1.1.)

Product of two polynomials: Let p1(s) and p2(s) be two polynomials. Use Matlab to determine the product of these two polynomials

Code:

```
clc;
p1=input("please enter coefficients of polynomial 1");
p2=input("please enter coefficients of polynomial 2");

tf=conv(p1,p2);
disp(tf)
```

Output:

```
please enter coefficients of polynomial 1
[1 2 2]
please enter coefficients of polynomial 2
[1 2 3]
     1     4     9    10     6


>>
```

1.2.)

Obtaining the overall transfer function (Block diagram algebra):
Let $G1(s) = n1(s)/d1(s)$ and $G2(s) = n2(s)/d2(s)$ be the two given transfer functions, where $G1(s)$ and $G2(s)$ are transfer functions of arbitrary order.

1.2.1. Transfer functions in cascade or series: Write a program that allows the user to input the numerator and denominator polynomials of two transfer functions, and outputs the numerator and denominator polynomials of the overall transfer function.

1.2.2. Transfer functions in parallel: Write a program that allows the user to input the numerator and denominator polynomials of two transfer functions, and outputs the numerator and denominator polynomials of the overall transfer function.

1.2.3. Transfer functions in a feedback loop: Write a program that allows the user to input the numerator and denominator polynomials of two transfer functions, and outputs the numerator and denominator polynomials of the overall transfer function. Here, $G1(s)$ is the forward path transfer function, and $G2(s)$ is the feedback path transfer function.

Code:

block_diagram_reduction.m × cs1.m * ×

```
n1=input("Numerator 1:");d1=input("denominator 1");
n2=input("Numerator 2");d2=input("denominator 2");
%1.2.1 The blocks are cascaded
n=conv(n1,n2);
d=conv(d1,d2);
fprintf("The transfer function when the blocks are cascaded:");
sys=tf(n,d)
%1.2.2 parallel system
np=conv(n1,d2)+conv(n2,d1);
d=conv(d1,d2);
fprintf("The transfer function when the blocks are parallel:");
sysp=tf(np,d)
%1.2.3 feedback system
%negative feedback
 num=conv(n1,d2);
 den=conv(d1,d2)+conv(n1,n2);
 fprintf("The transfer function for the negative feedback system:");
 sysf=tf(num,den)
%positive feedback
 num1=conv(n1,d2);
 den1=conv(d1,d2)-conv(n1,n2);
 fprintf("The transfer function for the positive  feedback system:");
 syspf=tf(num1,den1)
```

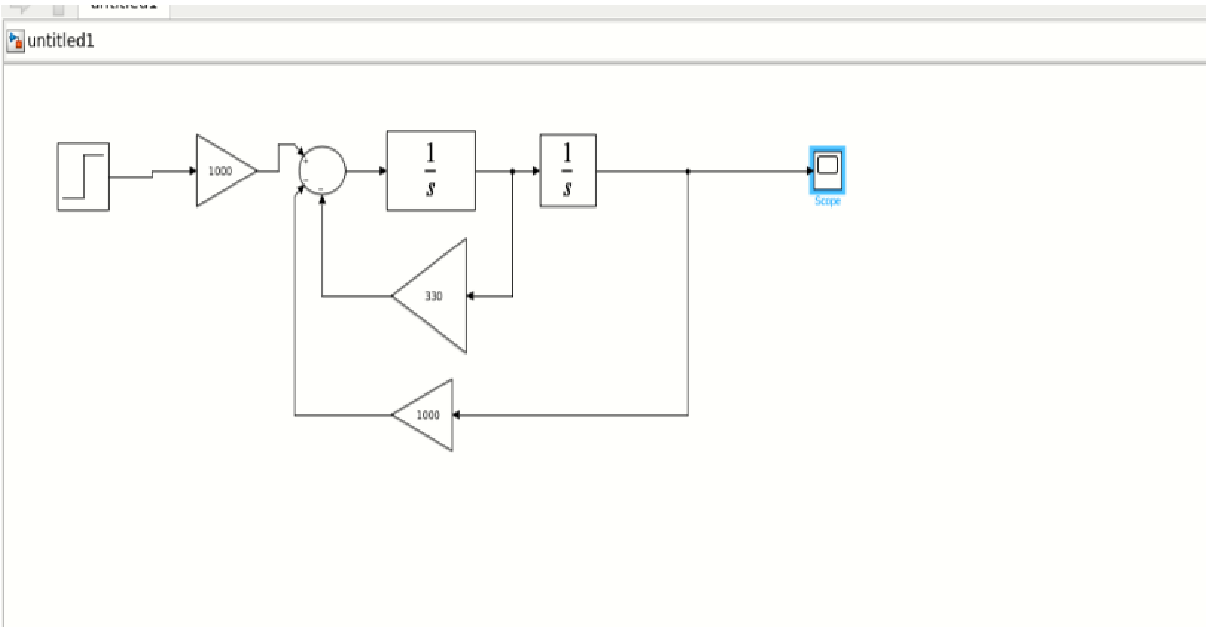
Output:

```
>> cs1
Numerator 1:
[1 2 1]
denominator 1
[0 0 2]
Numerator 2
[2 2 1]
denominator 2
[0 2 3]
The transfer function when the blocks are cascaded:
sys =

  2 s^4 + 6 s^3 + 7 s^2 + 4 s + 1
  -------------------------------
              4 s + 6

Continuous-time transfer function.

The transfer function when the blocks are parallel:
sysp =

  2 s^3 + 11 s^2 + 12 s + 5
  -------------------------
           4 s + 6
```

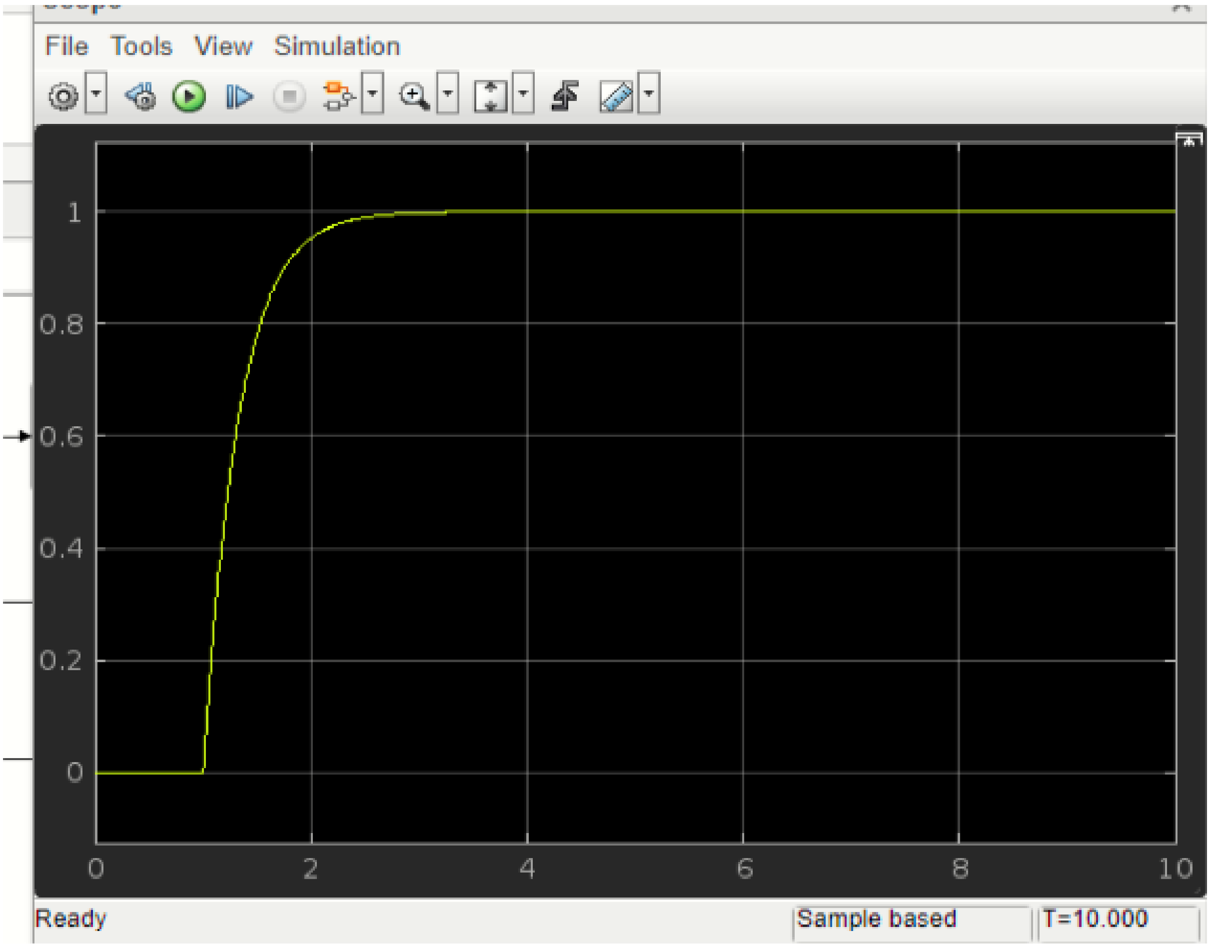
```
The transfer function for the negative feedback system:
sysf =

      2 s^3 + 7 s^2 + 8 s + 3
  -------------------------------
  2 s^4 + 6 s^3 + 7 s^2 + 8 s + 7

Continuous-time transfer function.

The transfer function for the positive  feedback system:
syspf =

  -2 s^3 - 7 s^2 - 8 s - 3
  -------------------------
  2 s^4 + 6 s^3 + 7 s^2 - 5

Continuous-time transfer function.

>>
```

2. The objective of this exercise is to learn to use Simulink. Consider a circuit with a resistance R, an inductance L, and a capacitance C, all in series with a voltage source. 2.1. Determine the transfer function from the applied voltage to the voltage across the capacitance. Hence, obtain the differential equation that governs the dynamics of the system. 2.2. Rewrite this differential equation by expressing the highest derivative of the dependent variable in terms other derivatives of the dependent variable and the input. 2.3. Implement this in simulink as follows: At the Matlab command prompt type simulink. A Simulink library browser opens up. Using the pull-down menu listed under File open a new model. Double click on the Continuous simulink library. Copy the integrator block onto the new model that you created. Create another copy, and connect the two integrators. (Let these two integrators be named One and Two.) From the commonly used blocks get a copy of the summer (sum) the gain block, and the scope block. Connect the output of Two to the scope. Make a second copy of the gain block, and connect the output of one to one gain block and Two to the other gain block. (For visual correctness, it is possible to flip the direction of the gain block by using Ctrl-r.) Double click on the summer and add one more input to the block. Connect the outputs of the gain blocks to two of these inputs. Connect the step block (from the sources simulink library) to the first input through an appropriate gain block. For C = 0.01 F, L = 0.1 H, and R = 33 ohms,simulate and observe the response.

Block diagram:



Output:

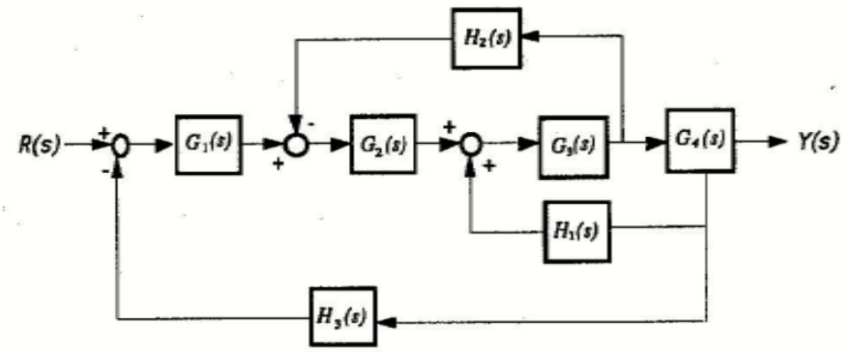


3. The objective of this exercise is to use the building blocks for block-diagram algebra for transfer functions developed in question 1., and to perform basic analysis.

3.1.Using the program/programs developed in question 1., determine the overall transfer function for the figure shown below. The individual transfer functions are as follows:

$$G_1(s)=\frac{1}{s+10};G_2(s)=\frac{1}{s+1};G_3(s)=\frac{s^2+1}{s^2+4s+4};G_4(s)=\frac{s+1}{s+6};$$

$$H_1(s)=\frac{s+1}{s+2};H_2(s)=2;H_3(s)=1$$



Code:

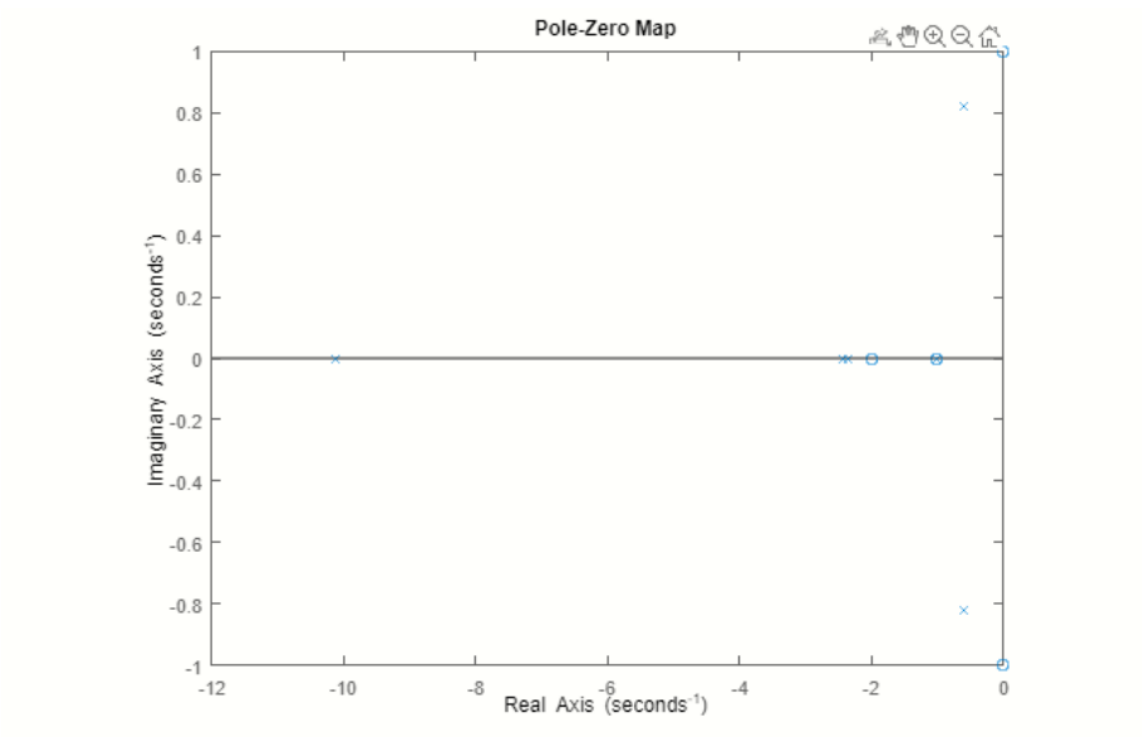
```
clc;clear ;close all;
ng1=[0 0 1];
ng2=[0 0 1];
ng3=[1 0 1];
ng4=[0 1 1];
nh1=[0 1 1];
nh2=[0 0 2];
nh3=[0 0 1];
dg1=[0 1 10];
dg2=[0 1 1];
dg3=[1 4 4];
dg4=[0 1 6];
dh1=[0 1 2];
dh2=[0 0 1];
dh3=[0 0 1];

%H2/G4
prn=[0 2 12];
prd=[0 1 1];

%H2/G4=2s+12/s+1
[nc1,dc1]=blockdiag(ng3,ng4,dg3,dg4,'series');
[nf1,df1]=blockdiag(nc1,nh1,dc1,dh1,'pfeedb');
[nc2,dc2]=blockdiag(nf1,ng2,df1,dg2,'series');
[nf2,df2]=blockdiag(nc2,prn,dc2,prd,'nfeedb');
[nc3,dc3]=blockdiag(ng1,nf2,dg1,df2,'series');
[nf,df]=blockdiag(nc3,nh3,dc3,dh3,'nfeedb');
sys=tf(nf,df)
```

Output:

```
sys =

              s^5 + 4 s^4 + 6 s^3 + 6 s^2 + 5 s + 2
  ----------------------------------------------------------------
  12 s^6 + 205 s^5 + 1066 s^4 + 2517 s^3 + 3128 s^2 + 2196 s + 712

Continuous-time transfer function.

>>
```

3.2. Generate a pole-zero map of the closed-loop transfer function in graphical form using the pzmap function.

Pole zero plot:



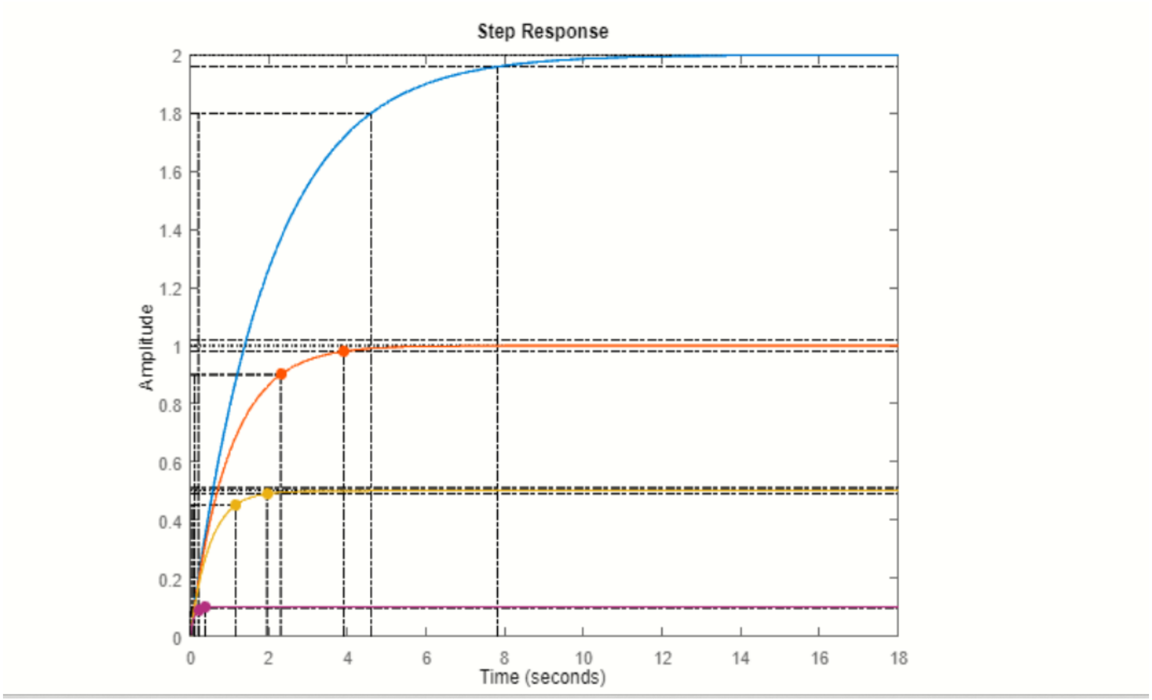
4. The objective of this exercise is to deduce the effect of location of pole and zero on the time-domain response of a system.

a. First-order systems: Consider $G1(s) = 1/s+p$. Compare in terms of rise time and steady-state value the step responses of this system for different values of p. Choose p = 0.5, 1, 2, and 10. (For purposes of this experiment, assume the following definition of rise time: the time taken for the output to reach 90% of the final steady-state value.) Based on this, where should one locate the pole if the requirement is a fast response? Where should one locate the pole if the steady-state value of the output is expected to be equal to the input value? Can one independently satisfy both requirements?

Code:

```
%first order system
clc;
clear all;
close all;

n1=[0 1];
d1=[1 0.5];
step(n1,d1)
hold on;

n2=[0 1];
d2=[1 1];
step(n2,d2)
hold on;

n3=[0 1];
d3=[1 2];
step(n3,d3)

n4=[0 1];
d4=[1 10];
step(n4,d4)
```

Output:

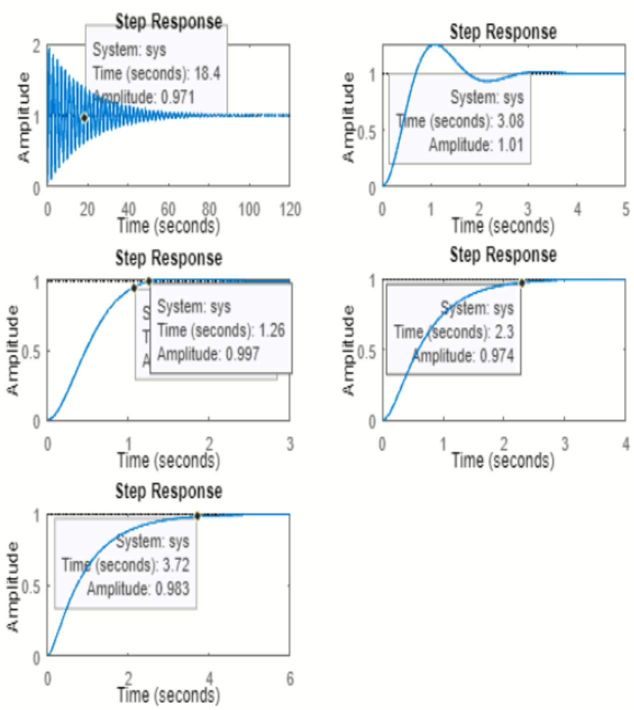


b. Second-order systems: Consider $G2\ (s)$ = 10s/2+as+10 . Compare in terms of rise time, the settling time, the peak overshoot, and steady-state value the step responses of this system for different values of a: Choose a = 0.1, 2.5, 5, 7.5, 10. (Use the definitions in the prescribed text-book.)

Code:

```
%2nd order system
clc;
clear all;
close all;
n1=[0 0 10];
d1=[1 0.1 10];
subplot(321);
step(n1,d1)
n2=[0 0 10];
d2=[1 2.5 10];

subplot(322);
step(n2,d2)
n3=[0 0 10];
d3=[1 5 10];

subplot(323)
step(n3,d3)
n4=[0 0 10];
d4=[1 7.5 10];

subplot(324);
step(n4,d4)
n5=[0 0 10];
d5=[1 10 10];

subplot(325)
step(n5,d5)
```

Output:

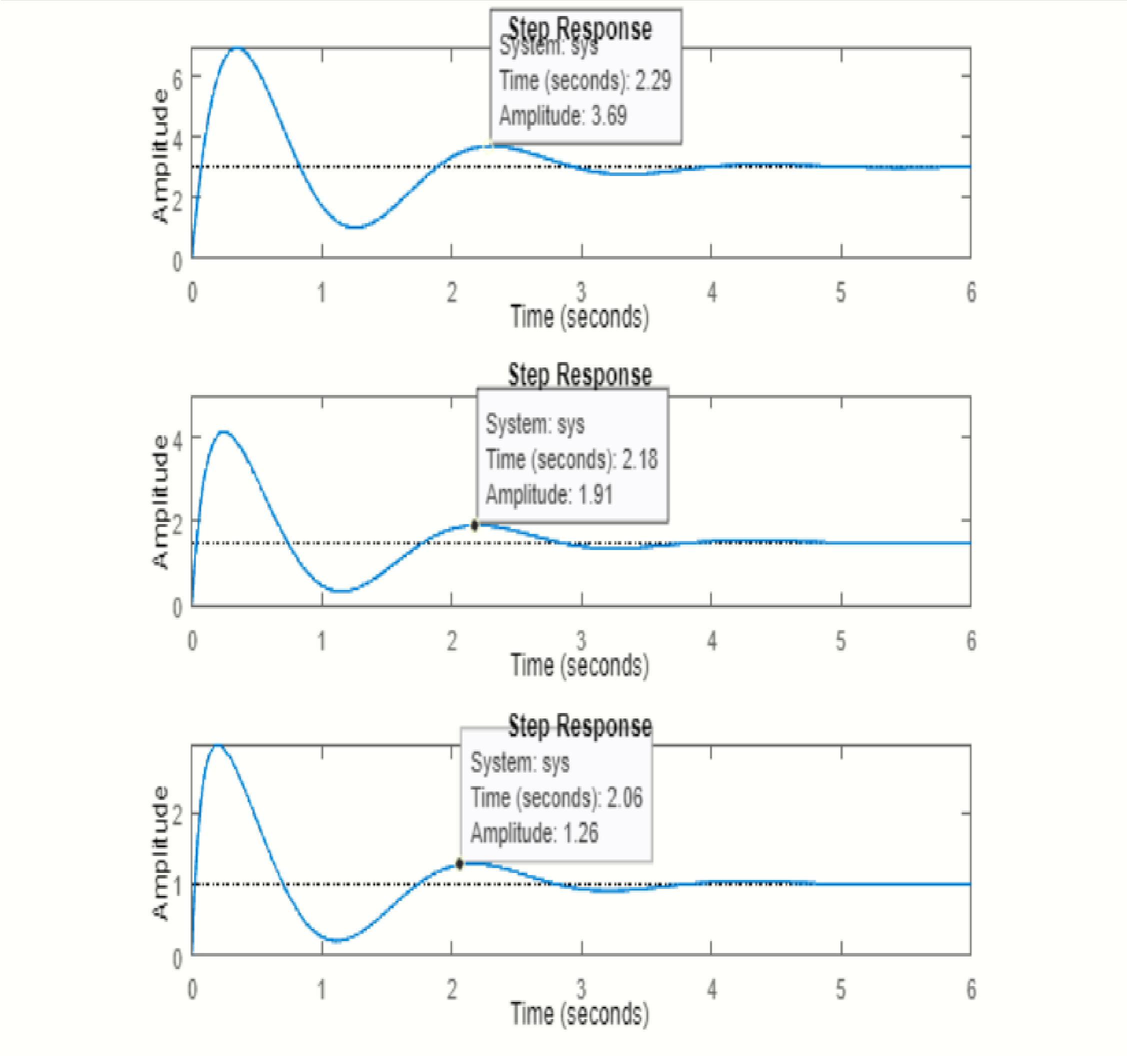


c. The effect of an additional pole: Consider $G3\ (s)$ = 10*s /2s^2+2s+10 in cascade with a first order system $G4\ (s)$ = $p/s+p$ Repeat the experiment 4.b for different values of p. Choose p = 5, 10, 20. In your discussions, include as well a comparison of these results with those obtained in 4.b.

Code:

```
clc;clear;close all;
p=[5 10 15];

g3n=10;
g3d=[1 2 10];
g4n=p;

for i=1:length(p)
    g4d=[0 1 p(i)];
    [num,den]=blockdiag(g3n,g4n,g3d,g4d,'series');
    sys=tf(num,den);
    display(sys)
    subplot(3,1,i)
    fprintf('p->%d',p(i));
    impulse(sys)
    step(num,den)
end
```

Output:


Step Response
System: sys
Time (seconds): 2.29
Amplitude: 3.69
Amplitude
Time (seconds)
Step Response
System: sys
Time (seconds): 2.18
Amplitude: 1.91
Amplitude
Time (seconds)
Step Response
System: sys
Time (seconds): 2.06
Amplitude: 1.26
Amplitude
Time (seconds)

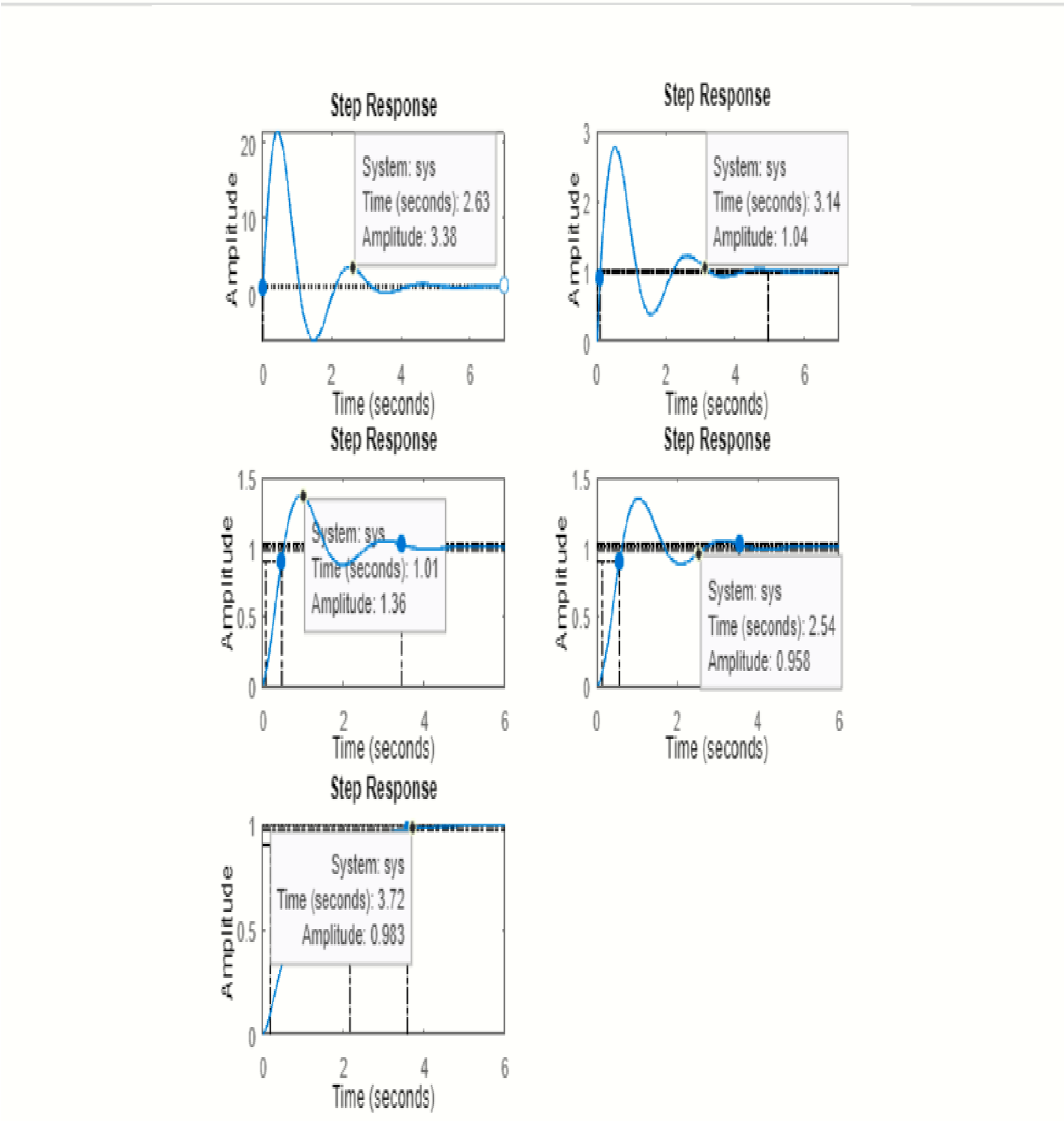
d.The effect of an additional zero: Consider $G4\ (s) = 10(\ s/a\ +1)\ /s$^2+2$s$+10 . Repeat the experiment 4.b for different values of a. Choose a = 0.1, 1, 10, 100. In your discussion, include as well a comparison of these results with those obtained in 4.b.

```
% as zero is dominant the rise time decreases hence the system is faster
```

code:

```
a=0.1;

n=[ 0 10/a 10];
d=[1 2 10];
subplot(321)
step(n,d)


a=1;
hold on;
n=[ 0 10/a 10];
d=[1 2 10];
subplot(322)
step(n,d)

a=10;
n=[ 0 10/a 10];
d=[1 2 10];
subplot(323)
step(n,d)


a=100;
n=[ 0 10/a 10];
d=[1 2 10];

subplot(324)

step(n,d)
```

output:

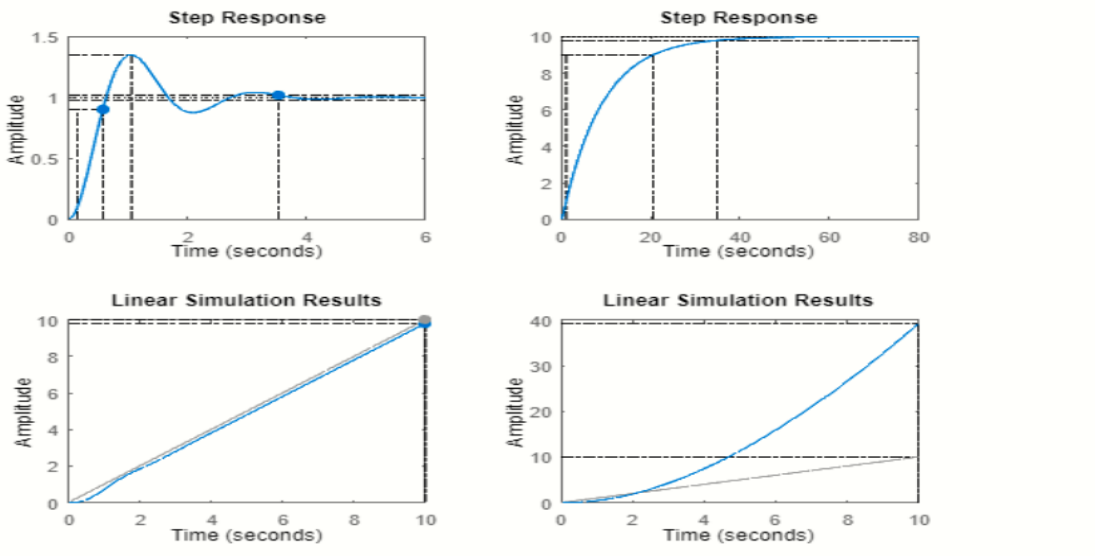

Step Response
System: sys
Time (seconds): 2.63
Amplitude: 3.38
Amplitude
Time (seconds)
Step Response
System: sys
Time (seconds): 3.14
Amplitude: 1.04
Amplitude
Time (seconds)
Step Response
System: sys
Time (seconds): 1.01
Amplitude: 1.36
Amplitude
Time (seconds)
Step Response
System: sys
Time (seconds): 2.54
Amplitude: 0.958
Amplitude
Time (seconds)
Step Response
System: sys
Time (seconds): 3.72
Amplitude: 0.983
Amplitude
Time (seconds)

5. The objective of this exercise is to compare the response of systems to different kinds of inputs. Consider the two systems $G3\ (s)$ = 10s/2+2s+10 and $G4\ (s)$ = 10(s+1) /s^3+10s^2+10s+10 . Compare the responses of these systems to a step input and a unit-ramp input. Do the outputs follow or track the input? If so, why? If not, why not? Can one theoretically deduce these results?

code:

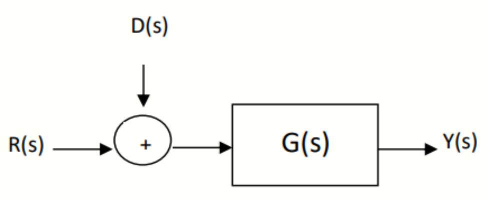
```
%Objective is to compare the response of systems to different inputs
clc;clear;close all;
g3n=[0 0 0 10];
g3d=[0 1 2 10];
g4n=[0 0 10 10];
g4d=[1 10 10 1];

%compare responses of these systems to step and unit-ramp inputs
%step response
subplot(221)
step(g3n,g3d);

subplot(222)
step(g4n,g4d);

sys3=tf(g3n,g3d);
sys4=tf(g4n,g4d);

t=0:0.01:10;
u=t;
subplot(223)
lsimplot(sys3,u,t);
subplot(224)
lsimplot(sys4,u,t)
```

Output:



6. The objective of this exercise is to deduce the effect of feedback on a system, specifically, a boring machine. (What were the boring machines that have been used for the "Namma Metro" construction?) The machines operating from both ends of a tunnel bore toward the middle. To link up accurately in the middle of the tunnel, a laser guidance system keeps the machines precisely aligned. (What would happen if this is not ensured?)

6.1.Open-loop response: Consider the following figure where the boring machine is represented by the transfer function G(s)=s(s +p) ,R(s) is the reference input which represents the desired angle-of-direction of travel of the machine, Y(s) is the actual angle-of-direction of travel, and D(s) is a disturbance input which represents the load on the machine. Compare the impulse reference response (i.e., r(t) = delta(t), d(t) = 0) for different values of p , the step reference response (i.e., r(t) = 1(t), d(t) = 0) for different values of p , and the situation wherein there is both a step reference and a specified disturbance input: r(t) =1(t) ,d(t)= {1,0<=t<=4
2,4<=t<=8} All simulations are to be carried out over the time range 0<=t <=8. Choose p = 1, 2,5 . Your discussions should include the effect of p on the steady-state values and the rise time. Can you theoretically determine these values?

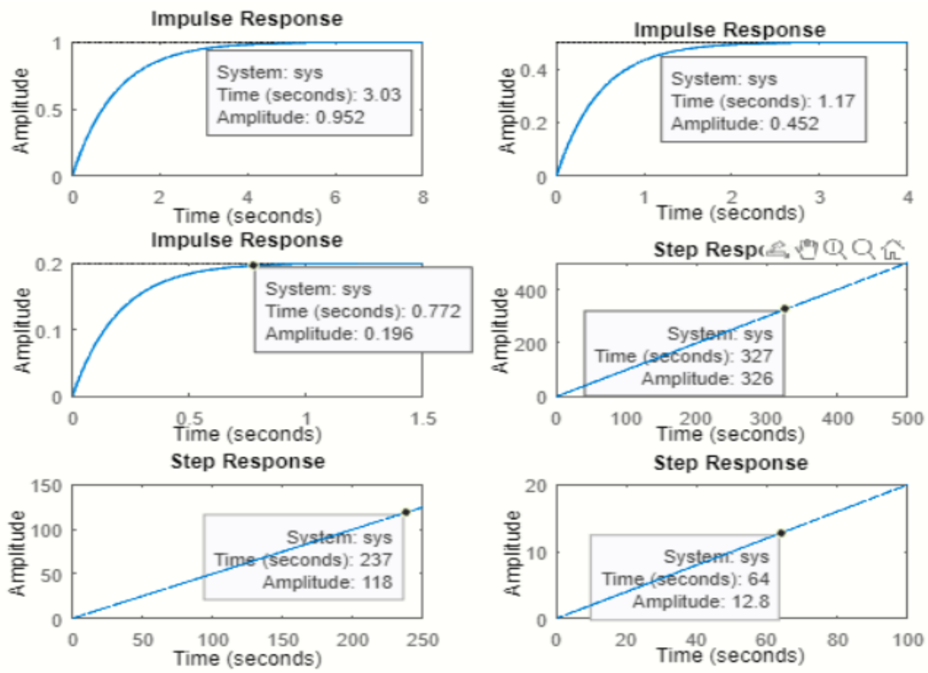


Code:

```
clc; clear ; close all;
num=1;
p=[1 2 5];
for i=1:length(p)
    den=[1 p(i) 0];
    sys=tf(num,den);
    subplot(3,2,i)
    impulse(sys);
    subplot(3,2,i+3)
    step(sys)
end
```

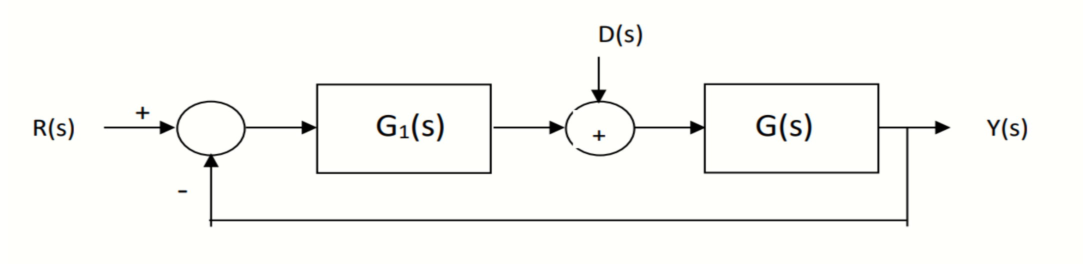
Output:

The subplots includes impulse responses and step responses of the system for p=1,2,5 sequentially.



6.2)Open-loop response: Consider the following figure where the boring machine is represented by the transfer function G1(s)=11s+k, R(s) is the reference input which represents the desired angle-of-direction of travel of the machine, Y(s) is the actual angle-of-direction of travel, and D(s) is a disturbance input which represents the load on the machine. Compare the impulse reference response (i.e., r(t) = delta(t), d(t) = 0) for different values of p , the step reference response (i.e., r(t) = 1(t), d(t) = 0) for different values of p , and the situation wherein there is both a step reference and a specified disturbance input: r(t) =1(t) ,d(t)= {1,0<=t<=4
2,4<=t<=8} All simulations are to be carried out over the time range 0<=t <=8. Choose p = 1, 2,5 .

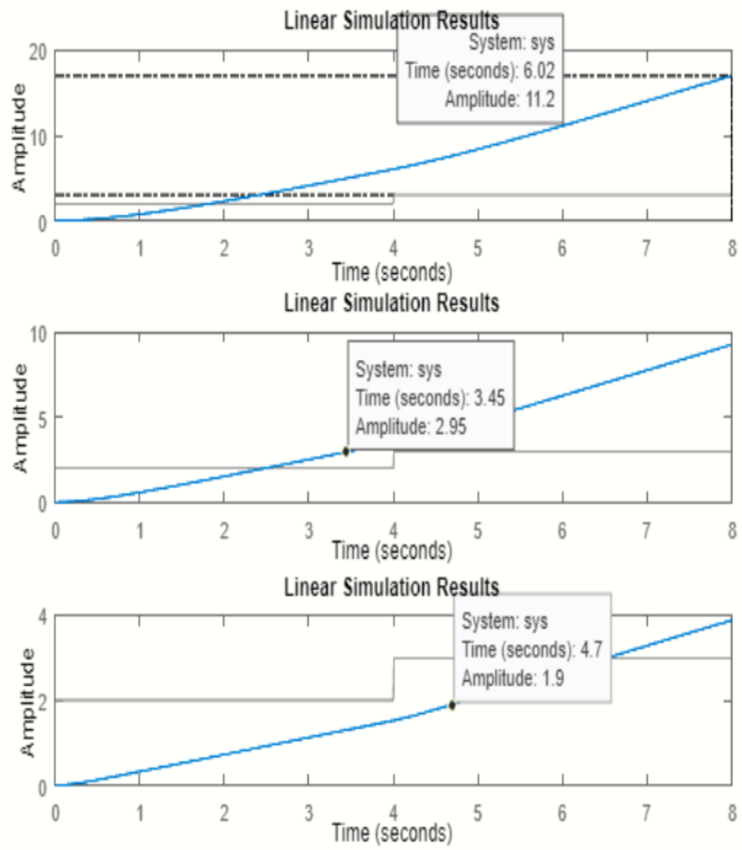
Choose p = 1, 2, 5, and consider two different situations: K = 20, 100. Compare these results with those obtained in 4.1a. Your discussions should include the effect of p on the steady-state values and the rise time. Can you theoretically determine these values?



Code:

```
clc;clear;close all;
t1=0:0.01:4;
t2=4.01:0.01:8;
t=[t1 t2];
p=[1,2,5];
u=[2*ones(1,length(t1)),3*ones(1,length(t2))];
for i=p:length(p)
    sys=tf(1,[1 p(i) 0]);
    subplot(3,1,i)
    lsimplot(sys,u,t)
end
```

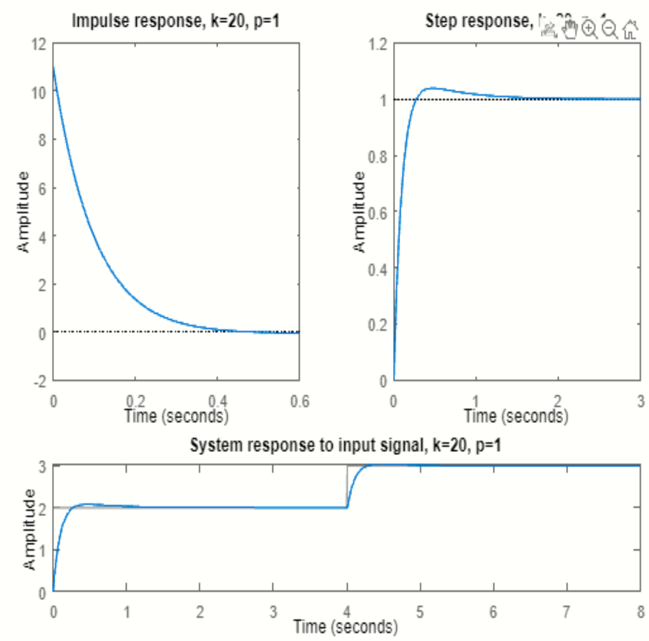
Output:

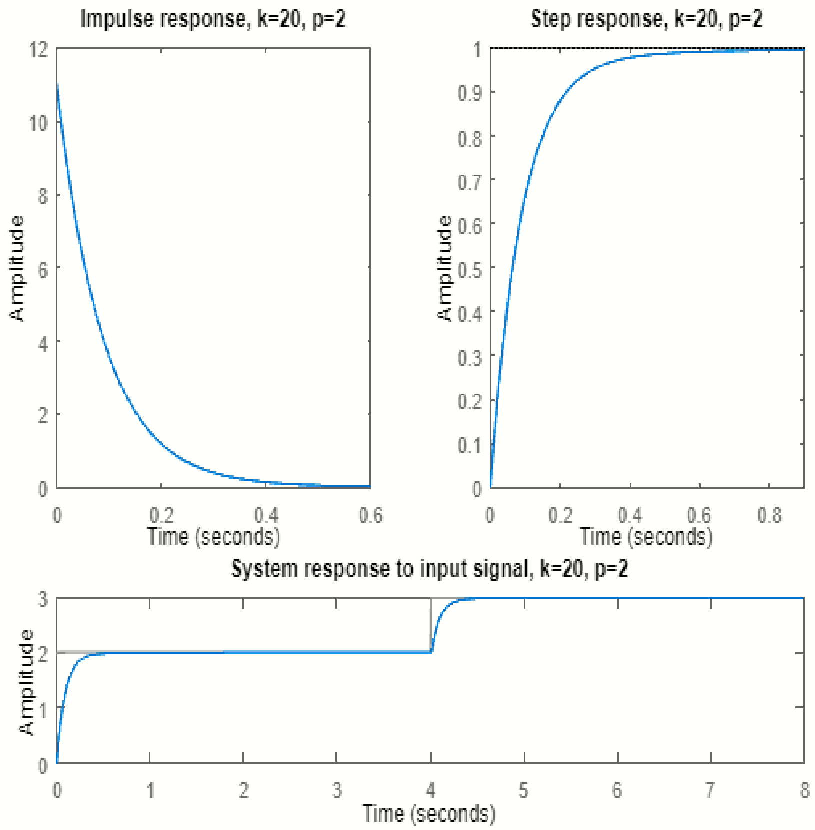


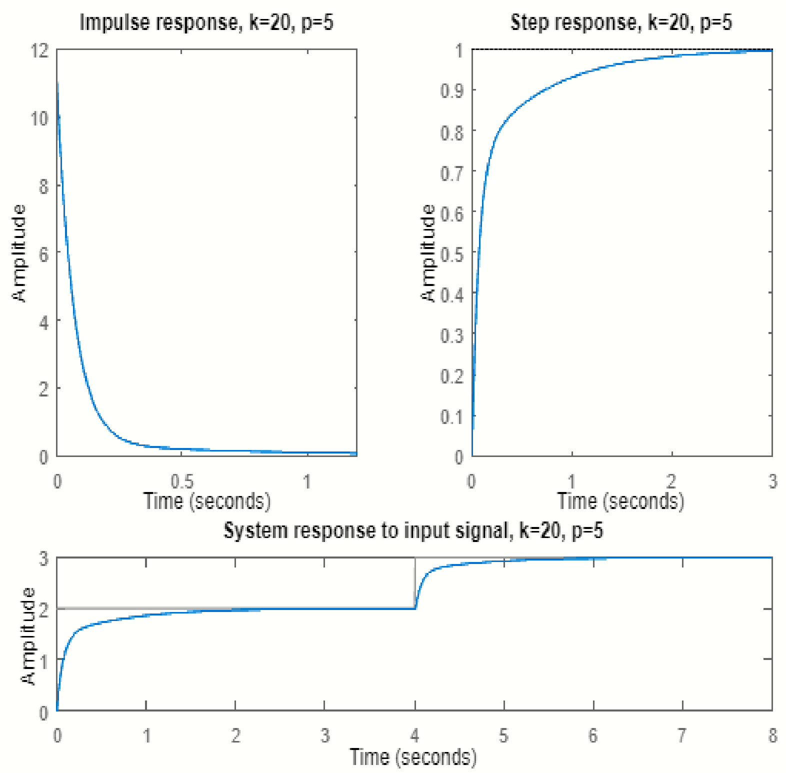
Code:

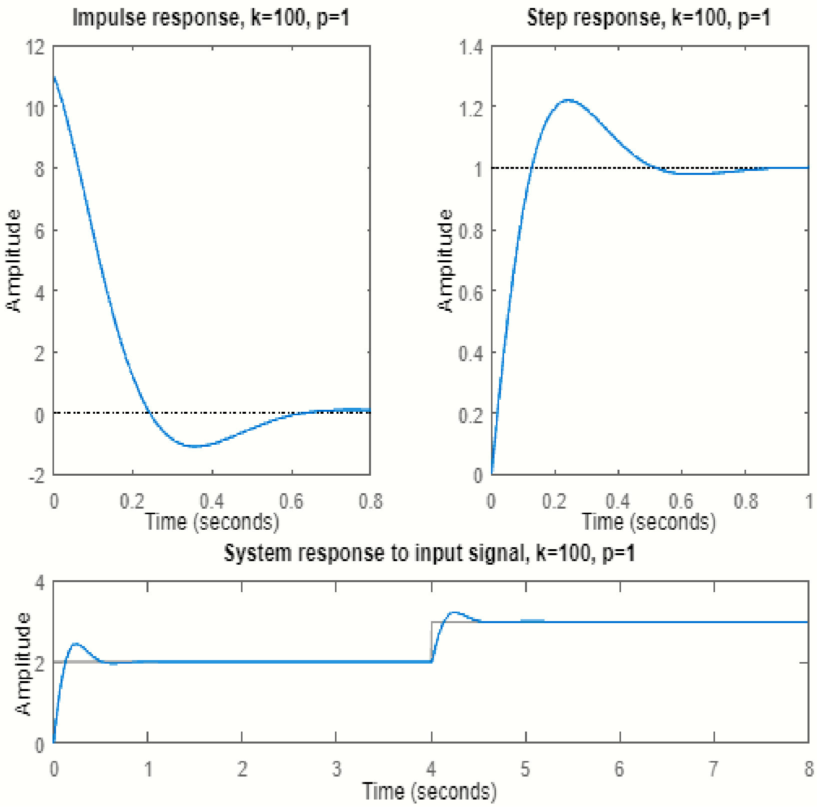
```matlab
clc; clear all; close all;
t1 = 0:0.01:4;
t2 = 4.01:0.01:8;
t = [t1 t2];
p = [1, 2, 5];
k = [20, 100];
u = [2*ones(1, length(t1)), 3*ones(1, length(t2))];

for i = 1:length(p)
    for j = 1:length(k)
        n1 = [0 0 1];
        d1 = [1 p(i) 0];
        n2 = [0 11 k(j)];
        d2 = [0 0 1];
        [n, d] = blockdiag(n1, n2, d1, d2, 'series');
        [n, d] = blockdiag(n, 1, d, 1, 'nfeedb');
        sys = tf(n, d);

        % Plot impulse and step response
        figure;
        subplot(3, 2, [1, 3]);
        impulse(sys);
        title(sprintf('Impulse response, k=%d, p=%d', k(j), p(i)));
        subplot(3, 2, [2, 4]);
        step(sys);
        title(sprintf('Step response, k=%d, p=%d', k(j), p(i)));

        % Plot system response to input signal
        subplot(3, 1, 3);
        lsimplot(sys, u, t);
        title(sprintf('System response to input signal, k=%d, p=%d', k(j), p(i)));
    end
end
```

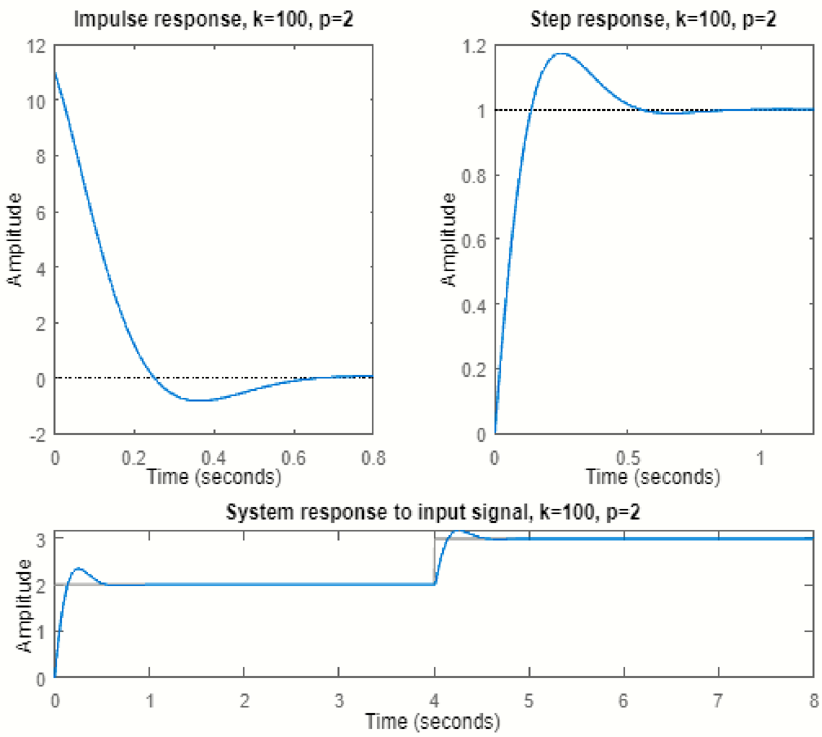
Output:

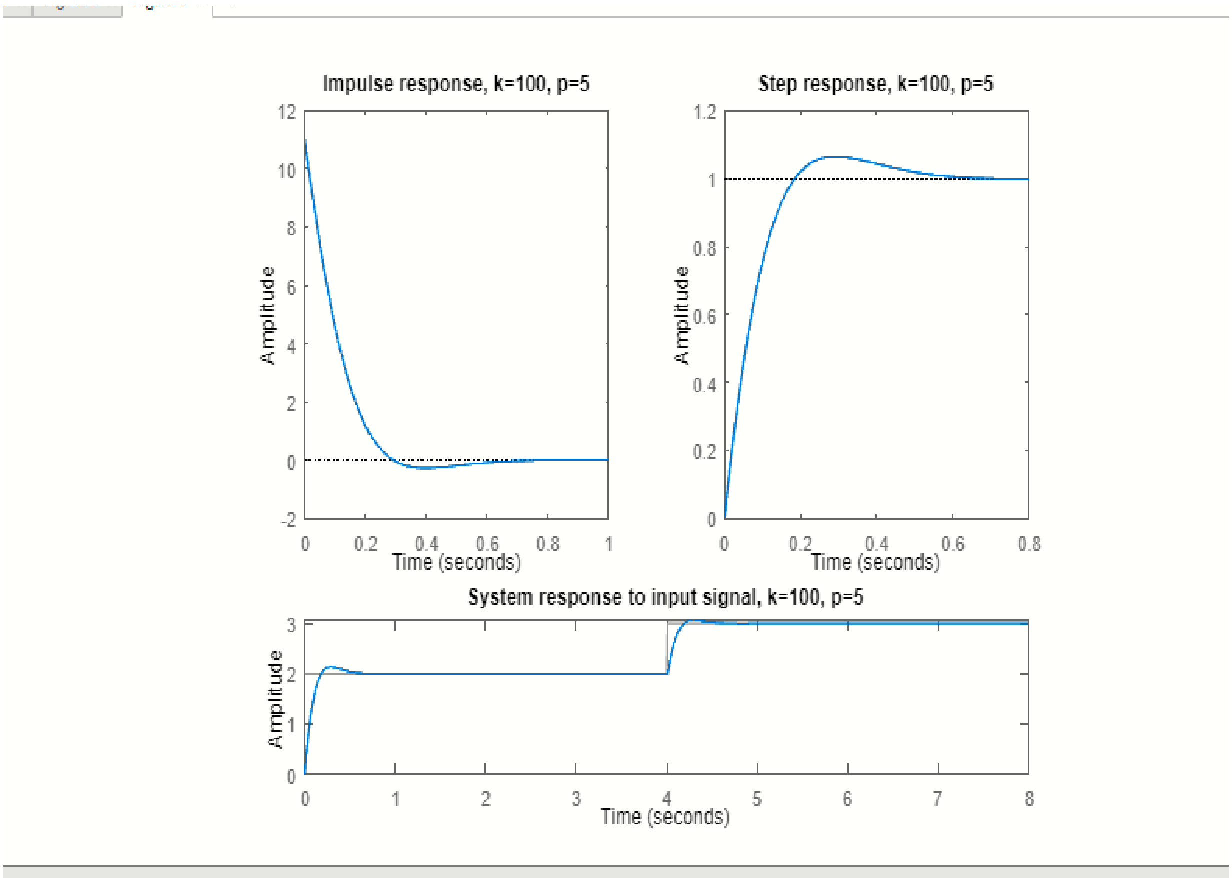



Impulse response, k=20, p=2
Step response, k=20, p=2
System response to input signal, k=20, p=2
Amplitude
Time (seconds)


Impulse response, k=20, p=5
Step response, k=20, p=5
System response to input signal, k=20, p=5
Amplitude
Time (seconds)


Impulse response, k=100, p=1
Step response, k=100, p=1
System response to input signal, k=100, p=1
Amplitude
Time (seconds)


Impulse response, k=100, p=2
Step response, k=100, p=2
System response to input signal, k=100, p=2
Amplitude
Time (seconds)



Impulse response, k=100, p=5
Step response, k=100, p=5
System response to input signal, k=100, p=5
Amplitude
Time (seconds)