=7
n3 = conv([28 1],num);
d3 = conv([28 4],den);
g3 = tf(n3,d3);
figure
bode(g3)
figure
nyquist(g3)
figure
rlocus(g3)
[n33,d33] = negfeedback(n3,1,d3,1);
tf3 = tf(n33,d33);
figure
step(tf3)
stepinfo(tf3)


alpha=1, tau=7
n4 = conv([7 1],num);
d4 = conv([7 1],den);
g4 = tf(n4,d4);
figure
bode(g4)
figure
nyquist(g4)
figure
```

```
rlocus(g4)
[n44,d44] = negfeedback(n4,1,d4,1);
tf4 = tf(n44,d44);
figure
step(tf4)
stepinfo(tf4)


alpha=9, tau=5
n5 = conv([35 1],num);
d5 = conv([35 9],den);
g5 = tf(n5,d5);
figure
bode(g5)
figure
nyquist(g5)
figure
rlocus(g5)
[n55,d55] = negfeedback(n5,1,d5,1);
tf5 = tf(n55,d55);
figure
step(tf5)
stepinfo(tf5)
```

```
function [num,den] = negfeedback(n1,n2,d1,d2)
num = conv(n1,d2);
den = conv(d1,d2)+conv(n1,n2);
end
```

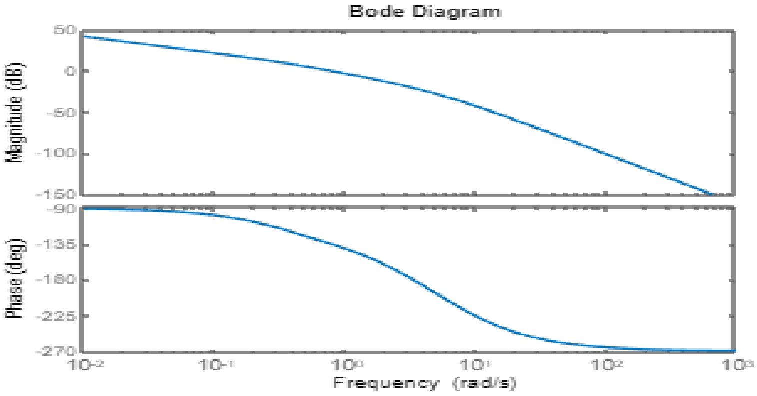
MATLAB OUTPUT:-

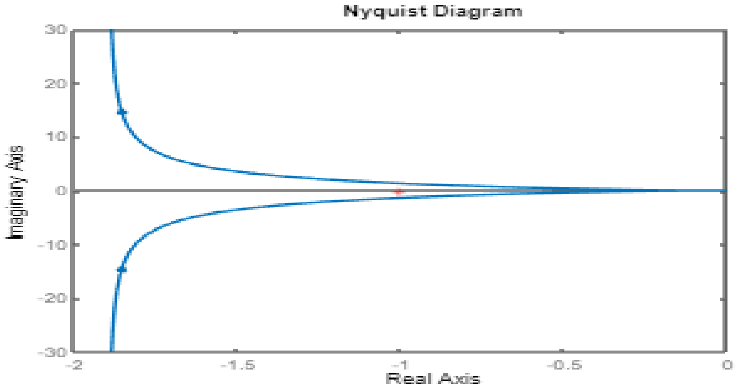


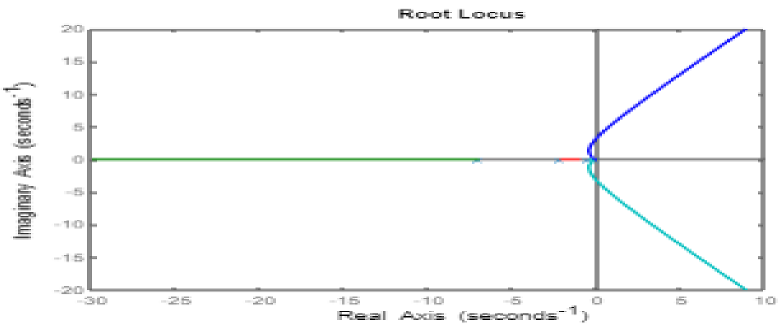


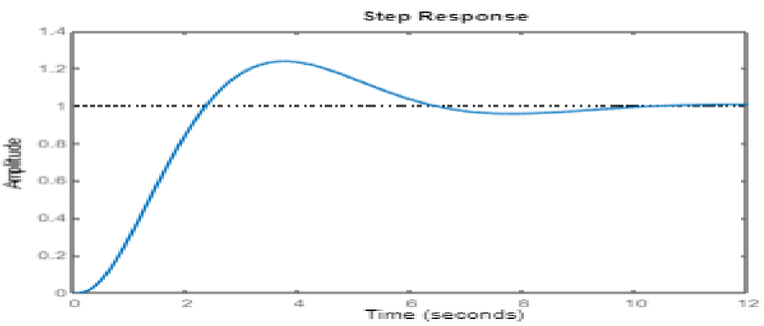
LAG COMPENSATOR:-

```
K = 1
alpha = 2
tau = 1
```



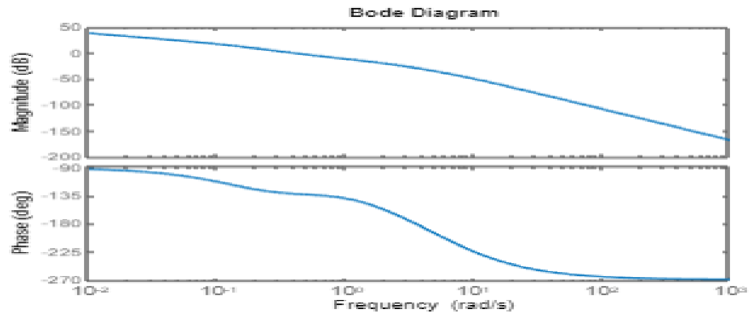


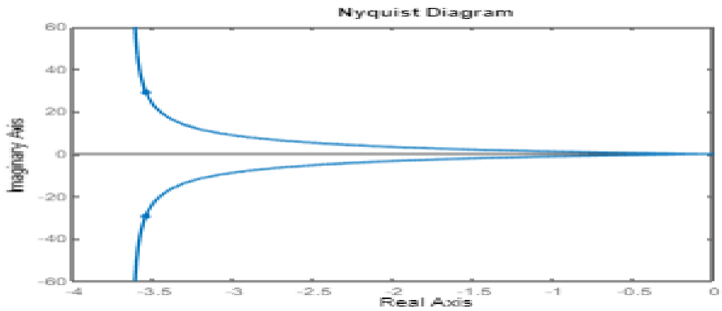


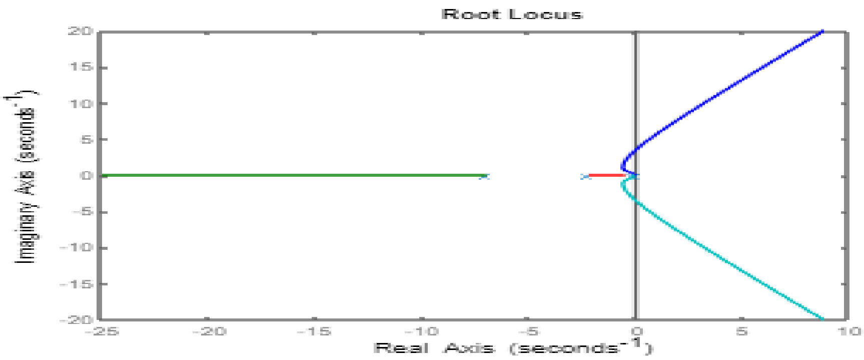


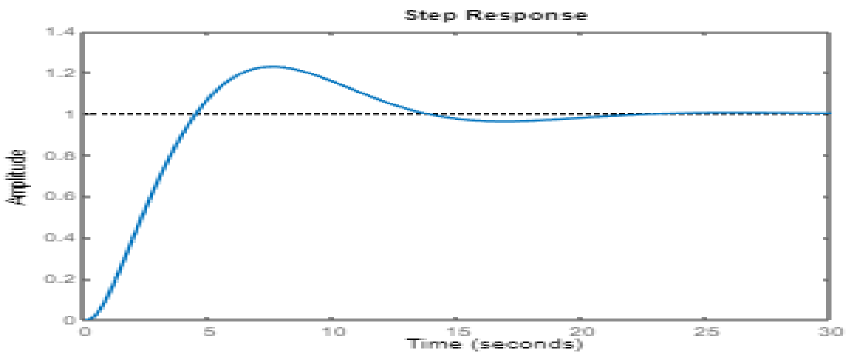

```
ans = struct with fields:
         RiseTime: 1.5294
    TransientTime: 9.3005
     SettlingTime: 9.3005
      SettlingMin: 0.9049
      SettlingMax: 1.2381
        Overshoot: 23.8112
       Undershoot: 0
             Peak: 1.2381
         PeakTime: 3.7547
```

```
alpha = 3
tau = 2
```



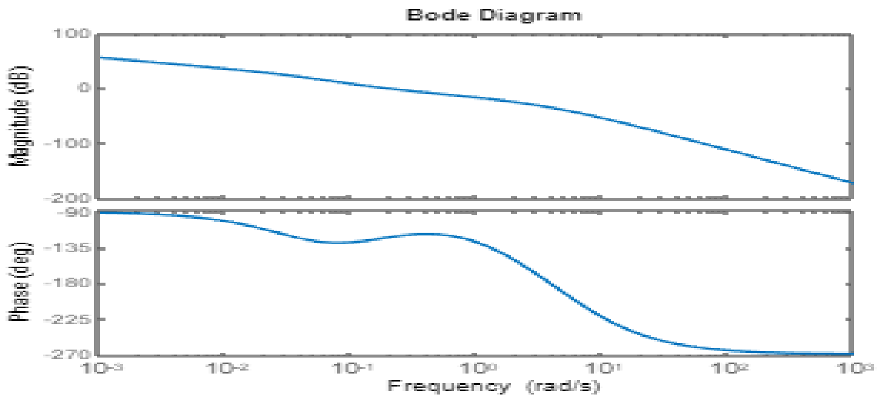


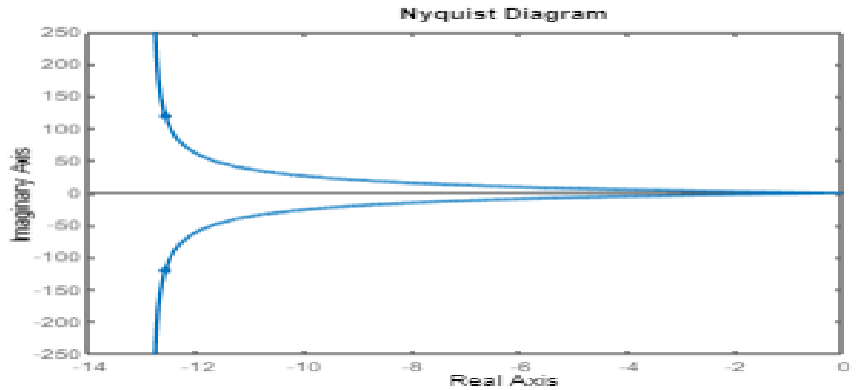


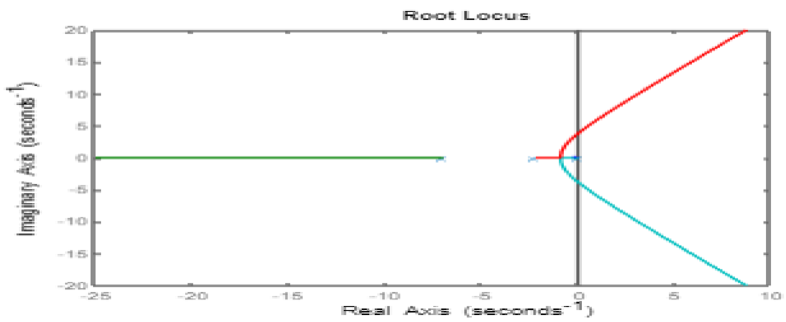


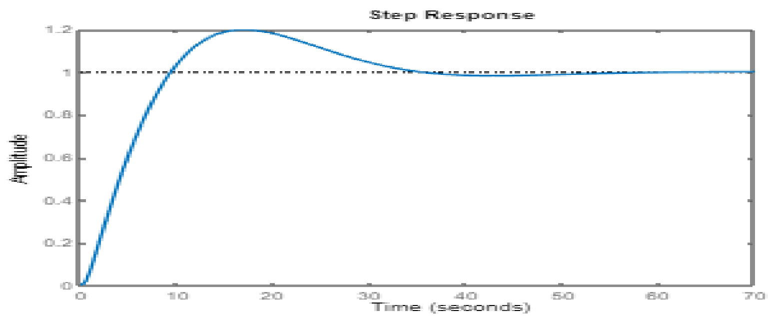

```
ans = struct with fields:
        RiseTime: 3.0675
   TransientTime: 19.8752
    SettlingTime: 19.8752
     SettlingMin: 0.9802
     SettlingMax: 1.2283
       Overshoot: 22.8343
      Undershoot: 0
            Peak: 1.2283
        PeakTime: 7.5338
```

```
        PeakTime: 7.5338
alpha = 4
tau = 7
```



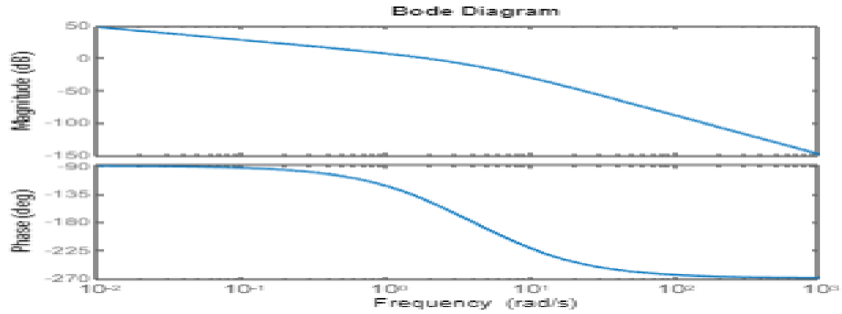


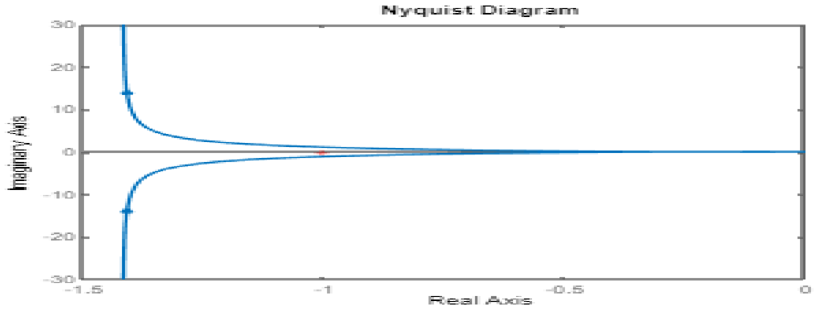


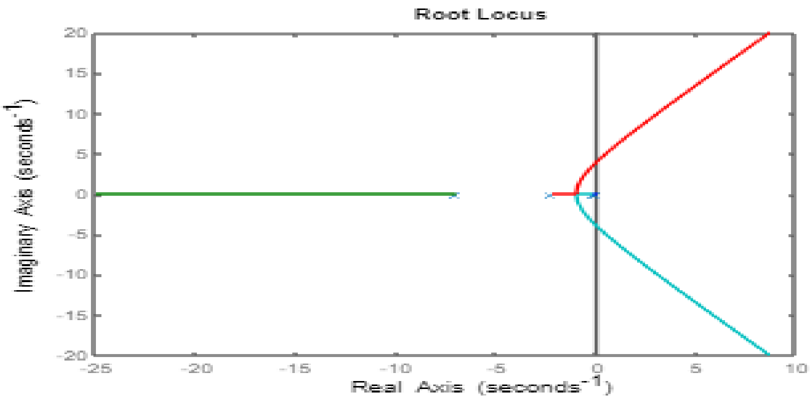


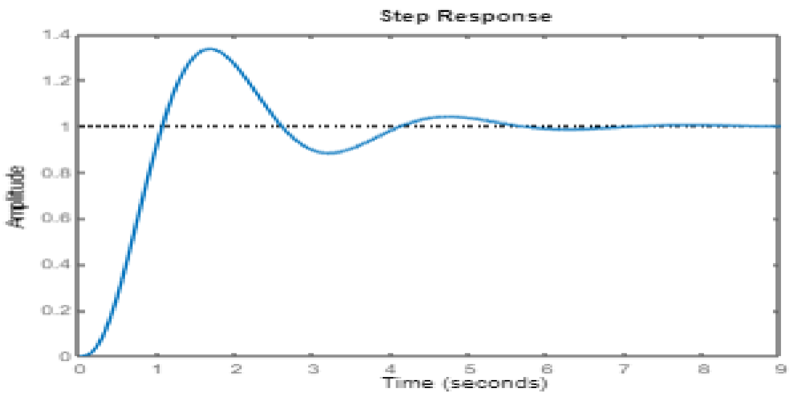

```
ans = struct with fields:
          RiseTime: 6.7323
     TransientTime: 32.5073
      SettlingTime: 32.5073
       SettlingMin: 0.9320
       SettlingMax: 1.1982
         Overshoot: 19.8185
        Undershoot: 0
              Peak: 1.1982
          PeakTime: 16.7029
```

```
alpha = 1
tau = 7
```



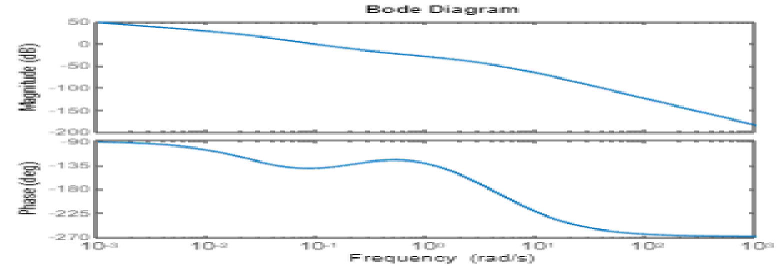


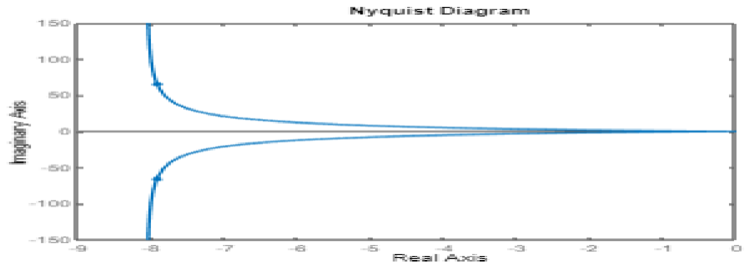


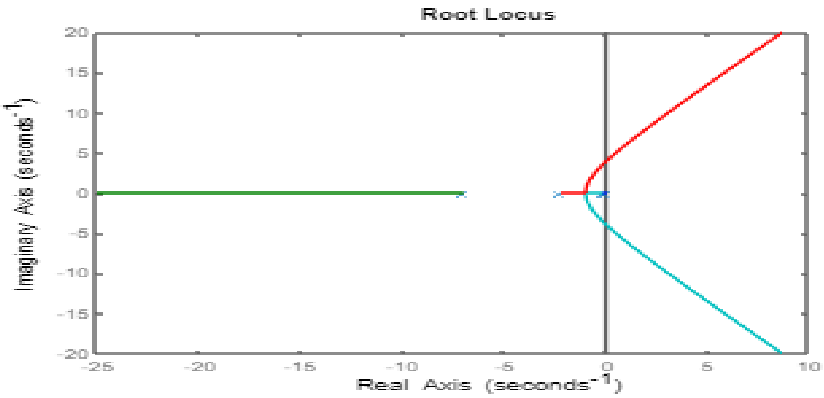


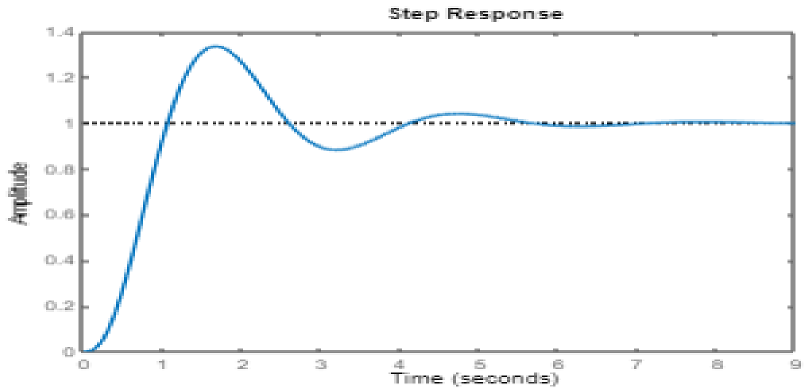

```
ans = struct with fields:
        RiseTime: 0.6592
   TransientTime: 5.2933
    SettlingTime: 5.2933
     SettlingMin: 0.8825
     SettlingMax: 1.3359
       Overshoot: 33.5871
      Undershoot: 0
            Peak: 1.3359
        PeakTime: 1.6810
```

```
alpha = 9
tau = 5
```





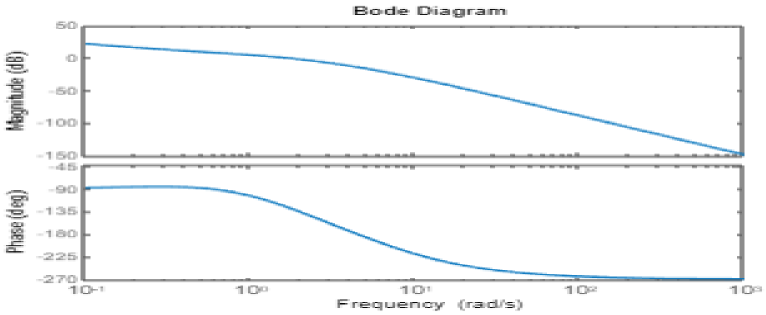


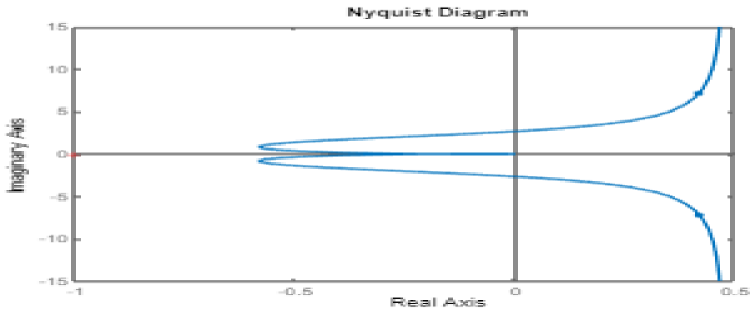


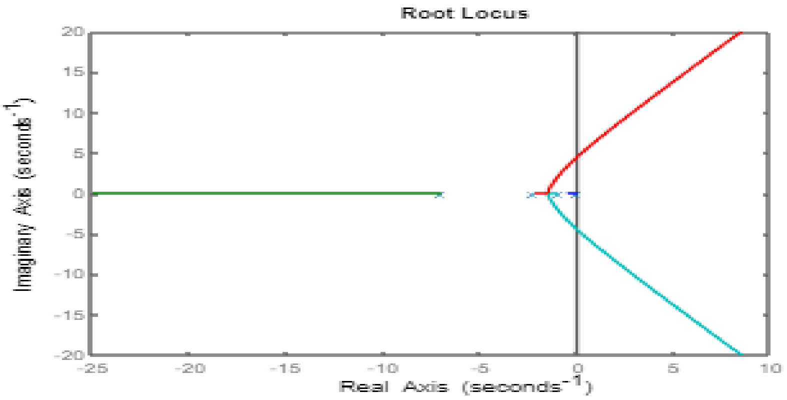

```
ans = struct with fields:
         RiseTime: 0.6592
    TransientTime: 5.2933
     SettlingTime: 5.2933
      SettlingMin: 0.8825
      SettlingMax: 1.3359
        Overshoot: 33.5871
       Undershoot: 0
             Peak: 1.3359
         PeakTime: 1.6810
```

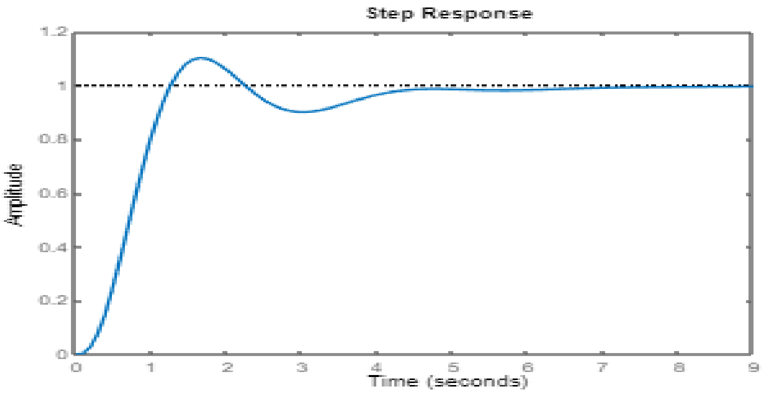
LEAD COMPENSATOR

```
K = 1
alpha = 2
tau = 1
```

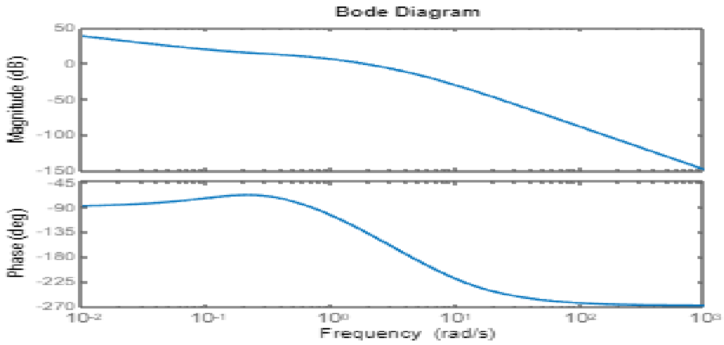


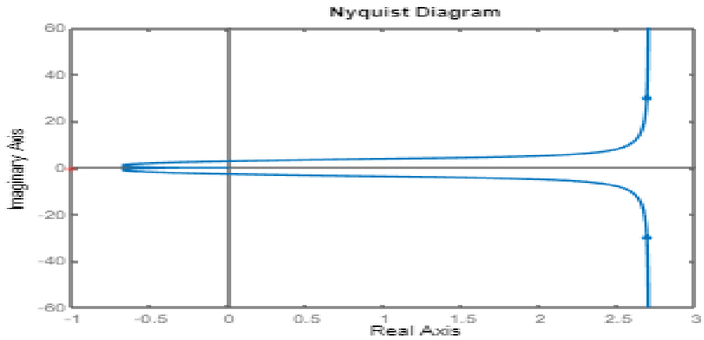


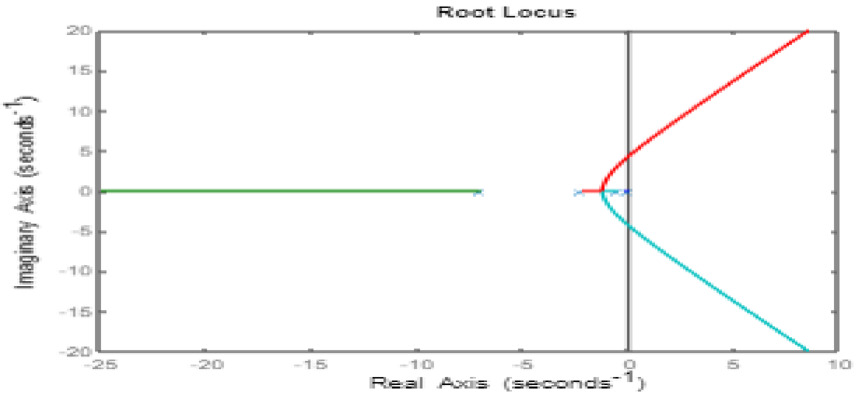


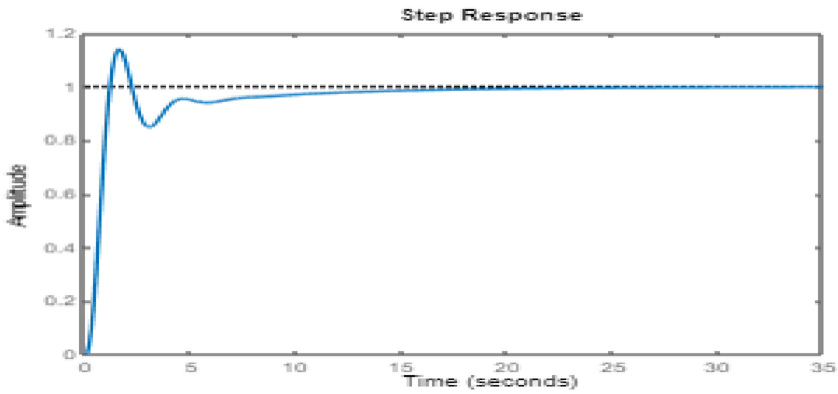


```
alpha = 3
tau = 2
```



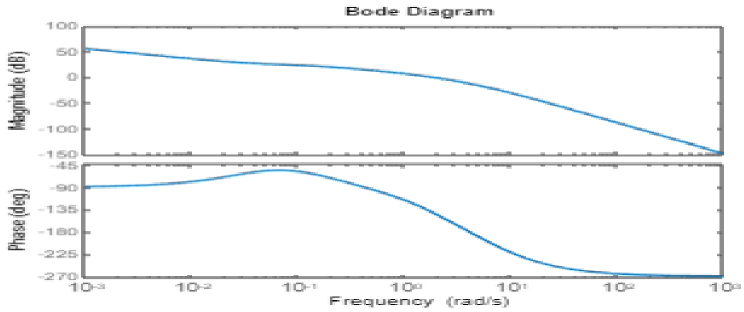


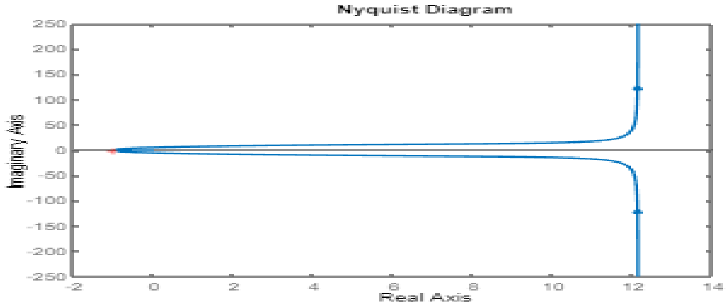


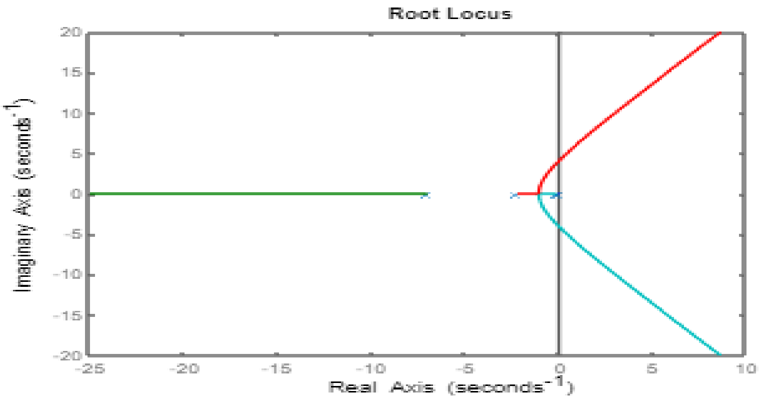


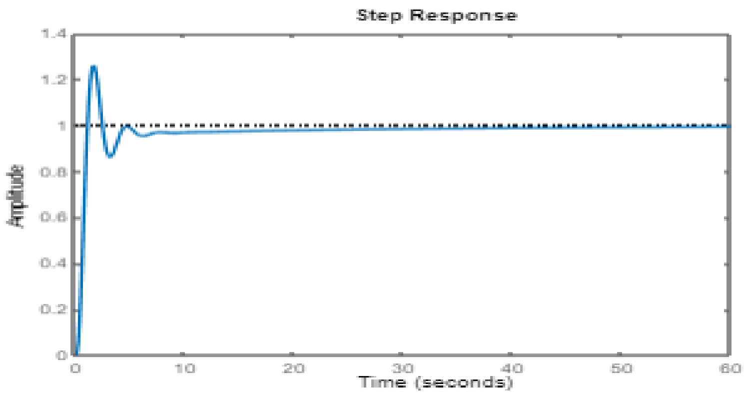

```
ans = struct with fields:
         RiseTime: 6.7323
    TransientTime: 32.5073
     SettlingTime: 32.5073
      SettlingMin: 0.9320
      SettlingMax: 1.1982
        Overshoot: 19.8185
       Undershoot: 0
             Peak: 1.1982
         PeakTime: 16.7029
```

```
alpha = 4
tau = 7
```



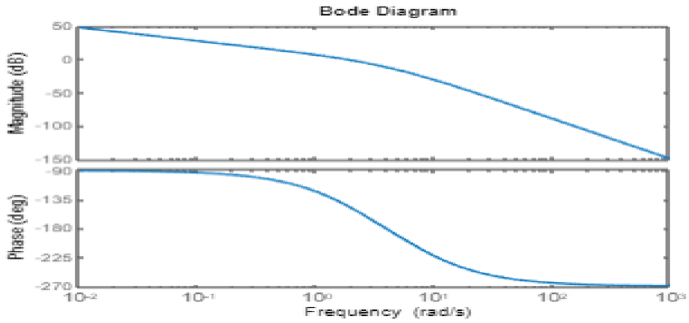


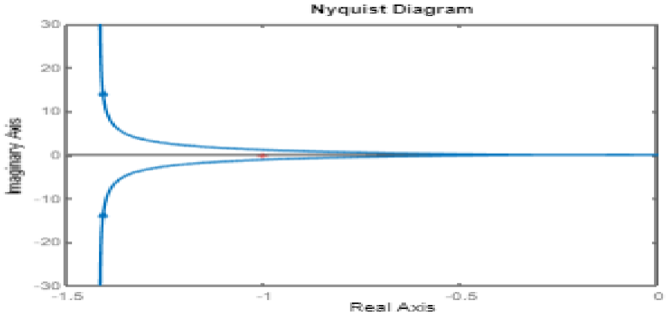


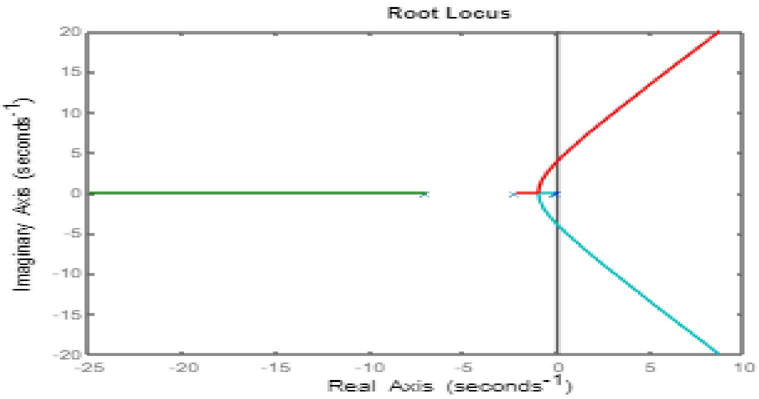


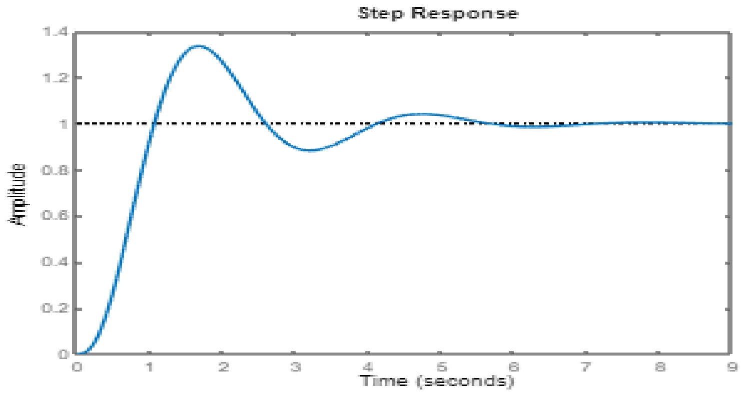

```
ans = struct with fields:
        RiseTime: 0.6826
   TransientTime: 22.7317
    SettlingTime: 22.7317
     SettlingMin: 0.8611
     SettlingMax: 1.2611
       Overshoot: 26.1147
      Undershoot: 0
            Peak: 1.2611
        PeakTime: 1.6298
```

```
alpha = 1
tau = 7
```










```
ans = struct with fields:
        RiseTime: 0.6592
   TransientTime: 5.2933
    SettlingTime: 5.2933
     SettlingMin: 0.8825
     SettlingMax: 1.3359
       Overshoot: 33.5871
      Undershoot: 0
            Peak: 1.3359
        PeakTime: 1.6810
```