Control Systems
UE20EC251

<u>PROJECT</u>

- Shalett Paulson PES2UG20EC078
- Sowmya Ravi PES2UG20EC091
- Suhas KV PES2UG20EC093
- Yagna Vivek PES2UG20EC109

1. A control system has the following transfer function $H(s) = \frac{1}{s(s+1)(s+2)}$

Compute the system response to the following inputs using MATLAB.

(i) $x(t) = u(t)$

(ii) $x(t) = tu(t)$

(iii) $x(t) = 2(sin2t)u(t)$

(iv) $x(t) = 2(sin10t)u(t)$

Also find the steady state error in case of step and ramp response ((i) and (ii)).

Comment on the responses in (iii) and (iv).

```
clc;
close all;
clear all;
%1.
%i
s1=tf([1],[1 3 2 0]);
title("Step response");
step(s1)

%ii
augs=series(tf([1],[1 0]),s);
step(augs)
title("Ramp response");

%iii
t1=0:0.1:10;
u=2*(sin(2*t1));
a=lsim(s,u,t1);
plot(t1,a);

```

```


%iv
t2=0:0.1:10;
u=2*(sin(10*t2));
b=lsim(s,u,t2);
plot(t2,b);

%steady state error

%unit step input
kp=dcgain(s1);
ess1=1/(1+kp);

%unit ramp input
augs2=series(tf([1],[1 0]),s1);
kv=dcgain(augs2);
ess2=1/kv;
```

i)

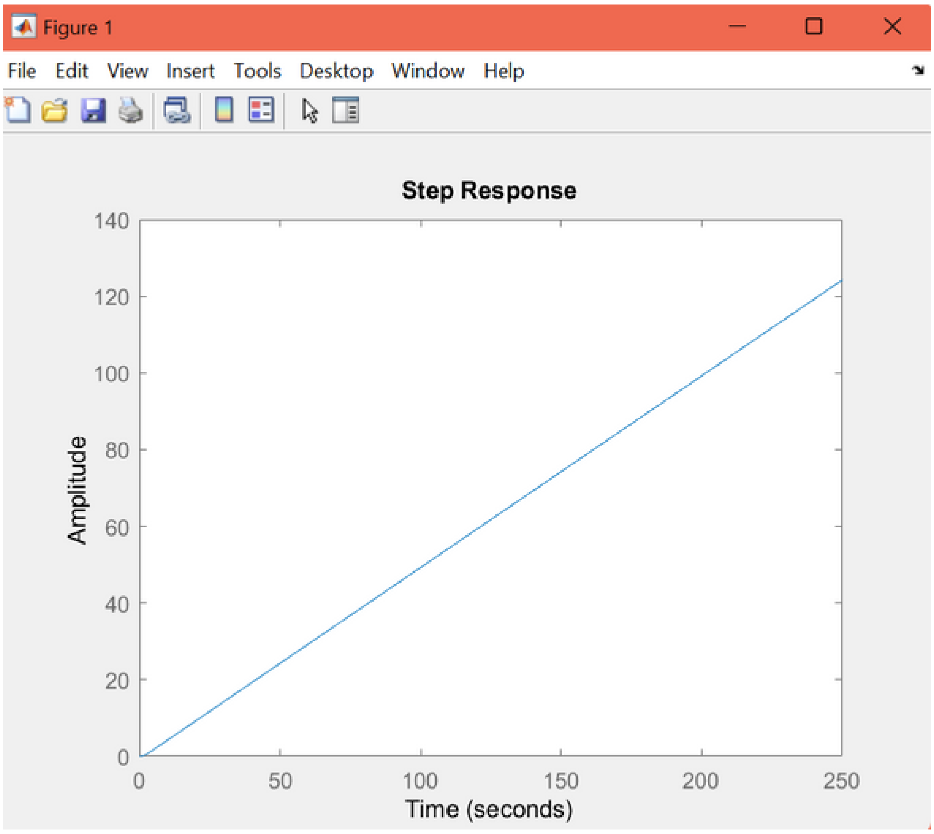


Figure 1
File Edit View Insert Tools Desktop Window Help
Step Response
Amplitude
140
120
100
80
60
40
20
0
0
50
100
150
200
250
Time (seconds)

ii)

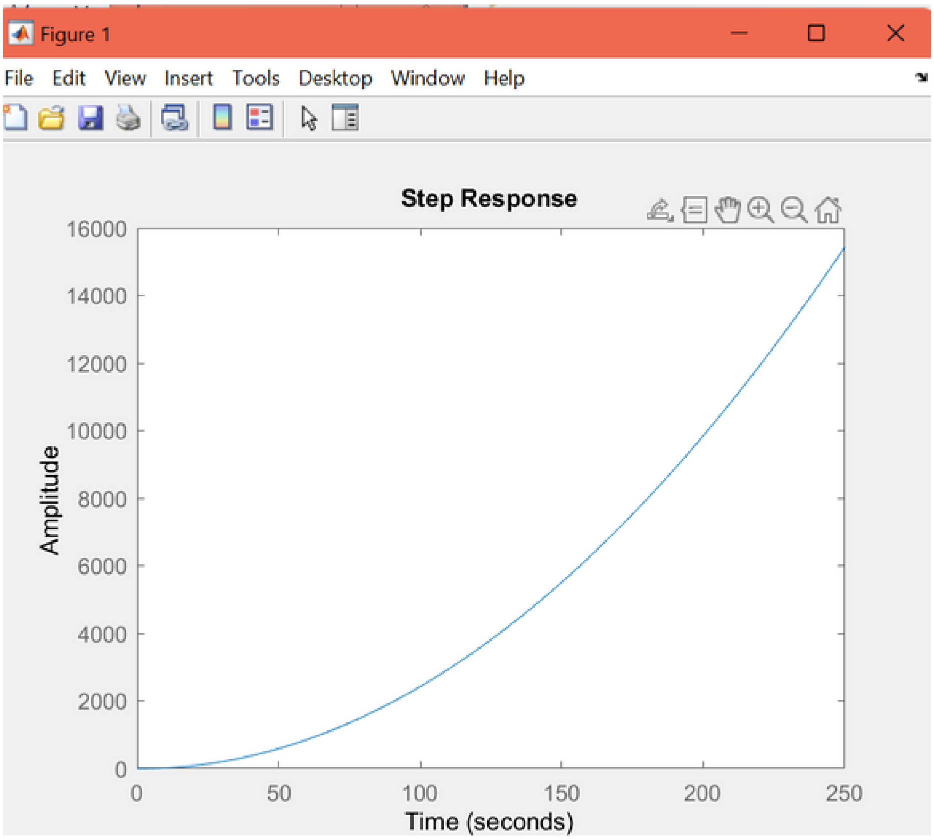


Figure 1
File Edit View Insert Tools Desktop Window Help
Step Response
Amplitude
16000
14000
12000
10000
8000
6000
4000
2000
0
0
50
100
150
200
250
Time (seconds)

iii)

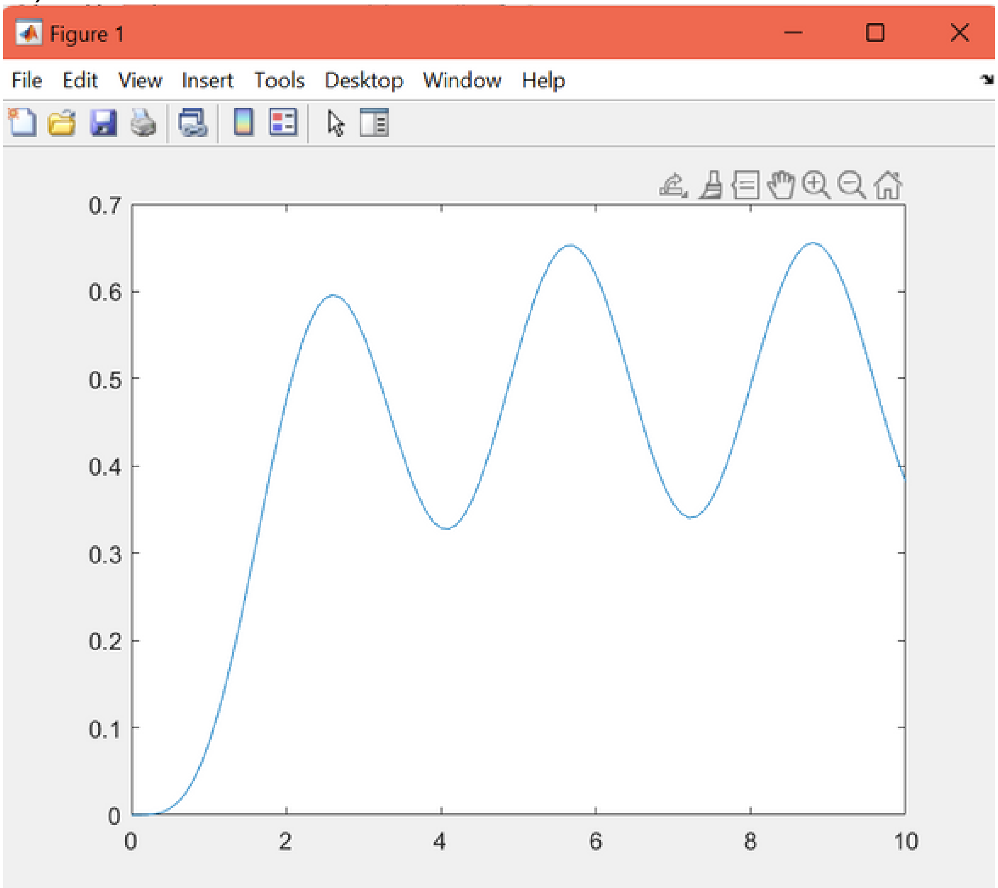


Figure 1
File Edit View Insert Tools Desktop Window Help
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0 2 4 6 8 10

iv)

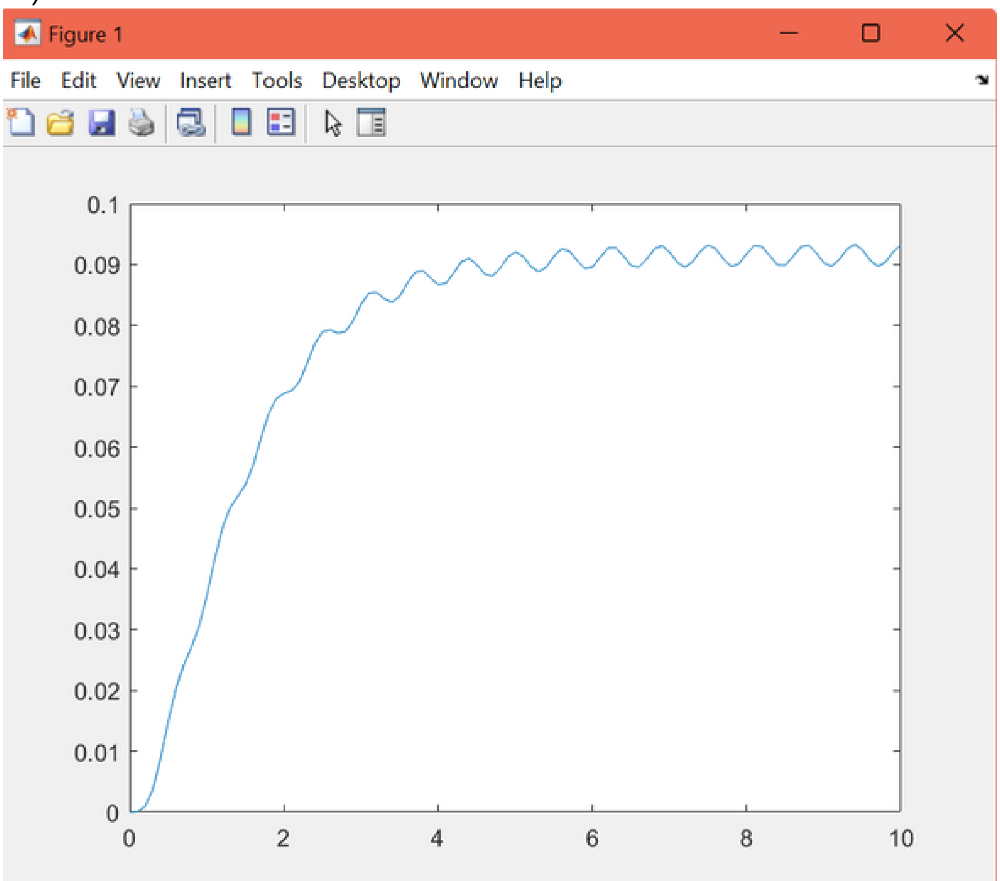


Figure 1
File Edit View Insert Tools Desktop Window Help
0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0 2 4 6 8 10

Steady state error calculation

Command Window

```
kp =

   Inf


ess1 =

     0

fx >>
```

Command Window

```
augs2 =

          1
  -------------------
  s^4 + 3 s^3 + 2 s^2

Continuous-time transfer function.


kv =

   Inf


ess2 =

     0

fx >>
```

3. Consider the third order system with the following transfer function.

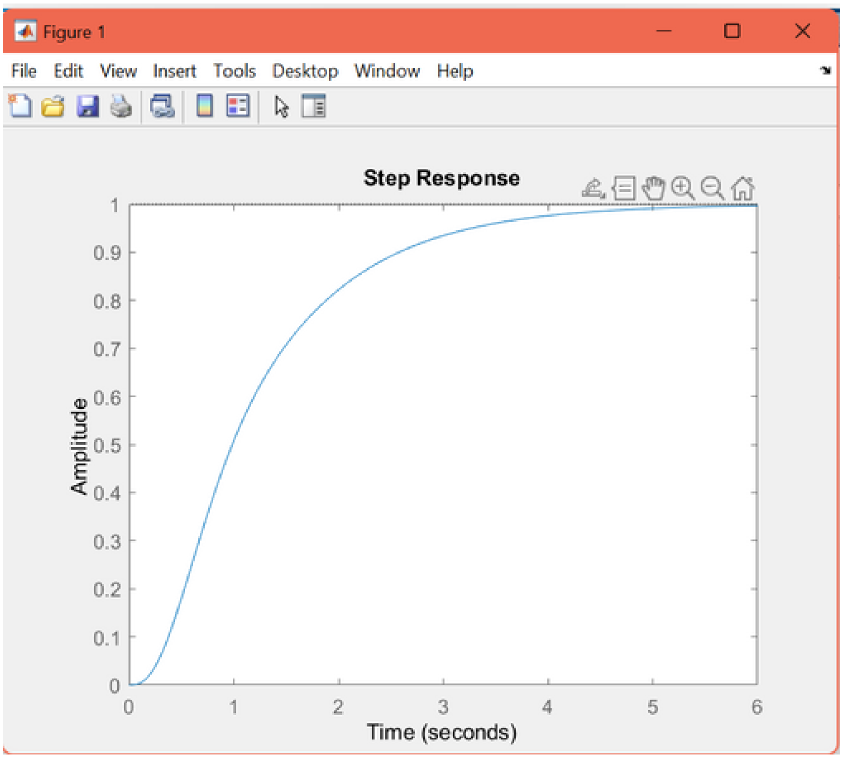
$$H(s) = \frac{25}{(s^2 + 7s + 25)(s + 1)}$$

a. Obtain the step response.

b. Now consider $H(s) = \frac{25}{s+1}$ and obtain the step response. Compare both the responses.

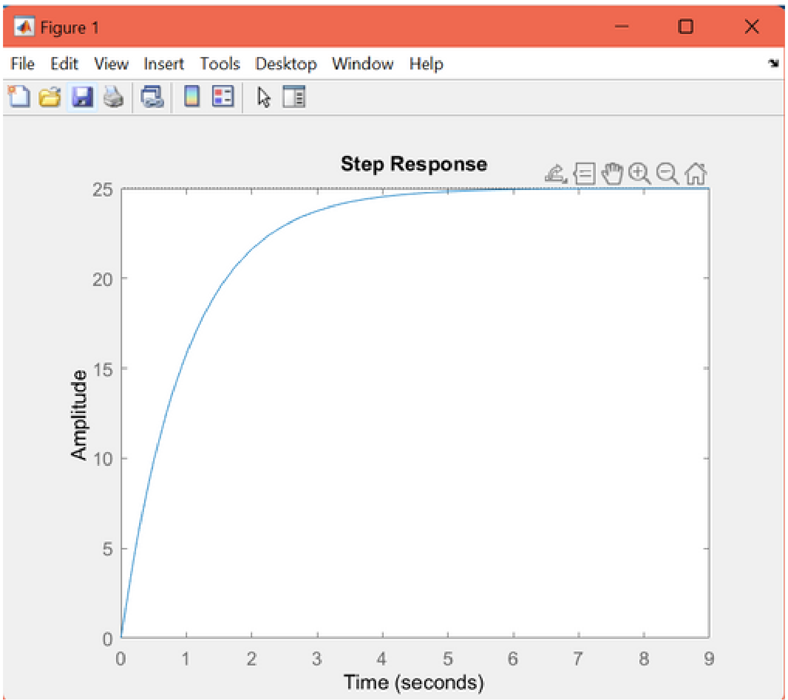
c. Now add a zero at s= -0.9 to the existing transfer function. Adjust the constant so that steady state value is still 1. Find the step response.
Comment on the observation w.r.t dominant poles (poles near jω axis).

```
close all
clear all
%a.
s1=tf([25],[1 8 32 25]);
step(s1);
kp1=dcgain(s1);
ess1=1/(1+kp1);
hold on

%b;
s2=tf([25],[1 1]);
step(s2);
kp2=dcgain(s2);
ess2=1/(1+kp2);

%c
k=input("enter a constant: ");%k=0 for steady state value 1
s3=tf([25 k],[1 1]);
step(s3);
kp3=dcgain(s3);
ess3=1/(1+kp3);
```



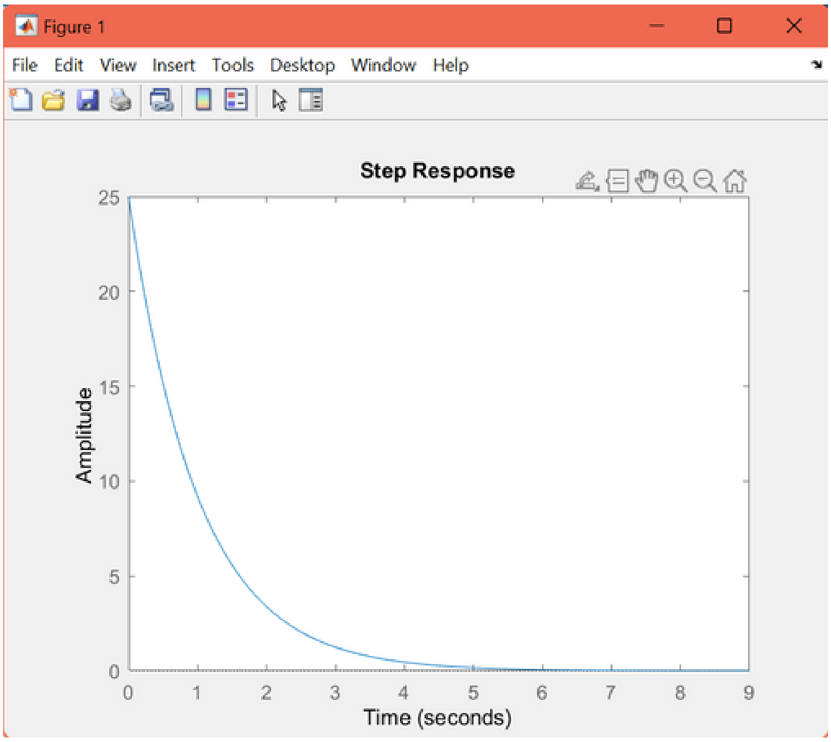
Command Window

```
kp1 =

     1


ess1 =

    0.5000
```

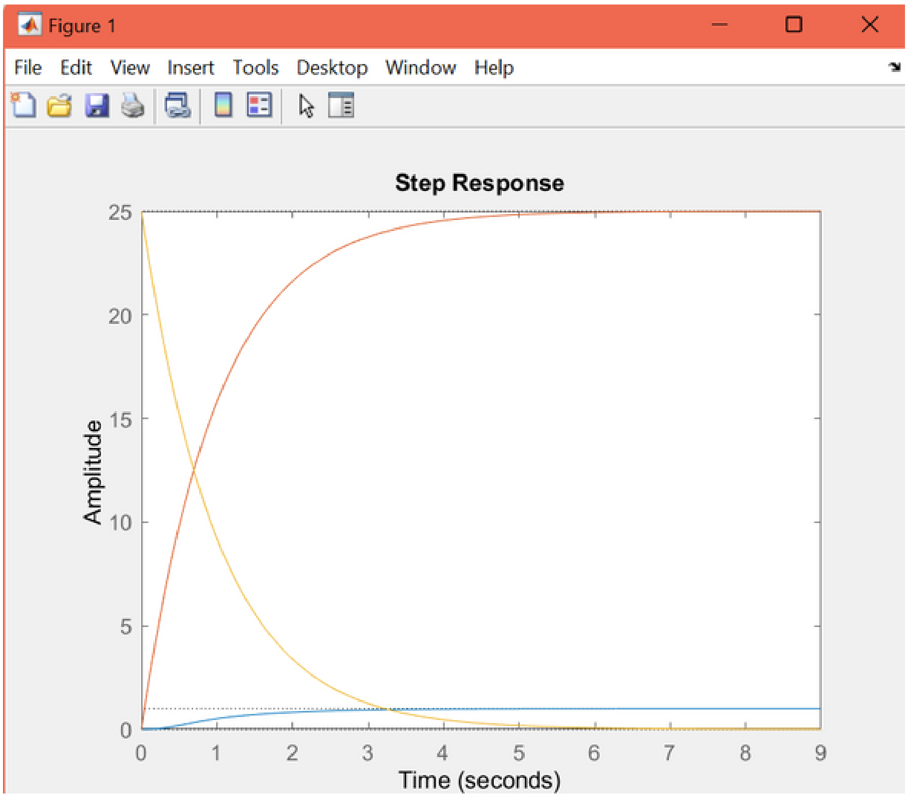


Command Window

```
kp2 =

    25


ess2 =

    0.0385
```




```
enter a constant: 0

kp3 =

     0


ess3 =

     1
```



Comparison

5. Consider a unity feedback system with $G(s) = \frac{K(s+30)}{(s+1)(s^2+20s+116)}$

a. Draw the root locus of the system manually and find the range of k for system to be stable. Verify the same using MATLAB.

b. Add a zero at s=-2 and again plot root locus.

c. Add a pole at s=-2 to G(s) and plot the root locus.

d. Add a zero at s=+2 to G(s) and plot the root locus.

e. Compare and comment on the stability of the system and response time using results in (b), (c) and (d).

For (b),(c) and (d) use Matlab.

```
%a
num1=[1 30];
den1=conv([1 1],[1 20 116])
sys1=tf(num1,den1);
rlocus(sys1)


%b
num2=conv([1 30],[1 2])
den2=[1 21 136 116];
sys2=tf(num2,den2);
rlocus(sys2)

%c
num3=[1 30];
d=conv([1 1],[1 20 116])
den3=conv(d,[1 2])
sys3=tf(num3,den3);
rlocus(sys3)

%d
num4=conv([1 30],[1 -2])
den4=[1 21 136 116]
sys4=tf(num4,den4);
rlocus(sys4)
```

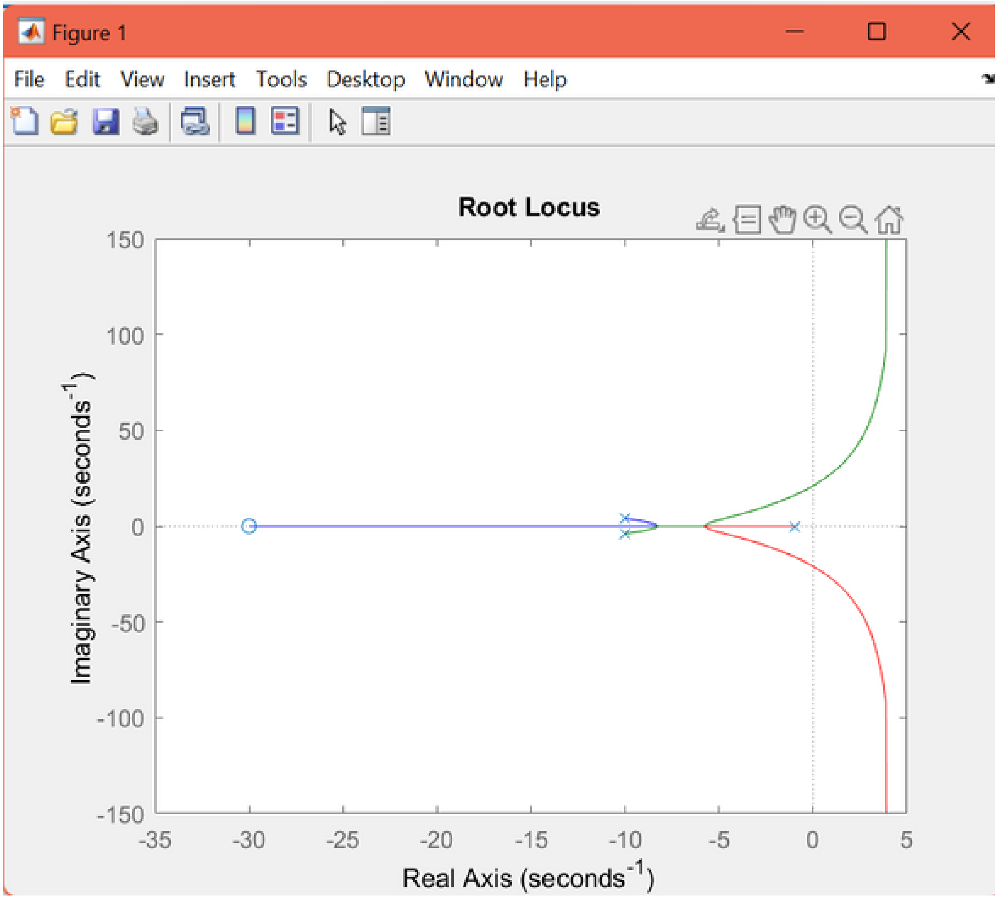


Figure 1
File Edit View Insert Tools Desktop Window Help
Root Locus
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})
150
100
50
0
-50
-100
-150
-35 -30 -25 -20 -15 -10 -5 0 5

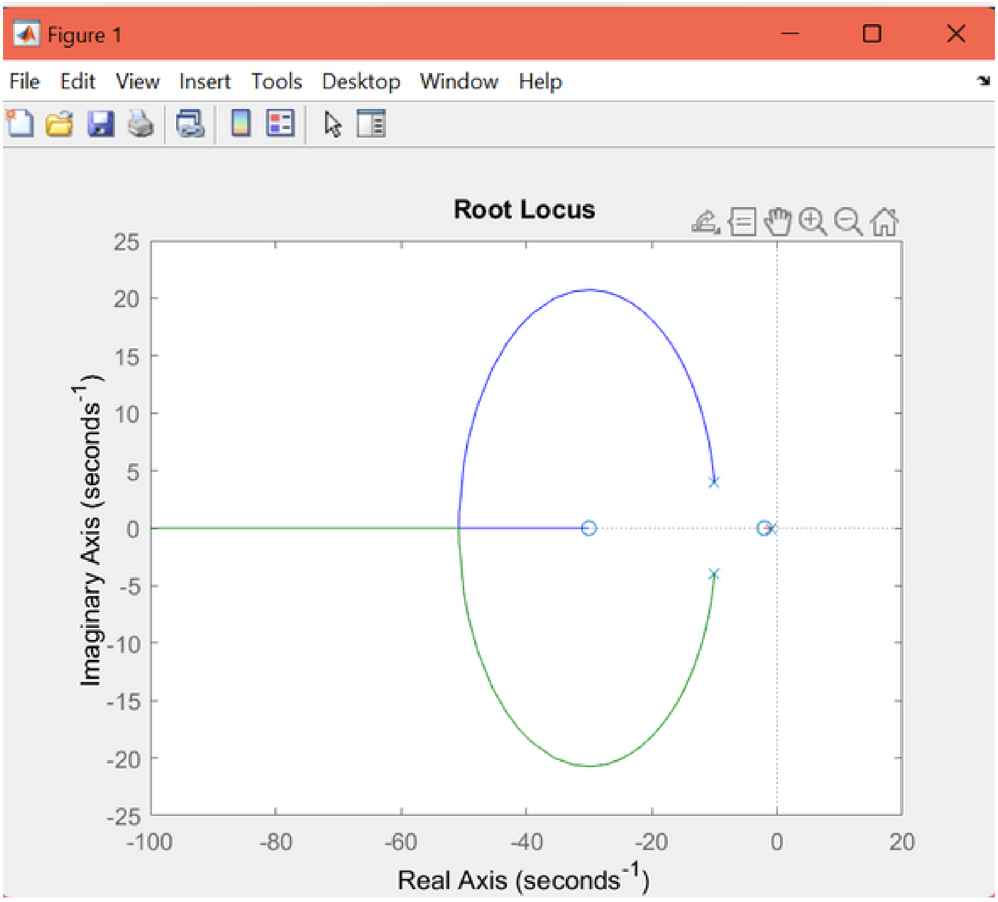


Figure 1
File Edit View Insert Tools Desktop Window Help
Root Locus
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})
25
20
15
10
5
0
-5
-10
-15
-20
-25
-100 -80 -60 -40 -20 0 20

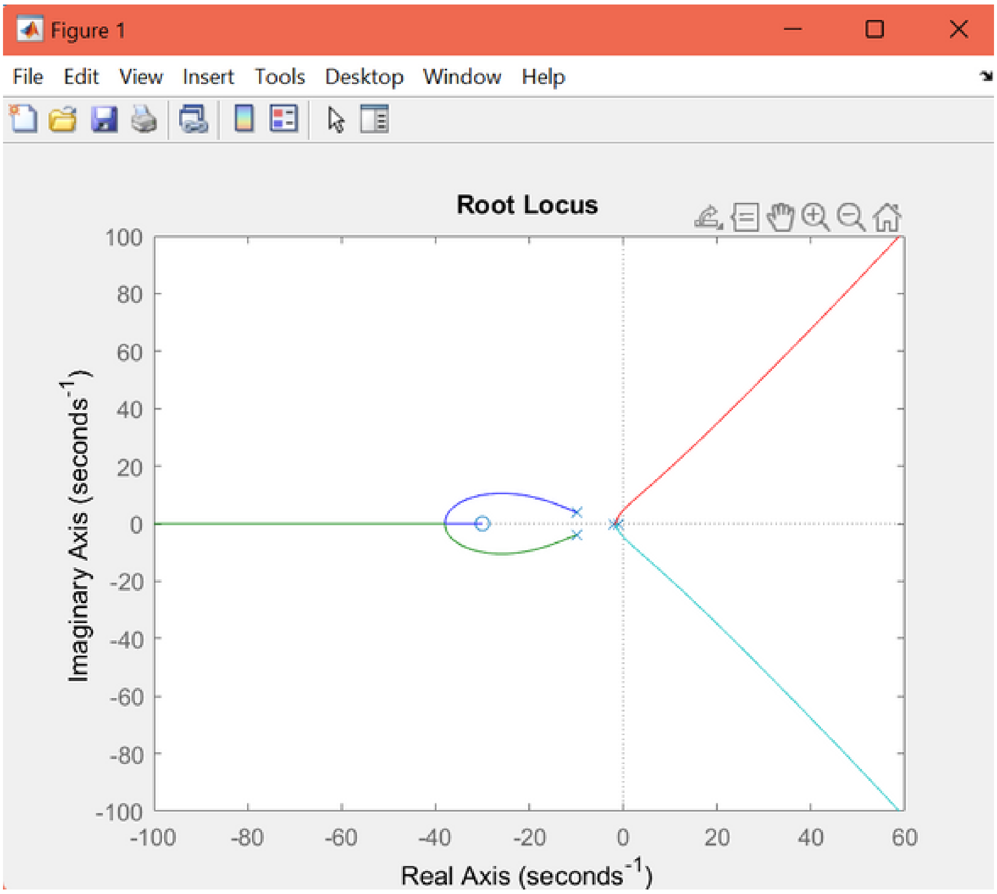


Figure 1
File Edit View Insert Tools Desktop Window Help
Root Locus
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})
100
80
60
40
20
0
-20
-40
-60
-80
-100
-100 -80 -60 -40 -20 0 20 40 60

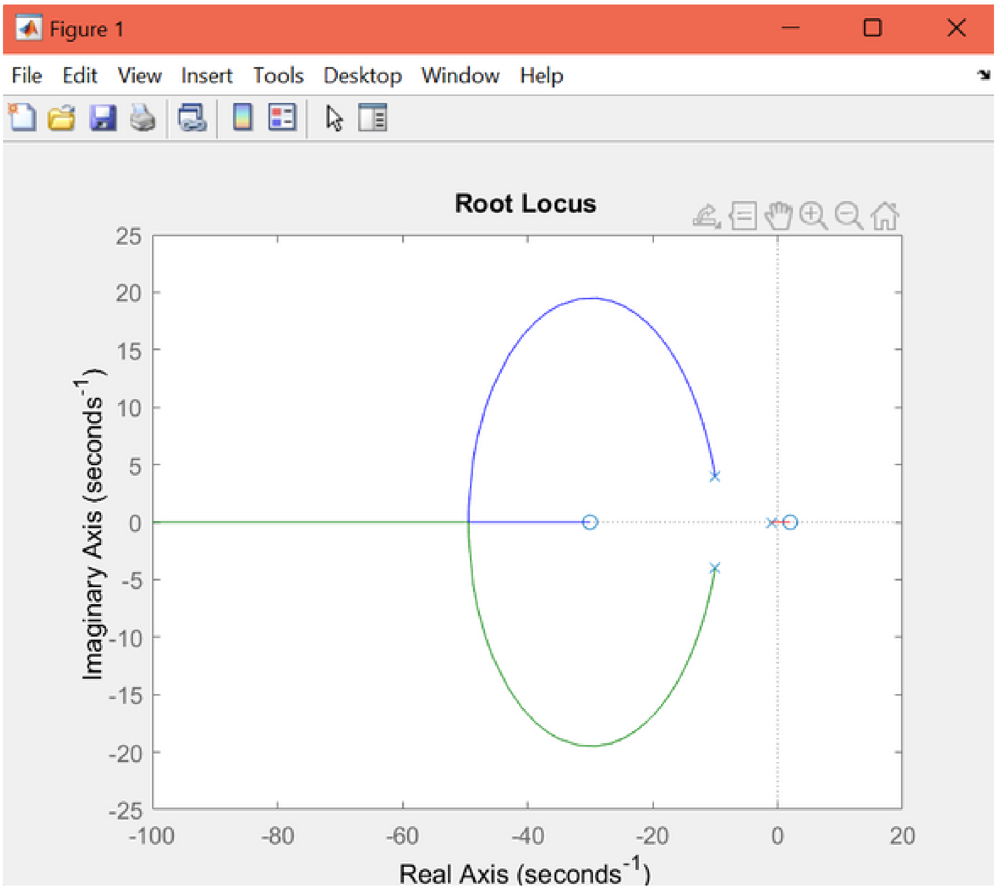


Figure 1
File Edit View Insert Tools Desktop Window Help
Root Locus
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})
25
20
15
10
5
0
-5
-10
-15
-20
-25
-100 -80 -60 -40 -20 0 20

7. Consider the plant G(s) and a proportional integrator (PI) compensator K(s) given as

$$G(s) = \frac{4}{s(s+2)} \; and \; K(s) = \frac{s+1}{s}$$

a. Show that if K(s) is in series with G(s) with unity feedback, the system will be stable and can track step and ramp inputs with zero error.

b. Place K(s) in feedback path and check its stability and steady state error to step and ramp inputs.

Comment on the results of (a) and (b).

a. Suppose during implementation, the following incorrect compensator is used

$K_2(s) = \frac{s+0.5}{s}$

Repeat part (a) with this compensator. Are the tracking properties lost?

```
%a
num=[4 4];
den=[1 2 0 0];
sy=tf(num,den);
sys=feedback(sy,1);
step(sys);
augs=series(tf([1],[1 0]),sys);
step(augs);
title("Ramp response");
%B=isstable(sys)%to check stability


%b
g=tf([4],[1 0 2]);
k=tf([1 1],[1 0]);
sys=feedback(g,k);
step(sys);
augs=series(tf([1],[1 0]),sys);
step(augs);
title("Ramp response");
%B=isstable(sys)%to check stability

%c
num=[4 2];
den=[1 2 0 0];
sy=tf(num,den);
sys=feedback(sy,1);
step(sys);
augs=series(tf([1],[1 0]),sys);
step(augs);
title("Ramp response");
B=isstable(sys)%to check stability
```

a)

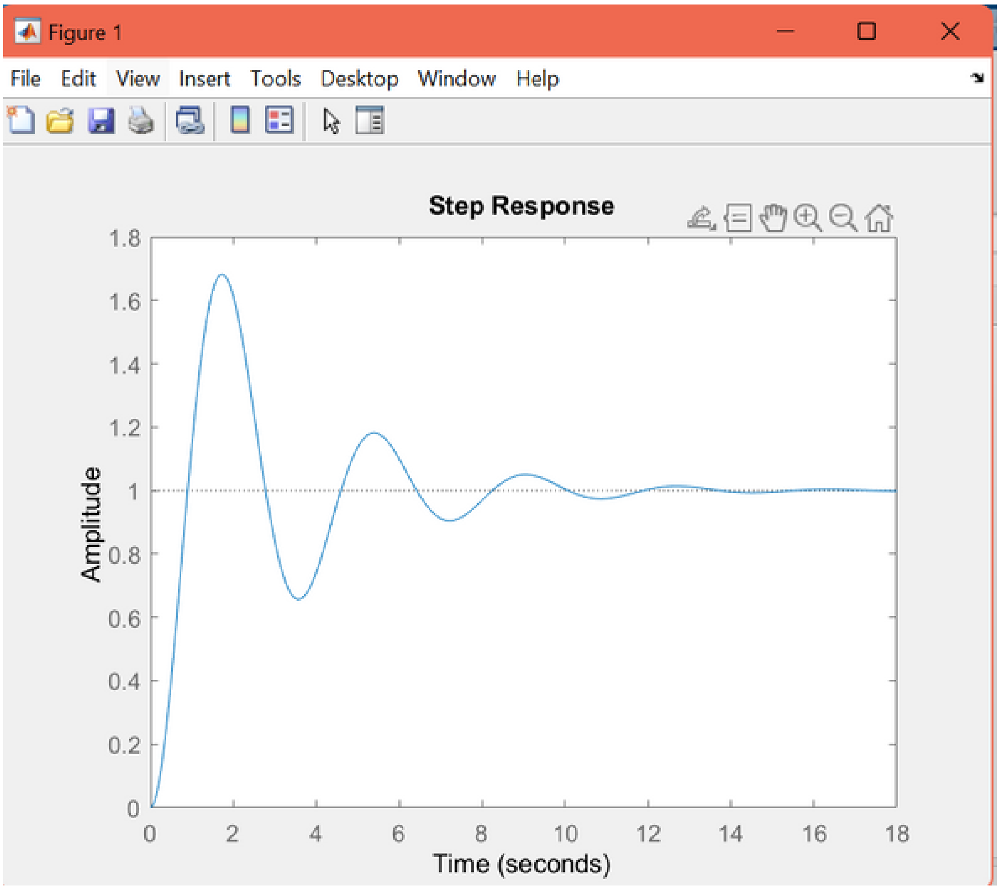


Figure 1
File Edit View Insert Tools Desktop Window Help
Step Response
Amplitude
Time (seconds)
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
0 2 4 6 8 10 12 14 16 18

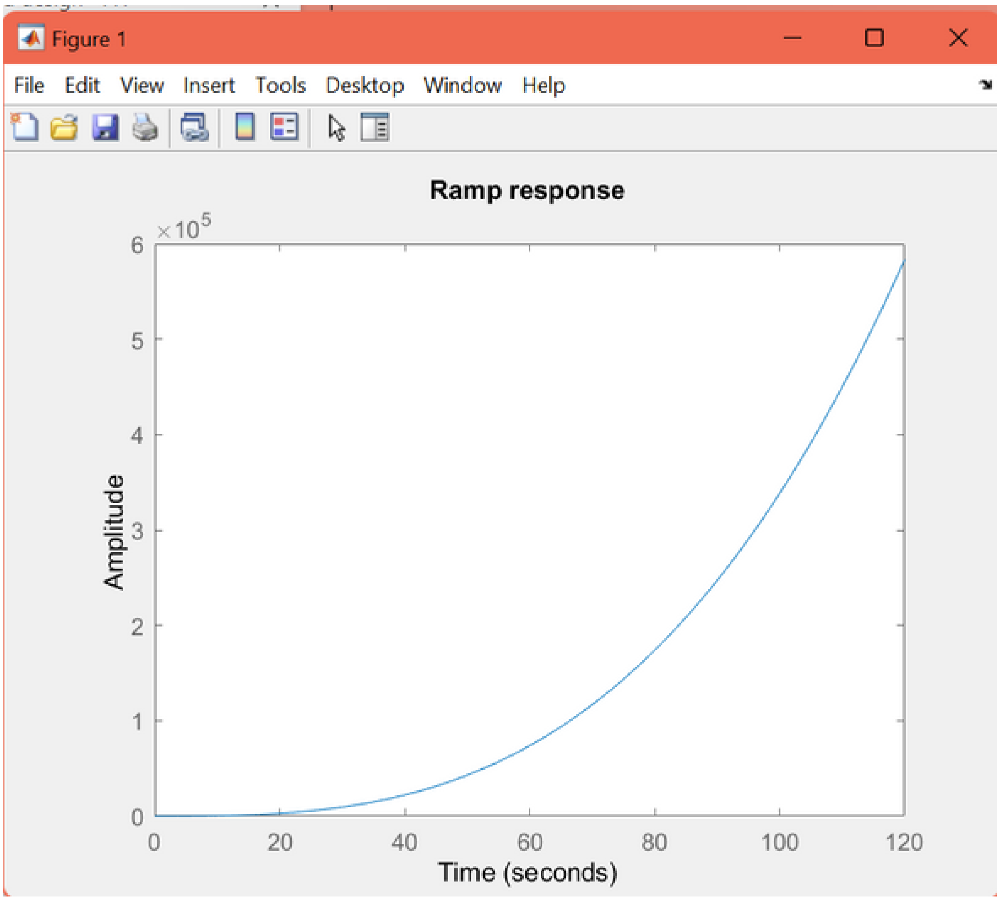
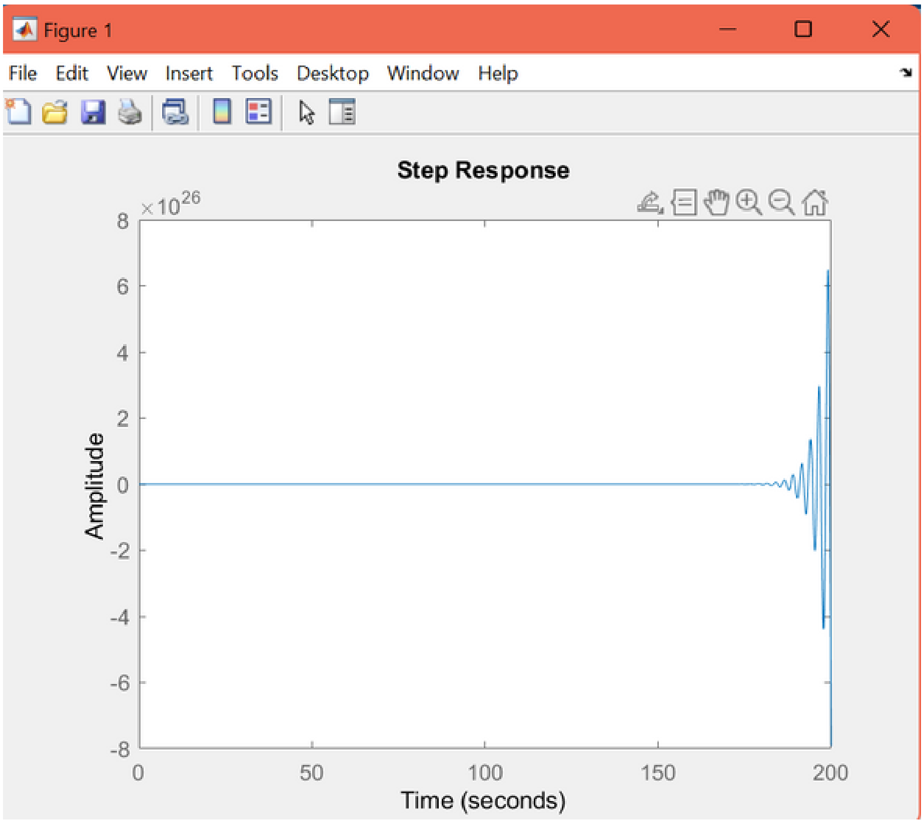
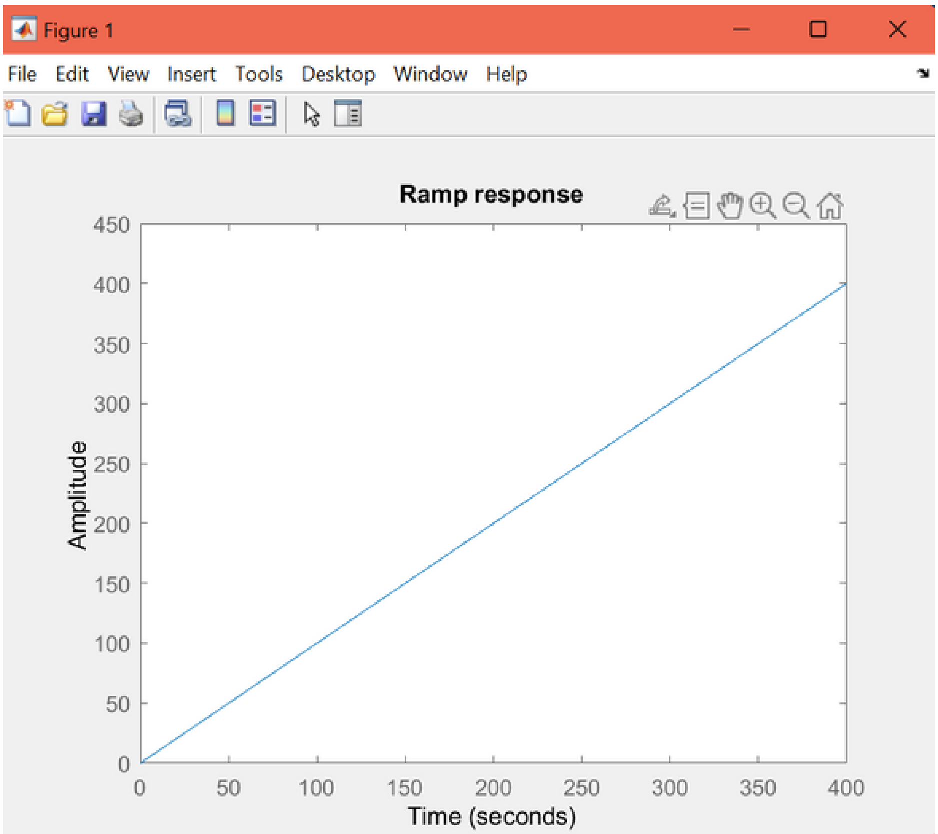


Figure 1
File Edit View Insert Tools Desktop Window Help
Ramp response
×10^5
Amplitude
Time (seconds)
0
1
2
3
4
5
6
0 20 40 60 80 100 120

```
>> q7m

B1 =

  logical

   1
```

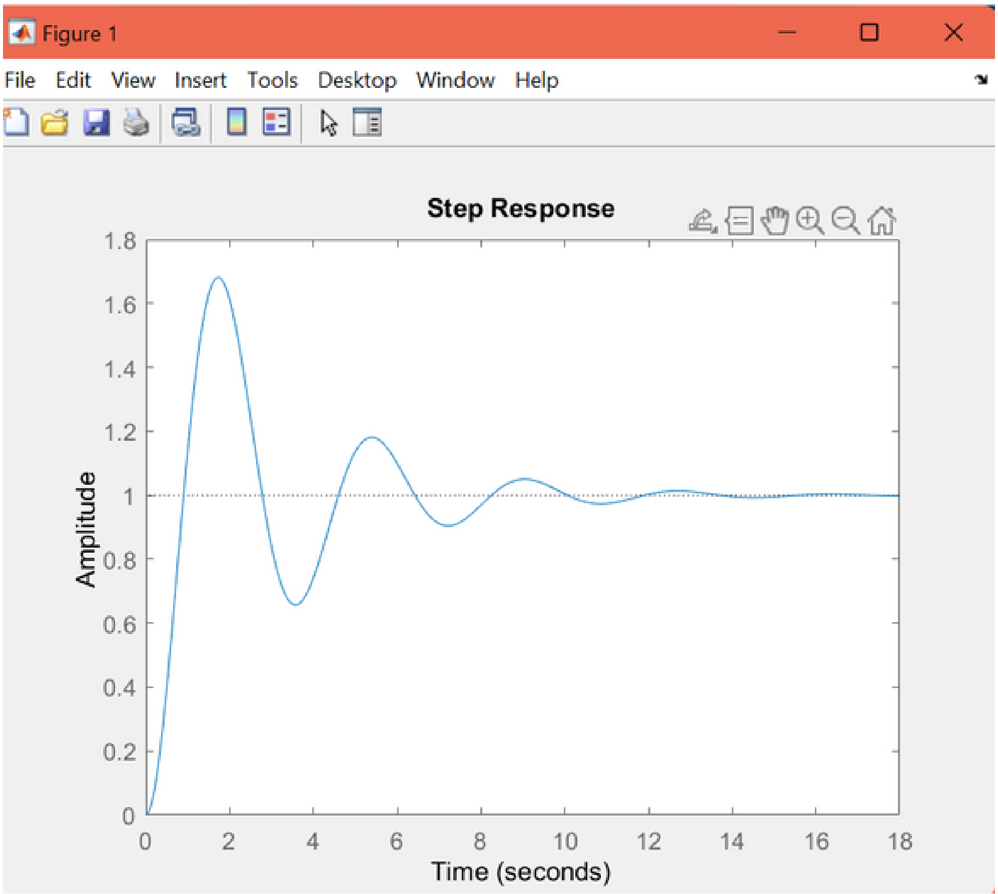
b)

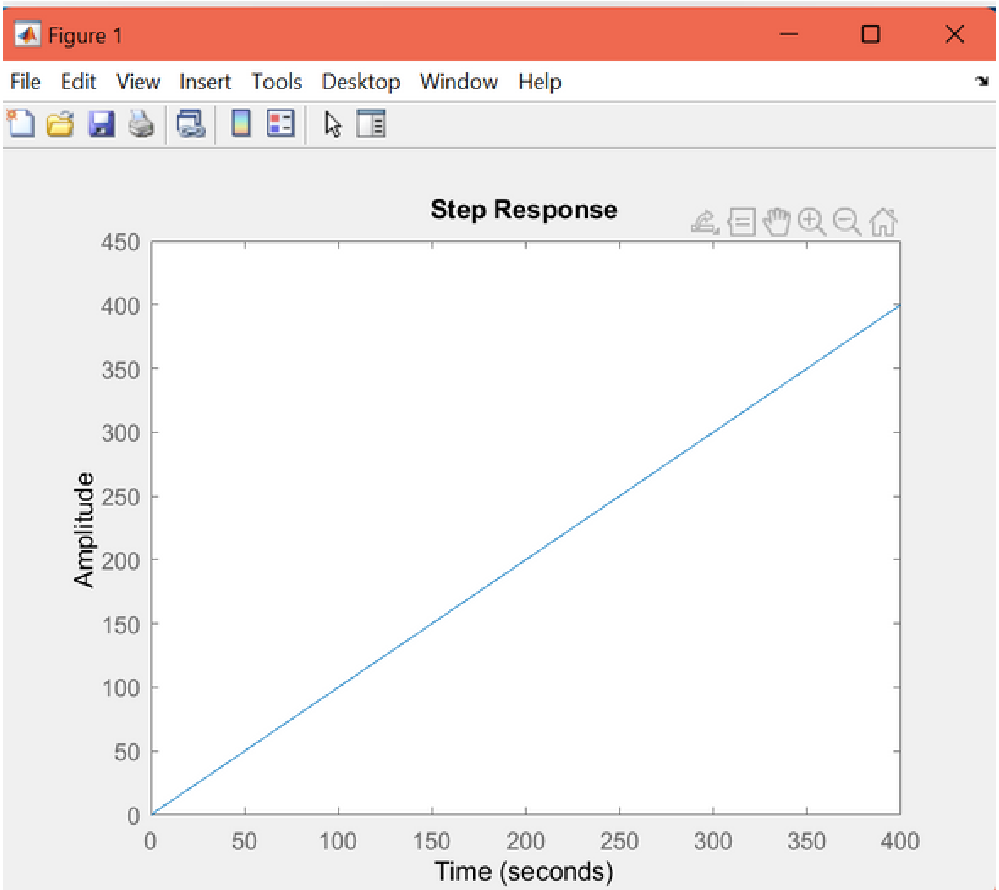




```
>> B2

B2 =

  logical

   1
```

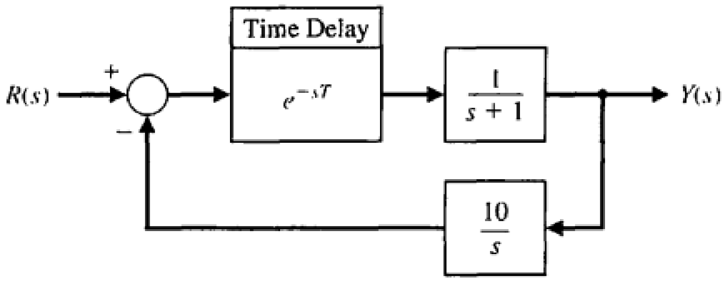
c)






```
>> B3

B3 =

  logical

   1
```

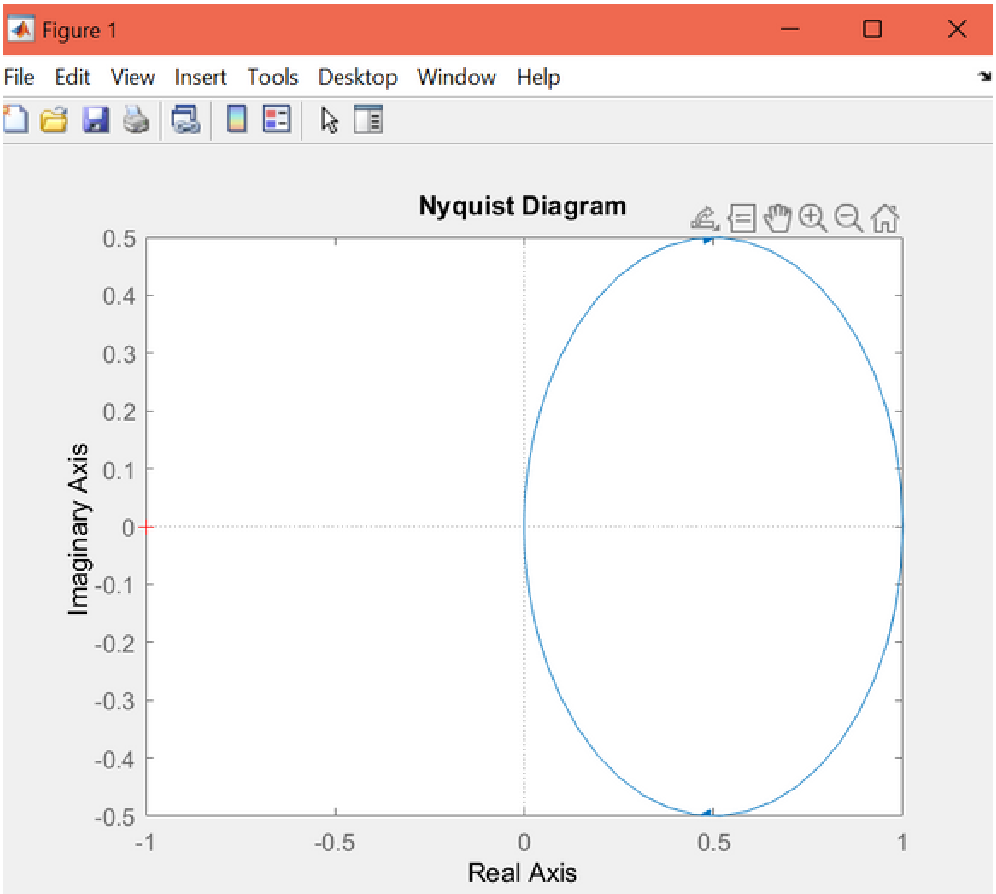
9. A closed-loop feedback system is shown in Figure below.



(a) Obtain the Nyquist plot and determine the phase margin. Assume that the time delay $T=0$ s. (b) Compute the phase margin when $T=0.05$ s. (c) Determine the minimum time delay that destabilizes the closed-loop system.

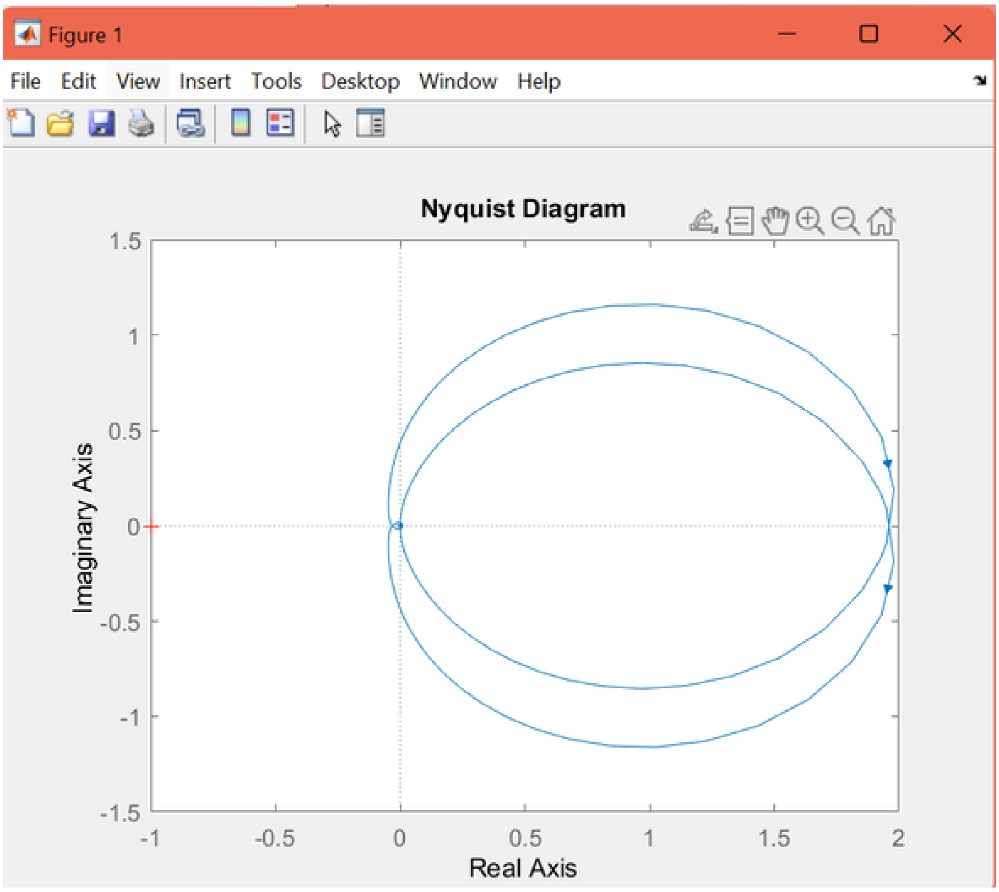
```
num=[1];
den=[1 1];
T=input("enter the time delay value: ")
P = tf(num,den,'InputDelay',T);
S=feedback(P,tf([10],[1 0]));
nyquist(S)
[Gm,Pm,Wcg,Wcp] = margin(S)
```

a)




```
enter the time delay value: 0

T =

     0


Gm =

   Inf


Pm =

  -180

Wcg =

   NaN


Wcp =

    3.1623
```

b)




```
enter the time delay value: 0.05

T =

    0.0500


Gm =

   31.8520


Pm =

  110.0276
```

```
Wcg =

   31.8384


Wcp =

    3.6007
```