İSTANBUL TEKNİK ÜNİVERSİTESİ

1773

KON326E COMPUTER CONTROLLED SYSTEMS

PROJECT 3

Prof. Dr. Yaprak Yalçın

TEAM :

Abdulbaki Aybakan / 040120523

Kanan Muradov / 040120914

Yousef Rahwan / 040140922

Özge Kaya

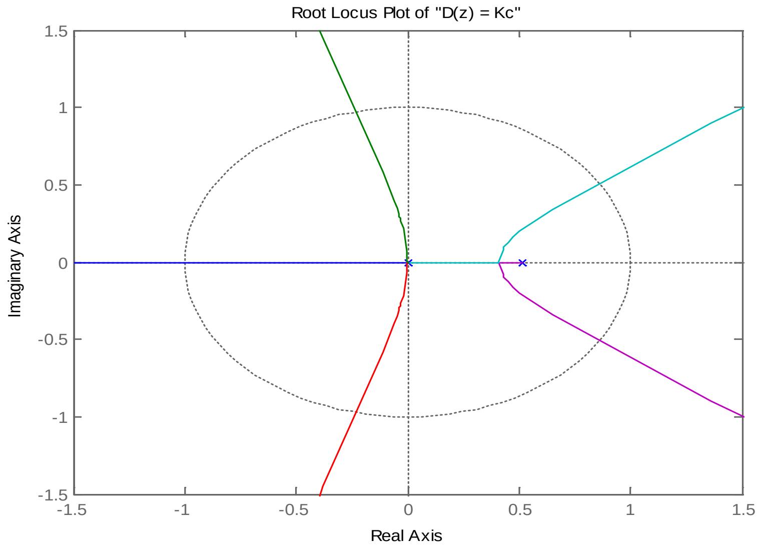
2015-2016

QUESTION-1-a :

The matlab code we used is

```
s = tf('s');

K = 0.6;
Td=0.08;
Ts=0.02;
Tau=0.03;

H = tf([K],[Tau 1],'InputDelay', Td)
sys_d = c2d(H,Ts,'zoh')

rlocus(sys_d)

title('Root Locus Plot of "D(z) = Kc"');
----------------------------------------------
The results are;
H =

                    0.6
  exp(-0.08*s) * ---------
                 0.3 s + 1

Continuous-time transfer function.


Sys_d =

             0.0387
  z^(-4) * ----------
           z - 0.9355

Sample time: 0.02 seconds
Discrete-time transfer function.
```

```
And the plot is as below;
```



QUESTION-1-b :

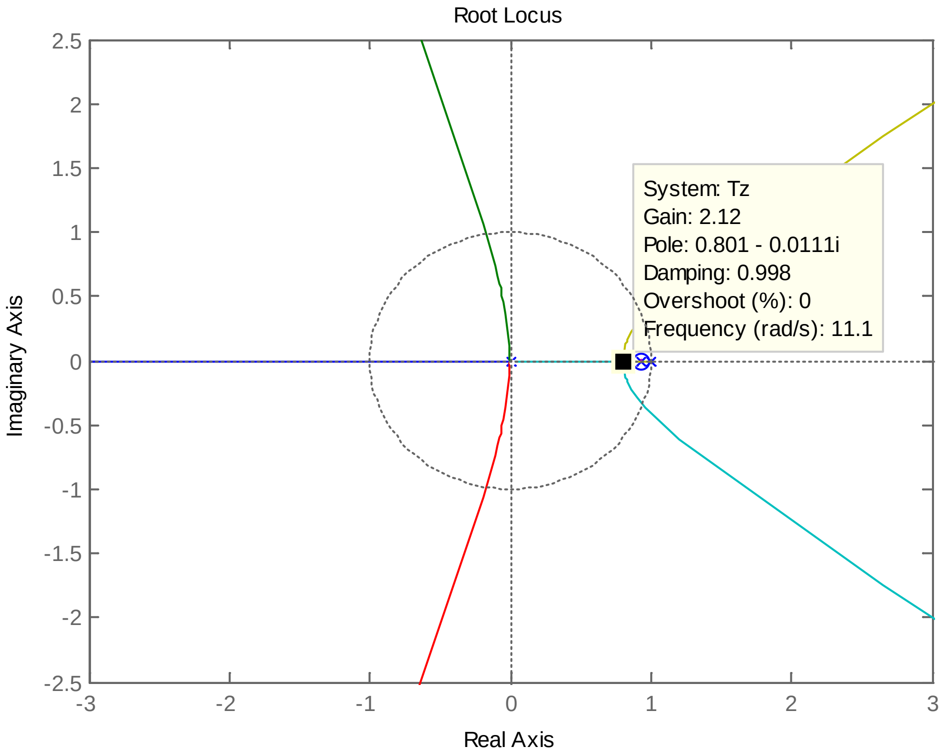
As we can see that the pole of H is **`0.9355`**

The matlab code we used is;

```
s = tf('s');
z = tf('z');

K = 0.6;
Td = 80*10^-3;
Tau = 300*10^-3;
Ts = 20*10^-3;
A = 0.9355

H = tf([K],[Tau 1],'InputDelay', Td)
sys_d = c2d(H,Ts,'zoh')
Dz = (z-A)/(z-1);
Tz = Dz*sys_d

rlocus(Tz)
```

```
The results are;

Tz =

             0.0387 z - 0.01987
  z^(-4) * ----------------------
           z^2 - 1.936 z + 0.9355

Sample time: 0.02 seconds
Discrete-time transfer function.
```

```
And the plot is as below;
```



QUESTION-1-c :

As we have taken the plot in part b with selected the zero overshoot point, Kc must be 2.12.

```
s = tf('s');
z = tf('z');

K = 0.6;
Td = 80*10^-3;
Tau = 300*10^-3;
Ts = 20*10^-3;
A = 0.9355;
Kc=2.12;

H = tf([K],[Tau 1],'InputDelay', Td);
sys_d = c2d(H,Ts,'zoh');
Dz = (z-A)/(z-1);
Tz = Kc*Dz*sys_d;
Tz_closed = = feedback(Tz,1) % The closed transfer function
with unit feedback
```

```
The result;
```

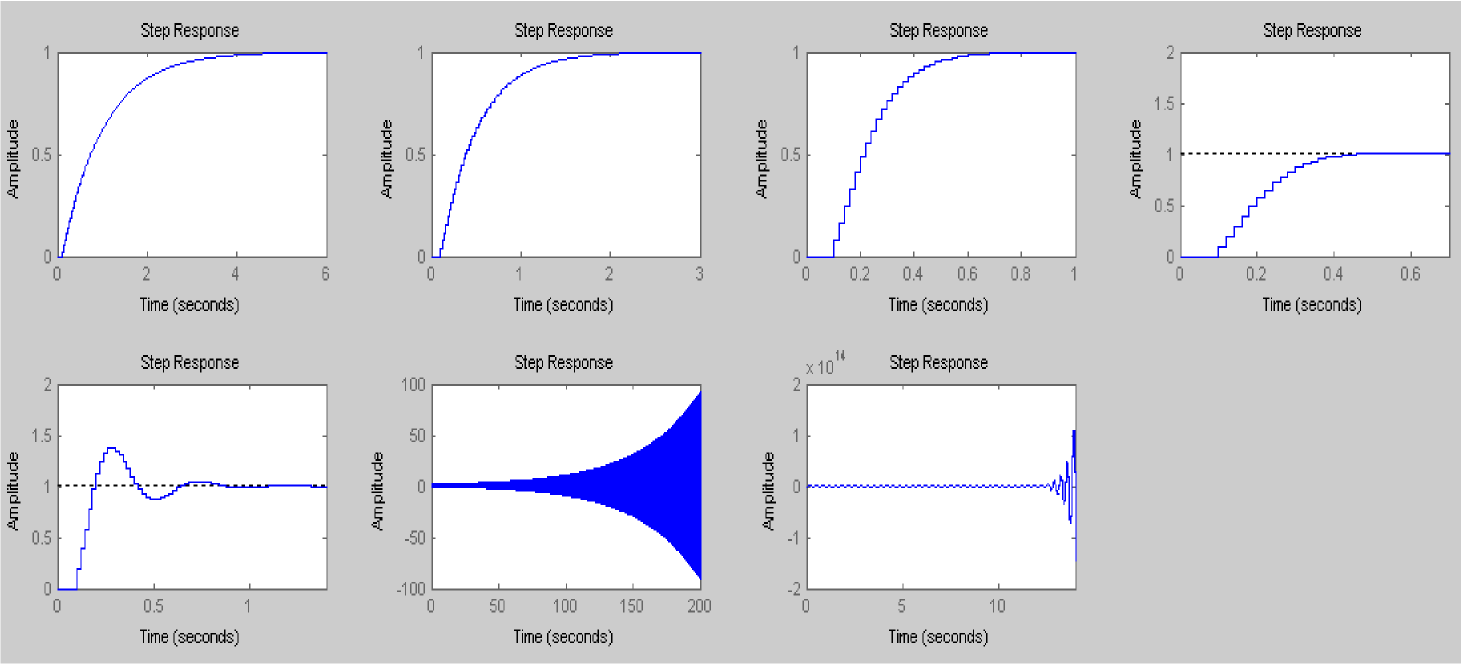
```
Tz_closed =

              0.08204 z - 0.07674
  -------------------------------------------------------
  z^6 - 1.936 z^5 + 0.9355 z^4 + 0.08204 z - 0.07674

Sample time: 0.02 seconds
Discrete-time transfer function.
```

QUESTION-1-d :

We have given 7 different values to Kc in order to see the difference. And these values are `0.5, 1, 2.12, 2.5, 5, 9, 12, 15.`

Our code;

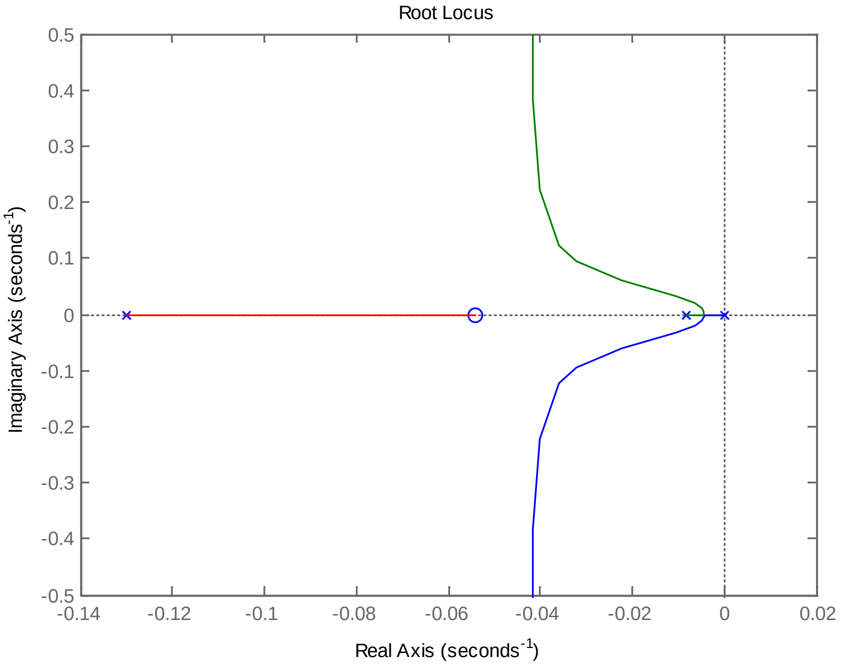
```
s = tf('s');
z = tf('z');

K = 0.6;
Td = 80*10^-3;
Tau = 300*10^-3;
Ts = 20*10^-3;
A = 0.9355;
Kc = [0.5 1 2.12 2.5 5 9 12 15];

H = tf([K],[Tau 1],'InputDelay', Td);
sys_d = c2d(H,Ts,'zoh');
Dz = (z-A)/(z-1);

for i=1:1:7;
    Tz = Kc(i)*Dz*sys_d;
    Tz_closed = feedback(Tz,1)
    step(Tz_closed)
     subplot(3,4,i)
end
```



Question-2

a)

```
>> s=tf('s');

>> Gs=(0.185*(18.5*s+1))/(s*(120*s+1)*(7.7*s+1));

>> rlocus(Gs)
```



a-1)

We have declared the Mp = 0.15 and Ts= 80 s and so,

$$\xi = 0.516931\ ve\ w_n = 0.0967247$$

According to $w_n x10 \div 15 \rightarrow w_s$,

$$w_s = 10w_n = 0.967247\ and\ \ T = \frac{2\pi}{w_s} = 6.49594$$

We have to choose a less sampling time than T value. $T = 0.6s\ is\ found.$.

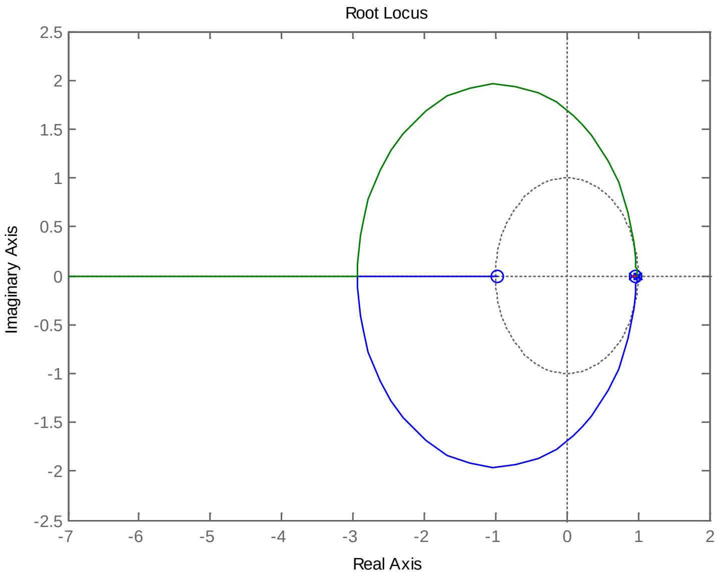
a-2)

PD controller can not eliminate the error so we can not use PD.

```
s=tf('s');
Gs=(0.185*(18.5*s+1))/(s*(120*s+1)*(7.7*s+1));
Gz=c2d(Gs,0.6)
rlocus(Gz)
```

Result;

```
Gz =


  0.0006557 z^2 + 9.983e-06 z - 0.0006242

  --------------------------------------

    z^3 - 2.92 z^2 + 2.84 z - 0.9204


Sample time: 0.6 seconds

Discrete-time transfer function.
```

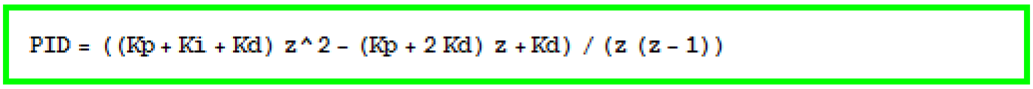


As you can see the stability interval is small.

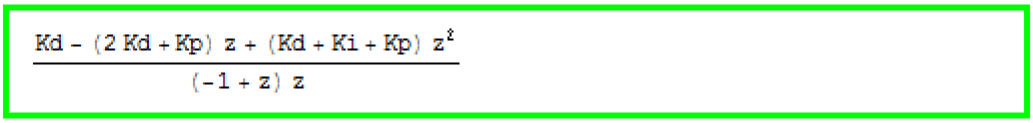
In z domain, there are poles so near to unit circle so system is so slow. Since the system has an integrator, control signal will have an increasing effect.

-Controller has been designed as PID olarak PID kontrolör tasarlandı. System response is quite well.

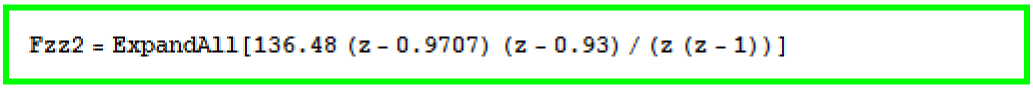
-PID is as below.



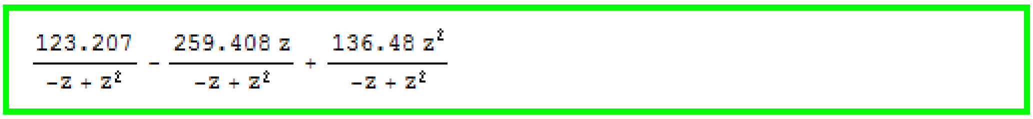
```
PID = ((Kp + Ki + Kd) z^2 - (Kp + 2 Kd) z + Kd) / (z (z - 1))
```



$$\frac{Kd - (2\,Kd + Kp)\,z + (Kd + Ki + Kp)\,z^2}{(-1 + z)\,z}$$

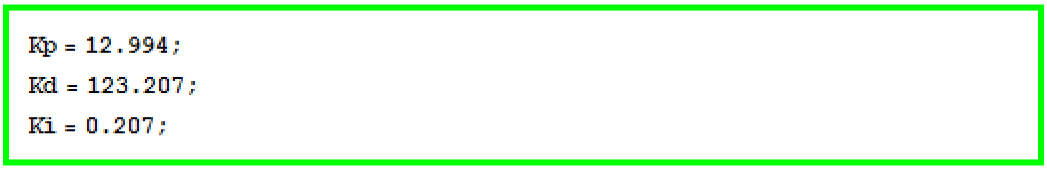


```
Fzz2 = ExpandAll[136.48 (z - 0.9707) (z - 0.93) / (z (z - 1))]
```



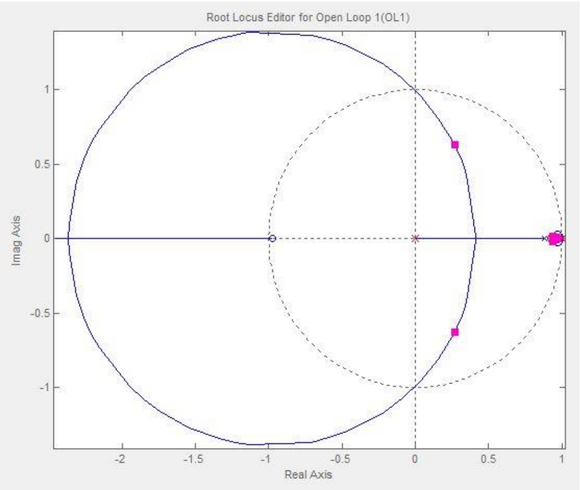
$$\frac{123.207}{-z + z^2} - \frac{259.408\,z}{-z + z^2} + \frac{136.48\,z^2}{-z + z^2}$$

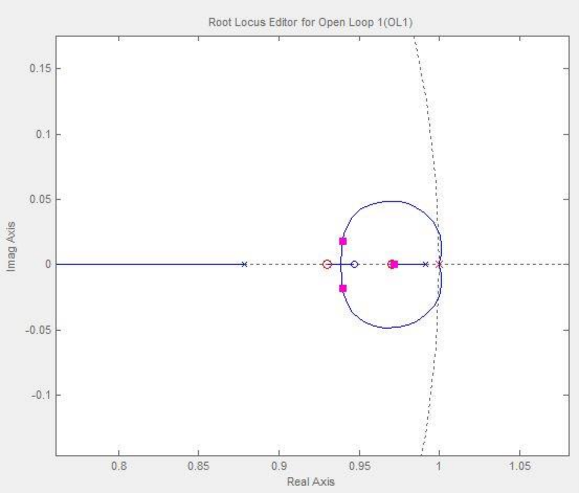
The parameters of Kp, Ki ve Kd are found as below;



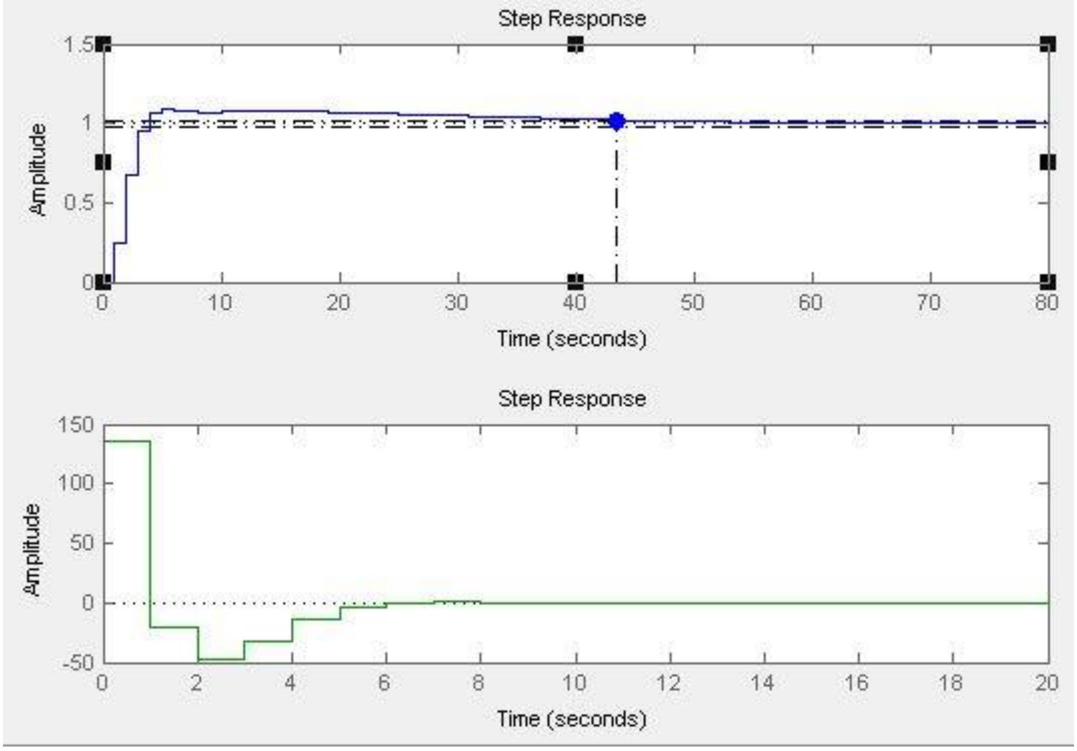
```
Kp = 12.994;
Kd = 123.207;
Ki = 0.207;
```

With the controller, open loop root locus is like below. As you can see, gain margin is high. Second plot show the part near to unit circle. Leaving poles goes to nearest zeros.



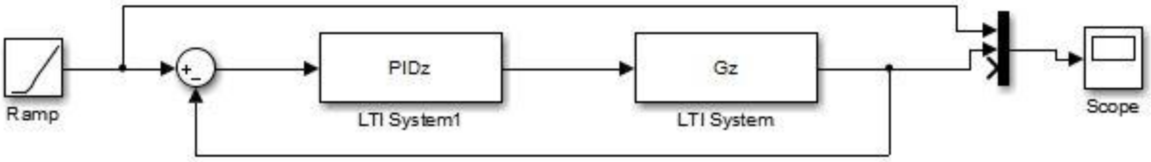


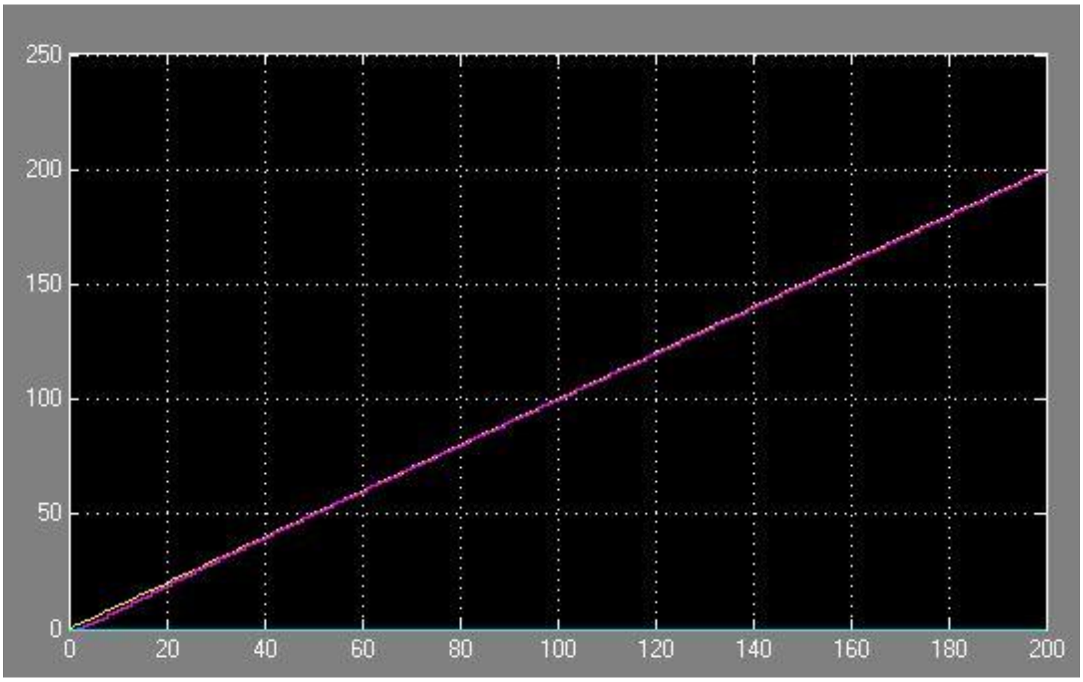
With the designed PID controller, system has 49.3 seconds of settling time and %10 overshoot.



The reason why control signal is so great is that PID's derivative coefficient is so high.

The ramp response has been simulated with simulink blocks as below and we got an error less than 0.75.

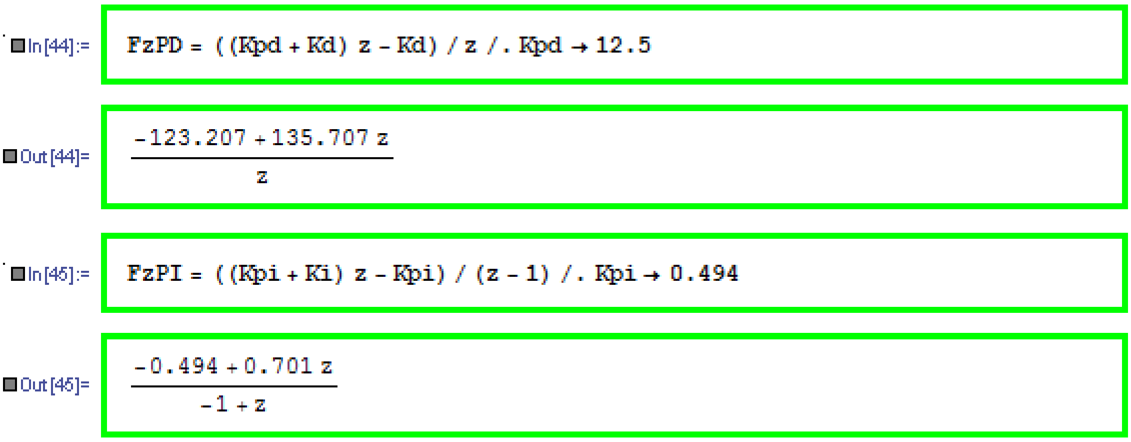




PI-PD

With the designed PID controller, Ǒϩyʃǘǌϩf ăʎǀyǎl·f ǘl·ʅŜă Kʎǀ K Ǿl·fdzŜă l·ǘ ŦʎǌăǘΝ ¢KŜǌŜŦϩǌŜΣ ǘKŜ tLπ t5 Ǒϩyʃǘǌϩff Ŝǌ ǿKʎǑK ʎă l·yϩǘKŜǌ ǘèLJŜ ϩŦ tL5 Ǒϩyʃǘǌϩff Ŝǌ ʎă ŘŜăʎǀyŜŘΦ 5dzǌʎyǀ ŘŜăʎǀyʃϩŦ ǘKŜ tLπt5Σ tL5 Ǒϩyʃǘǌϩf LJl·ǌl·ϒŜǘŜǌă l·ǌŜ dzăŜŘΦ ¦ăʎyǀ ǘKʎă ϒŜǘKϩŘ ōè LJdzǘǘʎyǀ t5 ʎyʃǘKŜ ŦŜŜŘōl·Ǒʅ Σ ǿŜ LJǌŜǾŜyǘŜŘ Ǒϩyʃǘǌϩf ăʎǀyǎl·f ǘϩ Kl·ǾŜ Kʎǀ K Ǿl·fdzŜăΦ

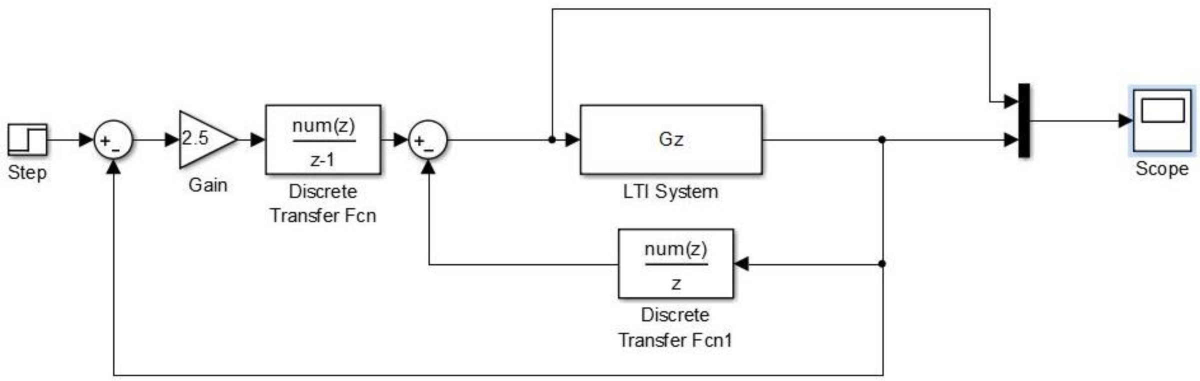
Lyʃ ǘKʎă ǿl·èΣ YLJʎ ʎă l· ŦǌŜŜ LJl·ǌl·ϒŜǘŜǌΦ .è ǘKŜ KŜfLJ ϩŦ Matlab/SIMULINK/SISOTOOLΣ ǿŜ Kl·ǾŜ ăŜŜyʃ YLJʎ ăϩ ăϒl·ff l·yŘ ʎă Ŧϩdzy Ř l·ă ōŜfϩǿΦ

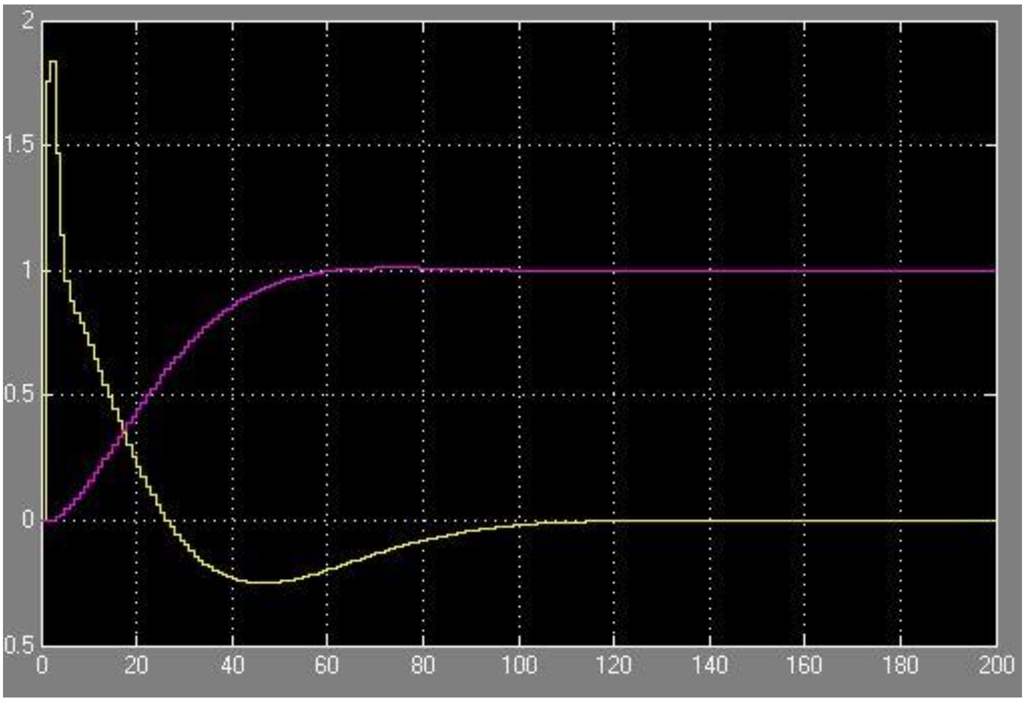


```
In[44]:= FzPD = ((Kpd + Kd) z - Kd) / z /. Kpd → 12.5

Out[44]= -123.207 + 135.707 z
         --------------------
                  z

In[45]:= FzPI = ((Kpi + Ki) z - Kpi) / (z - 1) /. Kpi → 0.494

Out[45]= -0.494 + 0.701 z
         ----------------
             -1 + z
```

Lyʃ ǘKŜ ăʎϒdzfl·ǘʎϩyΣ ǘKŜ ōŜăǘ ǀl·ʎyǾl·fdzŜ ʎă ŘŜǘŜǌϒʎyŜŘ l·ă н.pΦ

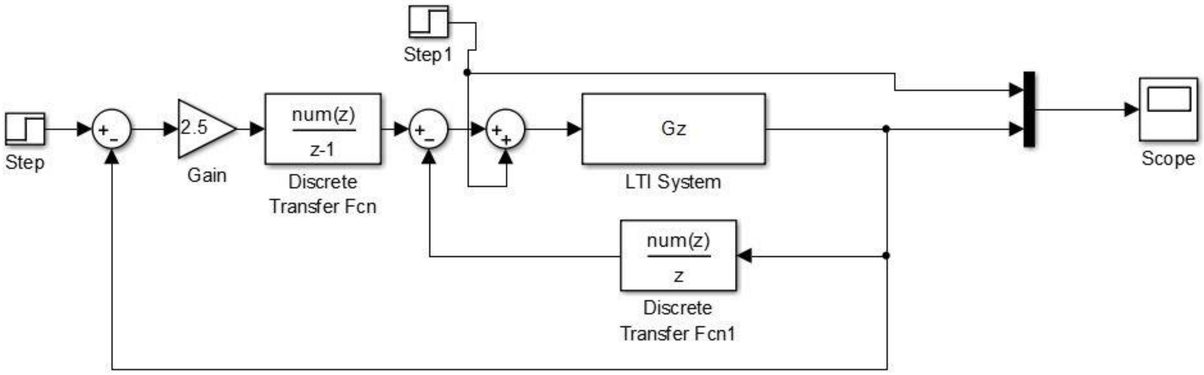
-Firstly, we have looked at the control signal and step response.

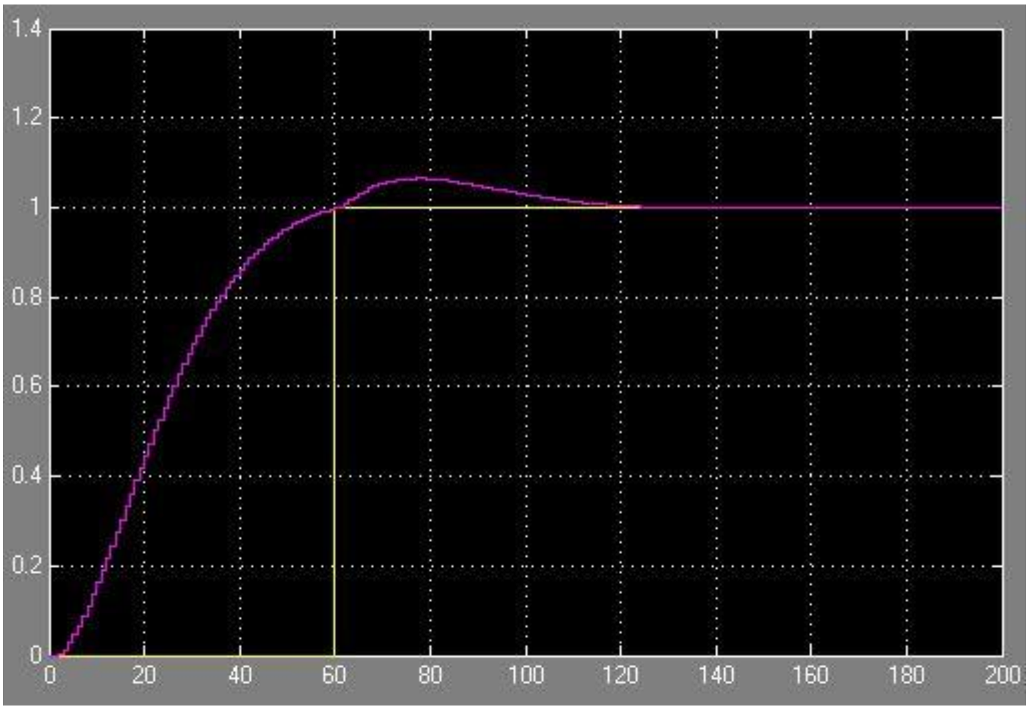


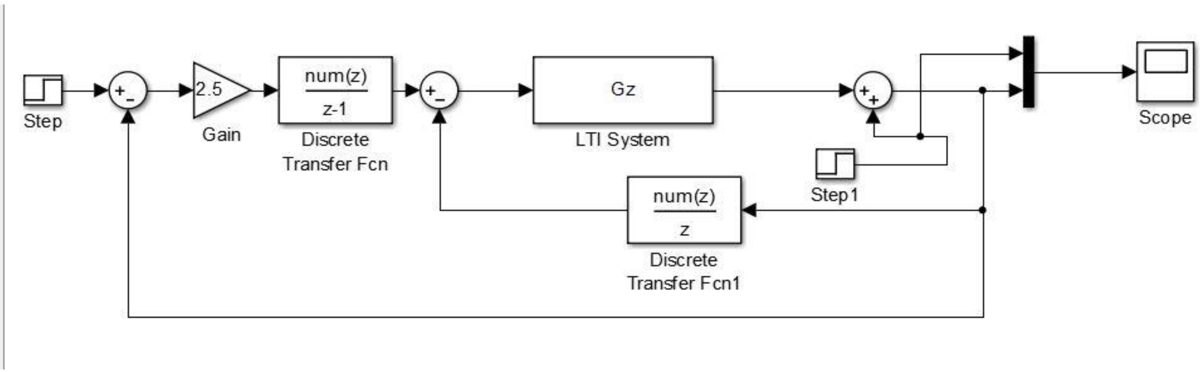


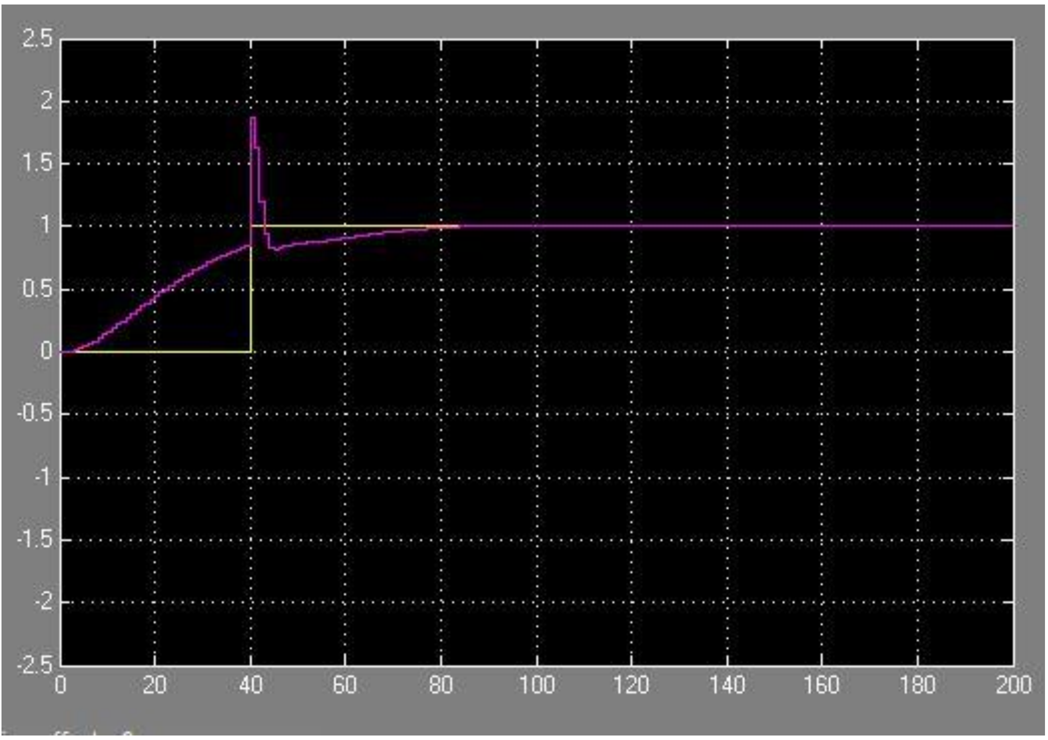
a-4)

Since we have an integrator in the cotroller that we designed, we will succeed to eliminate the I-O disturbances. These disturbances are tested in Simulink.




1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
20
40
60
80
100
120
140
160
180
200

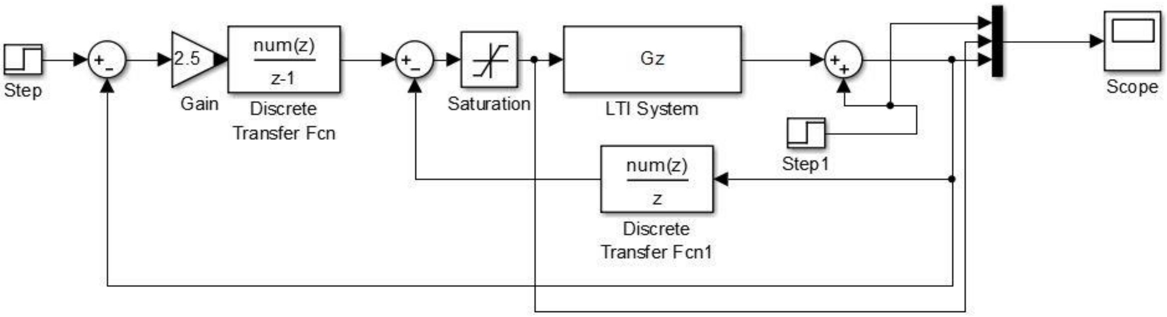

Step
+
-
2.5
Gain
num(z)
z-1
Discrete
Transfer Fcn
+
-
Gz
LTI System
+
+
Step1
num(z)
z
Discrete
Transfer Fcn1
Scope

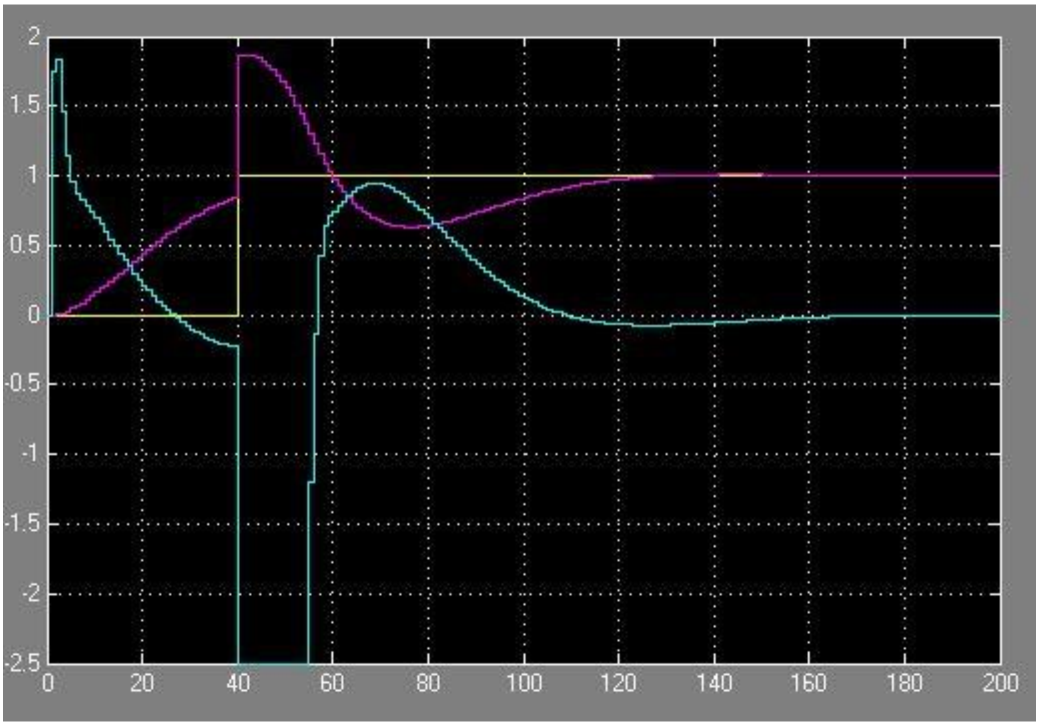

2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
0
20
40
60
80
100
120
140
160
180
200

,

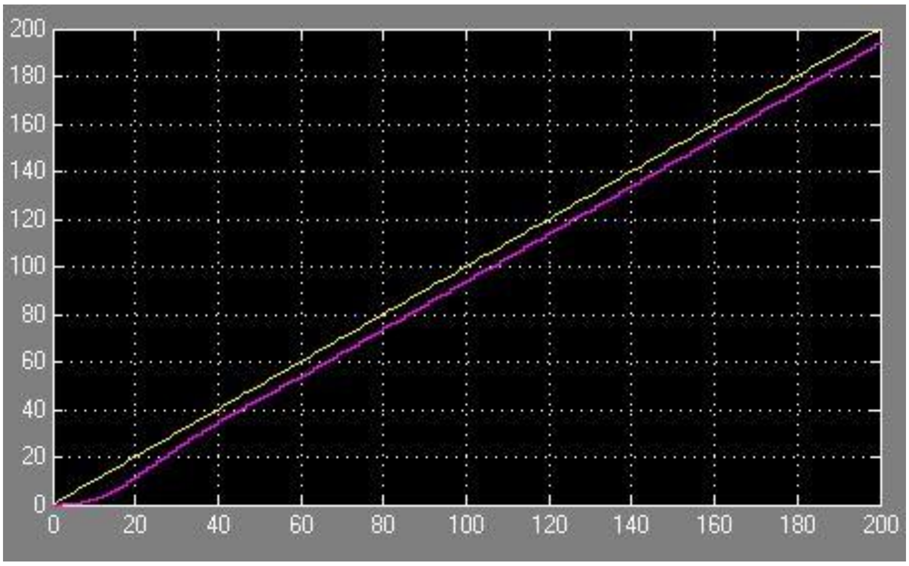
a-5)

We have use system saturation element with Matlab / Simulink to test if it meets all the criteria in the environment.





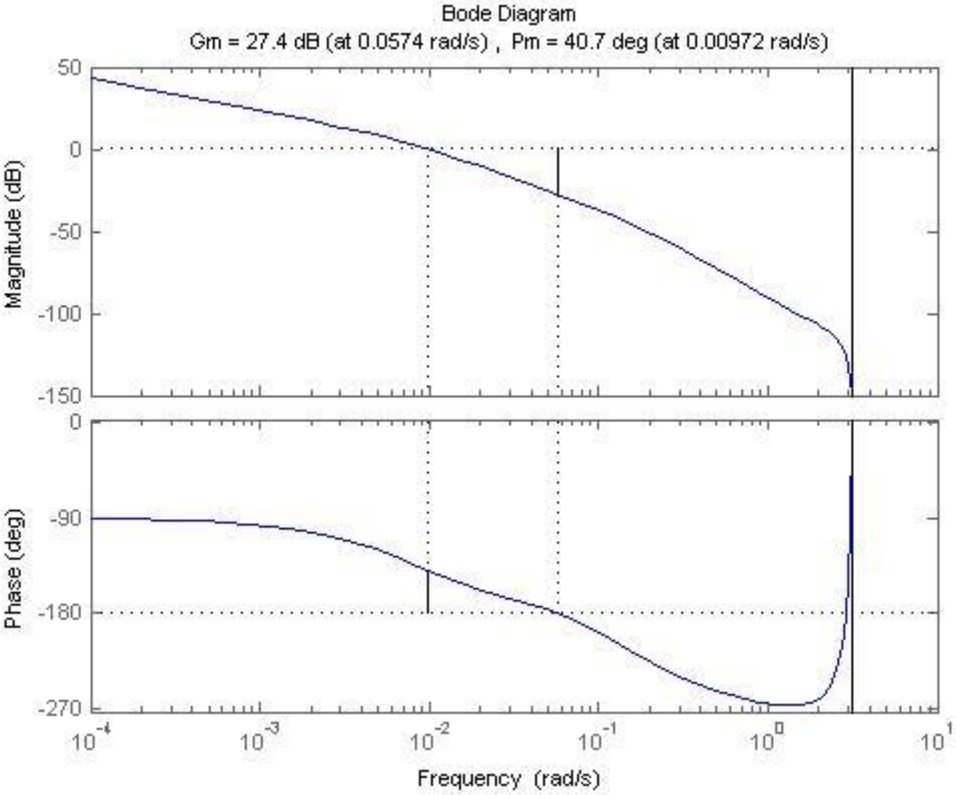
The ramp input response is as following. We observed the error a little more than PID.



A6)

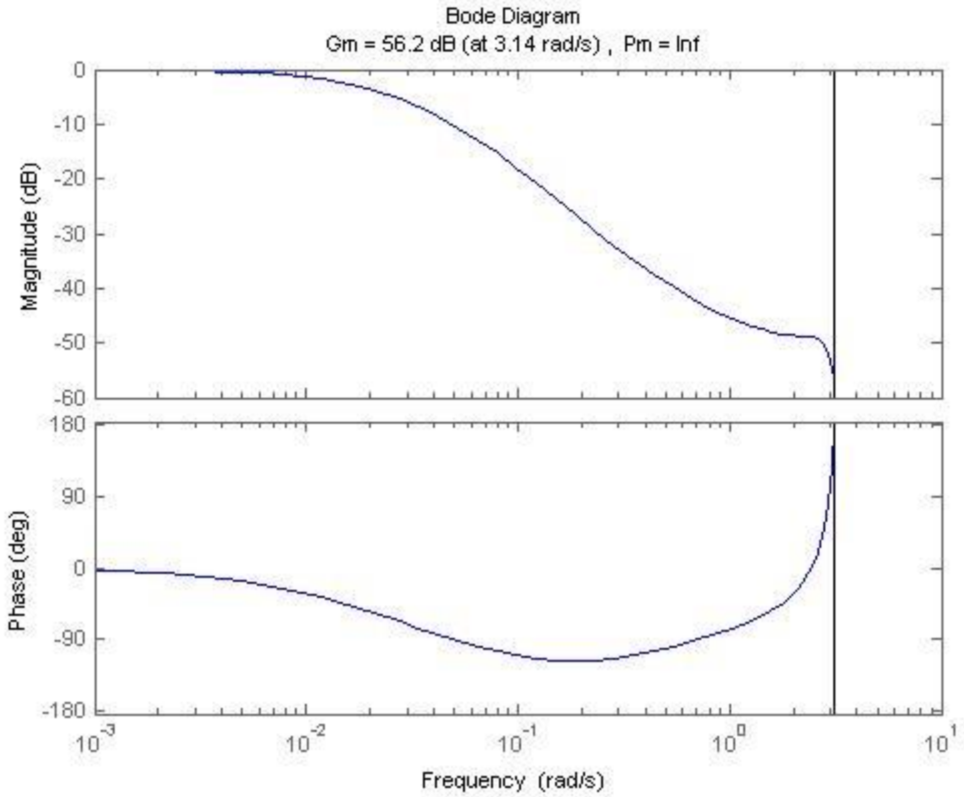
Since the first step of controller design, as seen in the block diagram in the presence of saturating element is made so as to change the system behavior.

B-)



As shown the figure system has 27.4 dB gain margin and 40.7 degree phase margin. So our system is a stable structure.

C-)



```
>> bandwidth(m)

ans =

   0.01847
```

As shown in the figure, resonance peak is 0 because bode curve starts from 0 and falls.

References:

http://www.mathworks.com/help/control/ref/c2d.html

http://www.mathworks.com/help/matlab/examples/displaying-multiple-plots-in-a-single-figure.html

L.Gören, Bilgisayar Kontrollü Sistemler, Ders Notları, 2012

M.T.Söylemez, Kontrol Sistem Tasarımı, Ders Notları, 2012

K. Ogata, *Modern Control Engineering*, 3rded., Prentice-Hall, New Jersey, 1997

Benjamin C. Kuo, *Otomatik Kontrol Sistemleri*, Literatür, İstanbul, 1999