KON326E COMPUTER CONTROLLED SYSTEMS

PROJECT 4

Prof. Dr. Yaprak Yalçın

TEAM 4:

Abdulbaki Aybakan / 040120523

Kanan Muradov / 040120914

Yousef Rahwan / 040140922

Özge Kaya / 040120517

2015-2016

1.

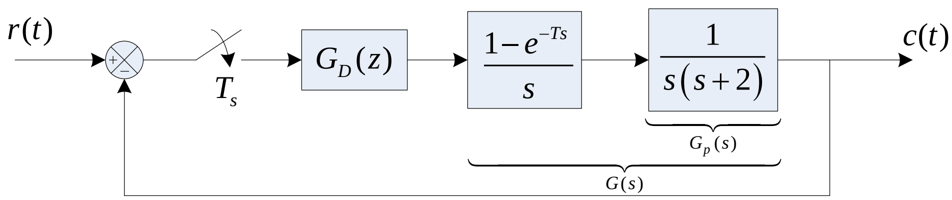


Figure 1.1

For this system, in w-domain a controller with phase margin of 55 degrees and gain margin of 10 dBs is asked to be designed.

a. First of all, in MATLAB, we converted our system to the discreet time system by using zero order hold.

```
>>Gp=tf([1],[1 2 0])

Gp =

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

Continuous-time transfer function.

>>Gz=c2d(Gp,0.1,'zoh')

Gz =

  0.004683 z + 0.004381
  ----------------------
  z^2 - 1.819 z + 0.8187

Sample time: 0.1 seconds
Discrete-time transfer function.
```

Then we transferred it to the w plane by using Mathematica.

In[3]:= $Gz = \frac{0.004683\ (0.9355 + z)}{(z - 1)\ (z - 0.8187)}$

Out[3]= $\frac{0.004683\ (0.9355 + z)}{(-1 + z)\ (-0.8187 + z)}$

In[6]:= $zdesired = \frac{1 + \left(\frac{T}{2}\right) w}{1 - \left(\frac{T}{2}\right) w}$ /. T → 0.1

Out[6]= $\frac{1 + 0.05\, w}{1 - 0.05\, w}$

In[7]:= Gw = Together[Gz /. z → zdesired]

Out[7]= $\frac{0.04683\ (1. - 0.05\, w)\ (1.9355 + 0.003225\, w)}{(0.1813 + 0.090935\, w)\ w}$

We designed our controller after transferring our system to the w domain. Because our system was a type 1 system and we were asked to keep the velocity error at the certain value, we had to design a type 0 controller.

In[8]:= $Gdw = Kc * \frac{1 + \tau\, w}{1 + \alpha\, \tau\, w}$;

We found our Kc value by using the velocity error value given.

In[9]:= Kv1 = Limit[w Gw Gdw, w → 0]

Out[9]= 0.499942 Kc

In[12]:= Kv = 5;

In[13]:= solKc = Solve[Kv == Kv1]

Out[13]= {{Kc → 10.0012}}

For the sake of simplicity, we will take Kc as 10 from this point on. After we found the gain, we multiply the system with that value and draw the Bode diagram.

We plot the bode diagram of Gz multiplied with Kc on MATLAB.

>> margin(Gz*10.0012)

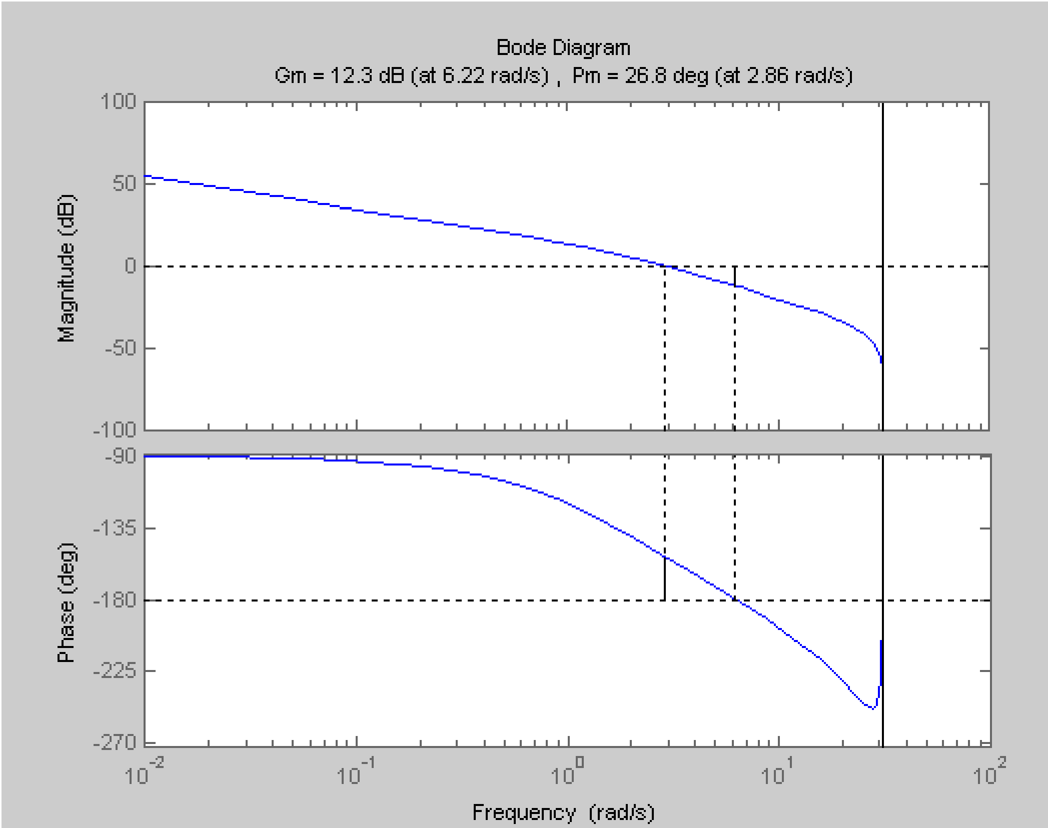


Figure 1.2: Bode Diagram of the system

We can also see from the diagram that out controller was in fact a phase leading controller. In the question, we are told to design a controller with the phase margin of 55 degrees. We can calculate the maximum phase added to the system by the controller by subtracting these two values. However, the controller doesn't only change the phase of the system but also the magnitude of the system. In order to minimize the error caused by this, we need to increase the phase of the system by 8-10 degrees.

```
In[100]:= PM = 55;
          ϕm = PM - 26.8 + 10

Out[101]= 38.2
```

When we took 38.2 as our phase, the system did not reach the phase margin that we wanted. For this reason, we repeated the calculations once more by increasing the phase 15 degrees instead of 10. In addition, this way overshoot will be decreased.

After obtaining the maximum phase margin, we calculated the controller parameters.

In[102]:= PM = 55;
ϕm = PM - 26.8 + 15

Out[103]= 43.2

In[94]:= $\alpha = \dfrac{1 - \mathrm{Sin}\left[\phi m * \frac{\pi}{180}\right]}{1 + \mathrm{Sin}\left[\phi m * \frac{\pi}{180}\right]}$

Out[94]= 0.187263

In[95]:= $\mathrm{wdm} = -20 * \mathrm{Log}\left[10, \dfrac{1}{\sqrt{\alpha}}\right]$

Out[95]= -7.27549

Under normal circumstances, if the take the w value for when the system has the magnitude of -7.27 dBs, the magnitude of the system will be zero and phase margin can be calculated by using this frequency.

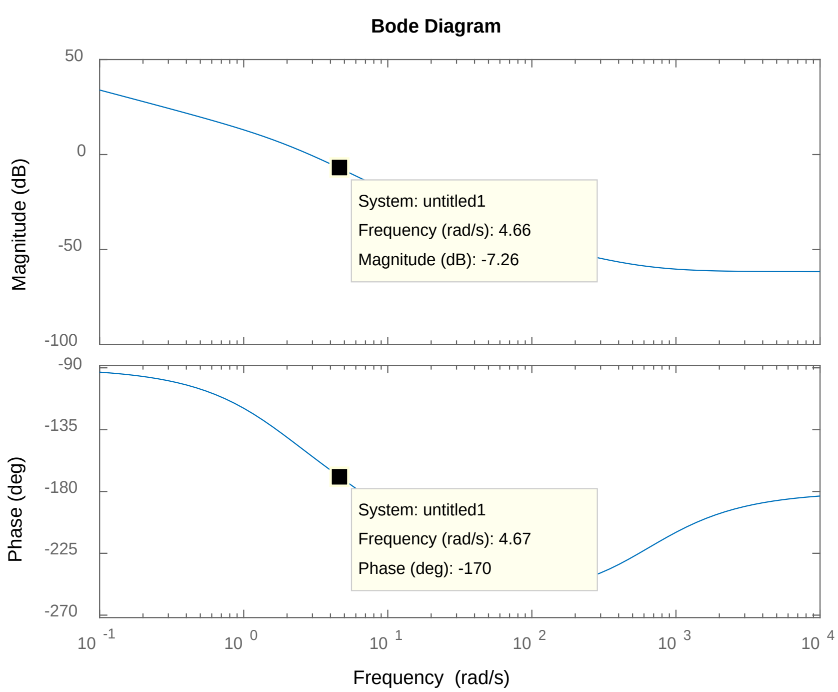


Figure 1.1: Bode Diagram of System

As seen here, w_{max} should be chosen as 4.67 rad/sn.

In[96]:= $\tau = \frac{1}{4.67\sqrt{\alpha}}$

Out[96]= 0.494832

Now that we know all the parameters for out controller, we can design it.

In[97]:= AchievedGdw = Gdw /. Kc → 10

Out[97]= $\frac{10\ (1 + 0.494832\ w)}{1 + 0.0926635\ w}$

Then we go back to z domain from w domain.

In[66]:= wdesired = $\frac{2}{T} * \frac{z-1}{z+1}$ /. T → 0.1

Out[66]= $\frac{20.\ (-1 + z)}{1 + z}$

In[98]:= AchievedGdz = Together[AchievedGdw /. w → wdesired]

Out[98]= $\frac{10.\ (-8.89663 + 10.8966\ z)}{-0.85327 + 2.85327\ z}$

After designing the controller, we obtained the system response for step and ramp input. In order to do that, we first created the Simulink blocks for the step input and then graphed the system response.

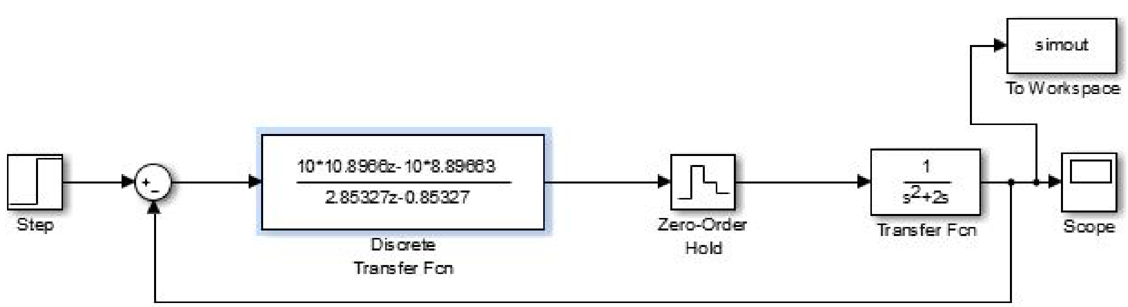


Figure 1.2: Simulink model for step response

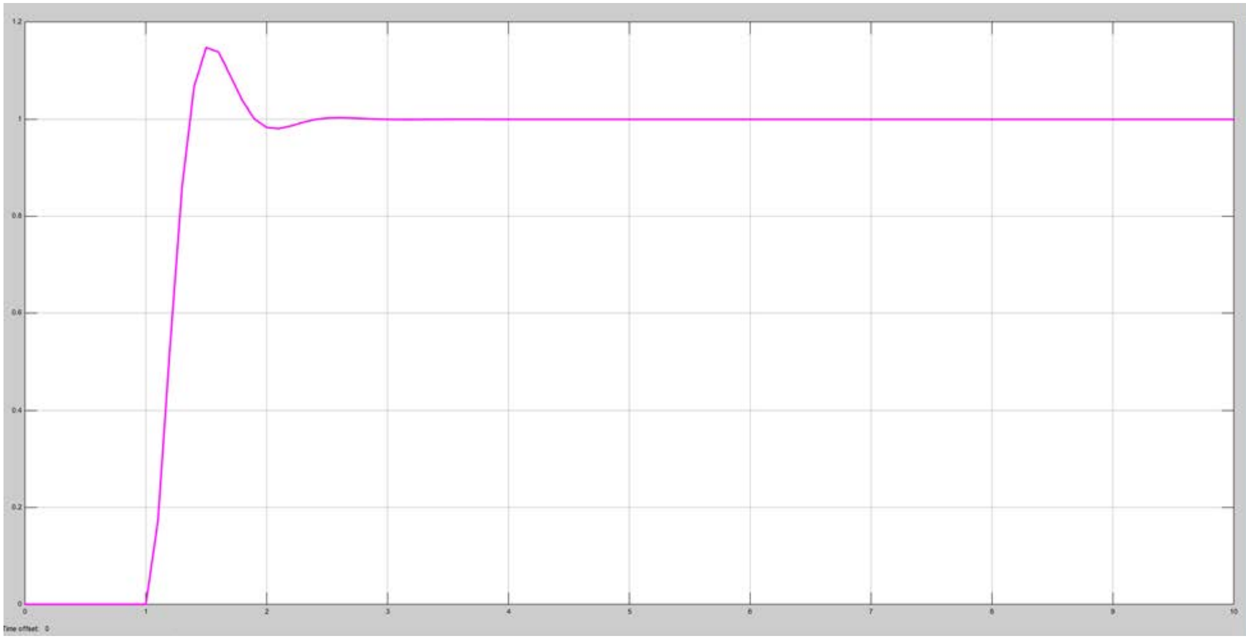


Figure 1.3:Step response of the system

We put the necessary information in the Simulink program to obtain the step response overshoot. By using the max(simout) command, we found the peak value of the step response and overshoot as %14.

```
>> max(simout)

ans =

    1.1421
```

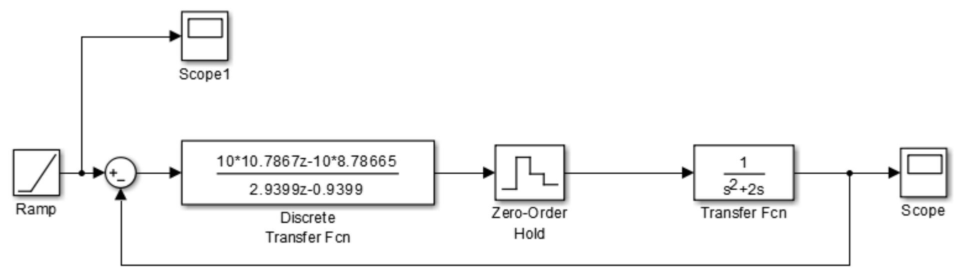


Figure 1. 4: Simulink model for ramp response

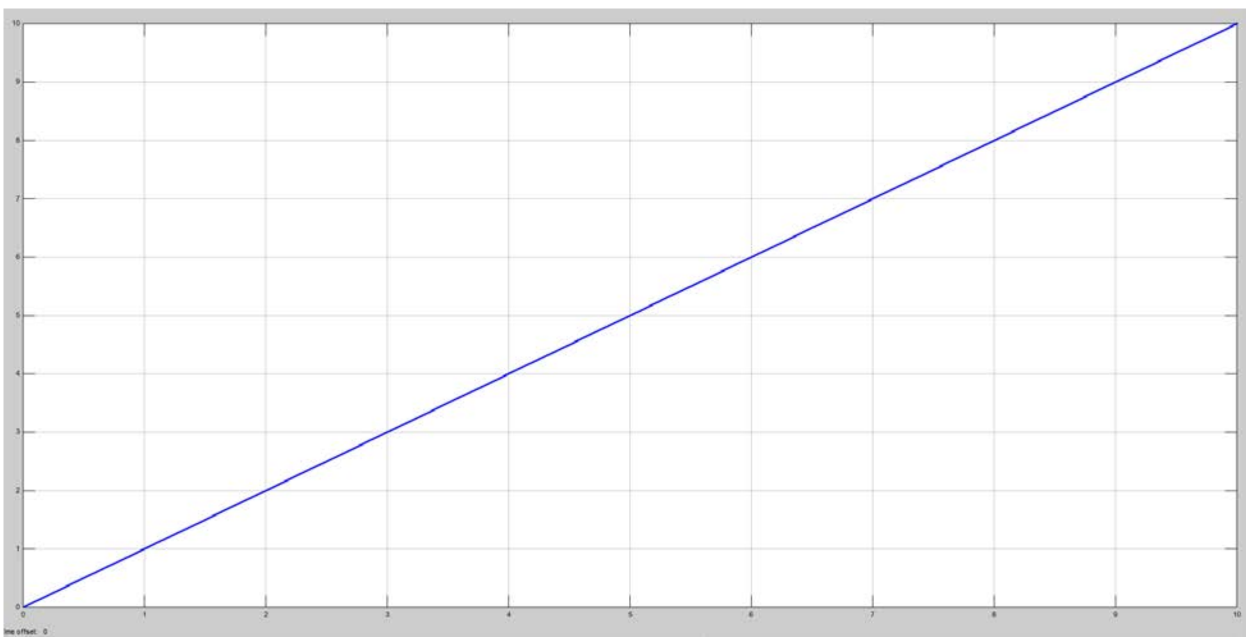


Figure 1.5: Ramp Input

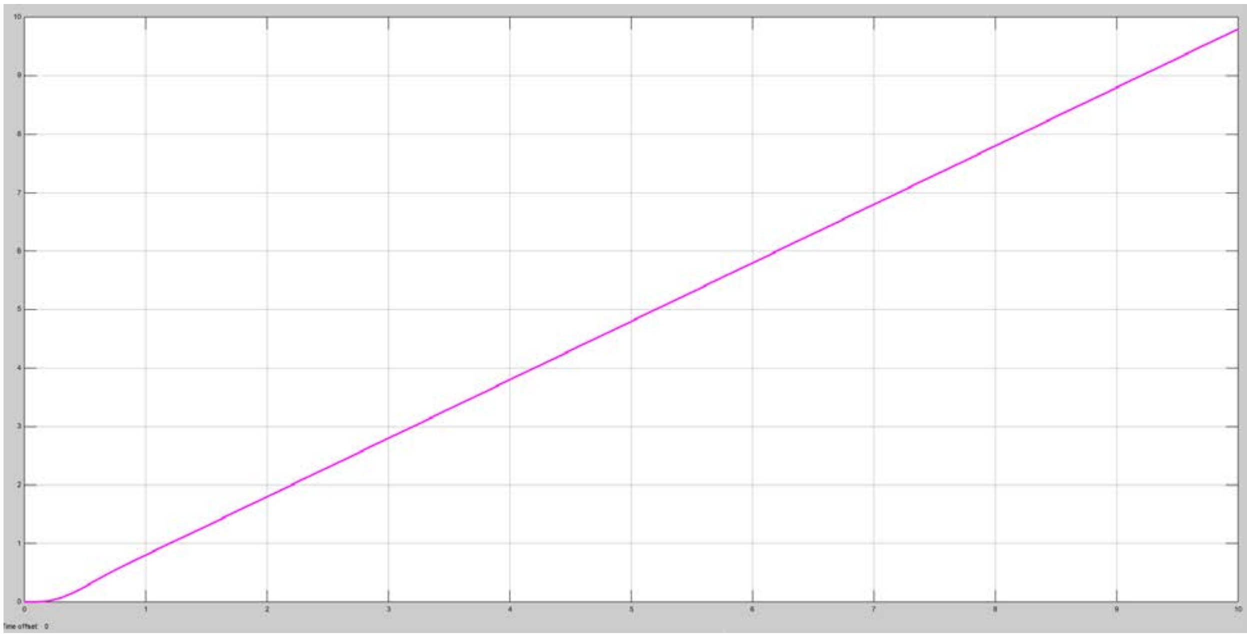


Figure 1.6: System Response

We found the steady state error for ramp input.

In[49]:= $\mathbf{ess} = \dfrac{\mathbf{1}}{\mathbf{Kv}}$

Out[49]= $\dfrac{1}{5}$

As we can see in the graphs, the system response is below the reference value.

Finally, we can talk about the system that we established. In image 9, we can see the Bode diagram of the system. The phase margin is roughly 61 degrees which is larger than 55 degrees. Phase margin and gain margin give us an idea about the system's stability. The larger these values are the more stable system is. Because our system has phase margin of 61 degrees and that is a higher value than what we were asked to achieve, our system is safe.

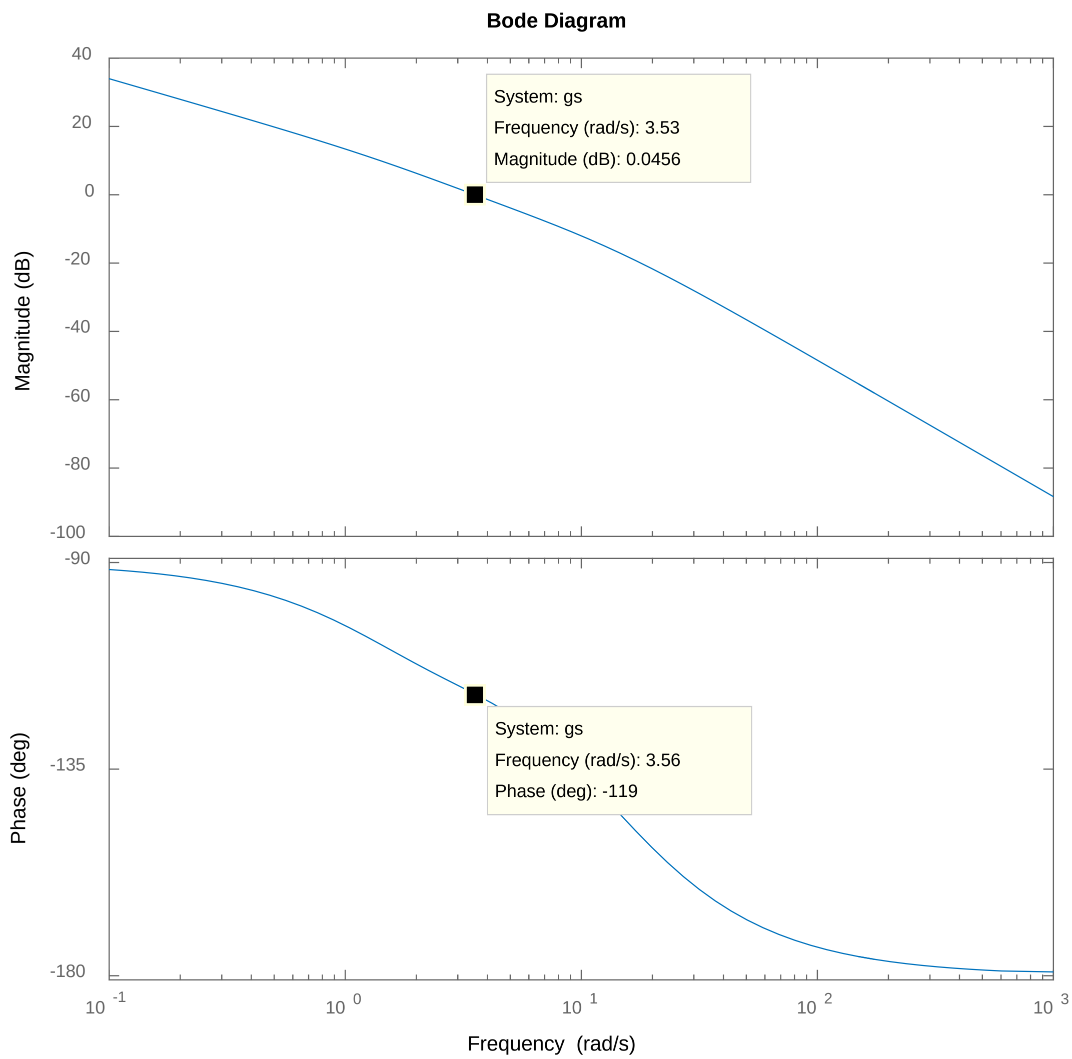


Figure 1.7: Bode Diagram of System

b.

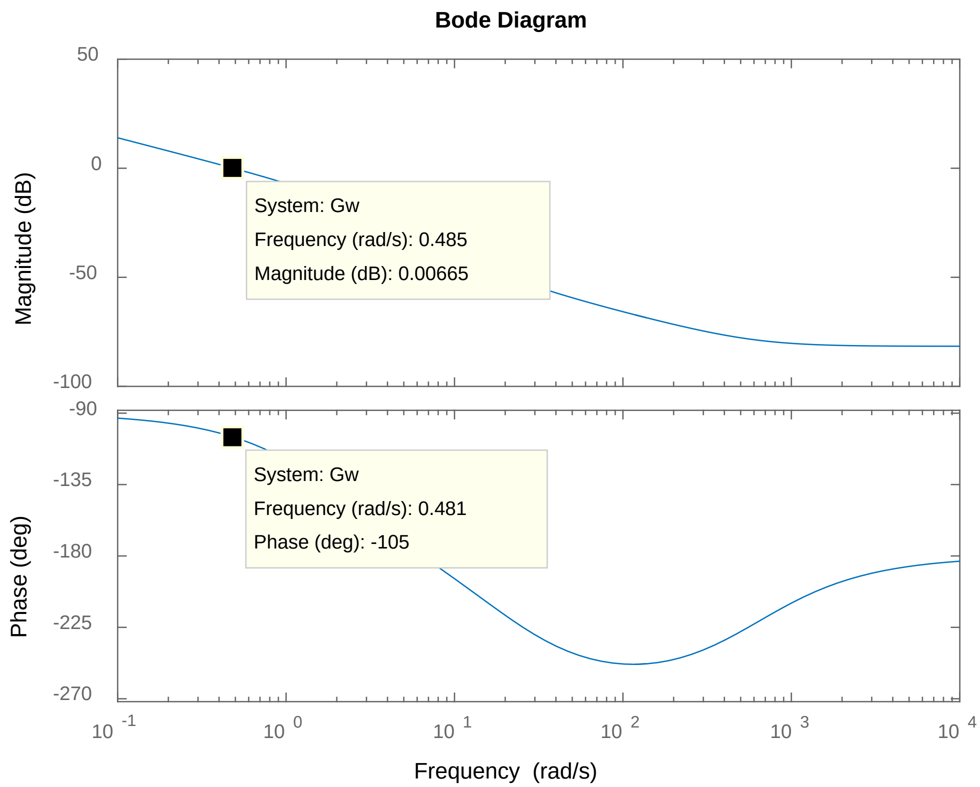


Figure 1.8: Bode Diagram of G (w)

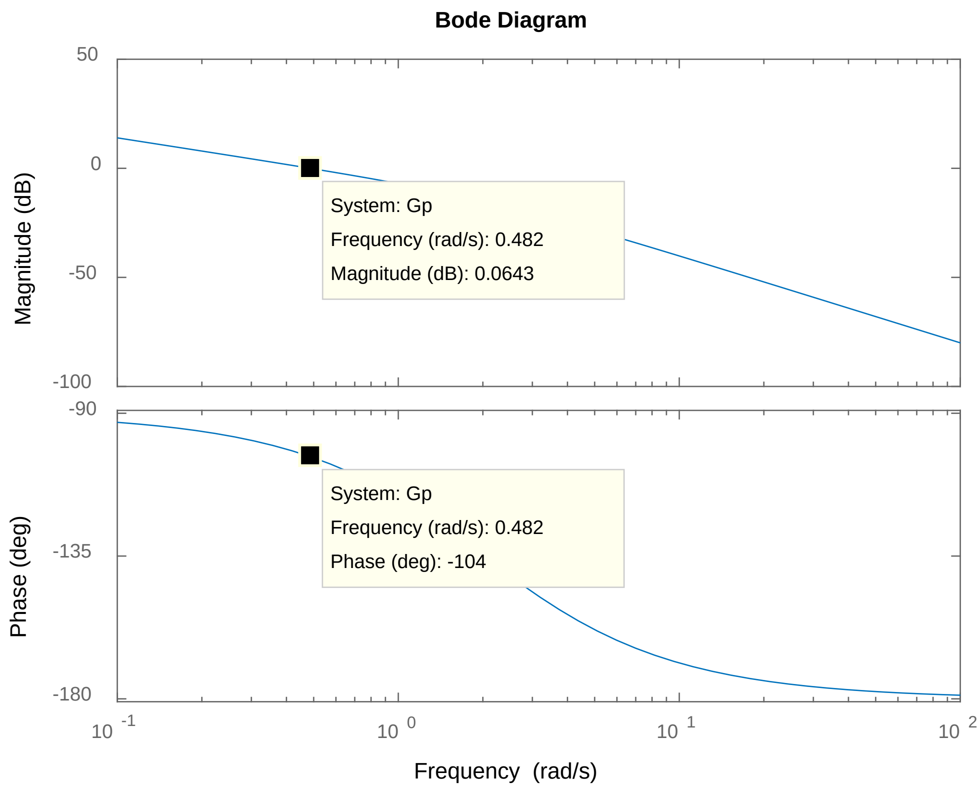


Figure 1.9: Bode Diagram of Gp(s)

In practice, s domain and w domain are very similar in a limited sense. Thus, as we can understand from the Bode diagram, in low frequencies the difference between s domain and w domain is fairly small but then it increases with the increasing frequency. This difference is set by the information that gets lost during the sampling operation. Both systems have similar phase margins but because in none of the systems the phase reaches 180 degrees for a finite w value, we cannot obtain a certain gain margin for either one of the systems.

c. Our control structure and the system are as follows;

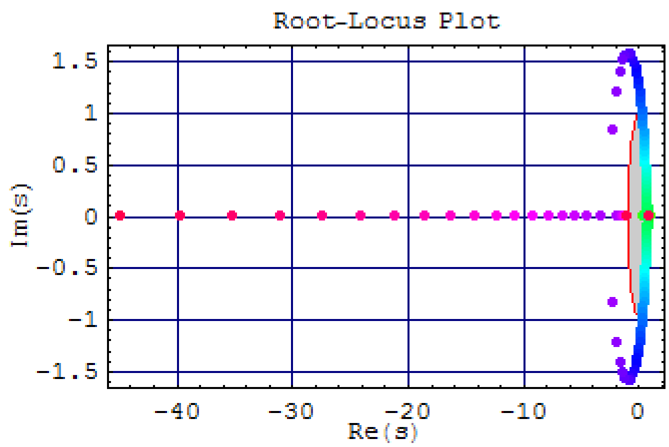
In[3]:= $Gz = \dfrac{0.004683\ (0.9355 + z)}{(z - 1)\ (z - 0.8187)}$

Out[3]= $\dfrac{0.004683\ (0.9355 + z)}{(-1 + z)\ (-0.8187 + z)}$

In[98]:= AchievedGdz = Together[AchievedGdw /. w → wdesired]

Out[98]= $\dfrac{10.\ (-8.89663 + 10.8966\ z)}{-0.85327 + 2.85327\ z}$

In[28]:= RootLoci[GzGdz]



Out[28]= - Graphics -

We investigated the Root Locus diagram by using both Mathematica and MATLAB. Closed loop poles can be either in stable or unstable region depending on the K value. Smaller the K value is the more stable the system gets. As the K value gets larger and larger the poles move toward the unstable region.

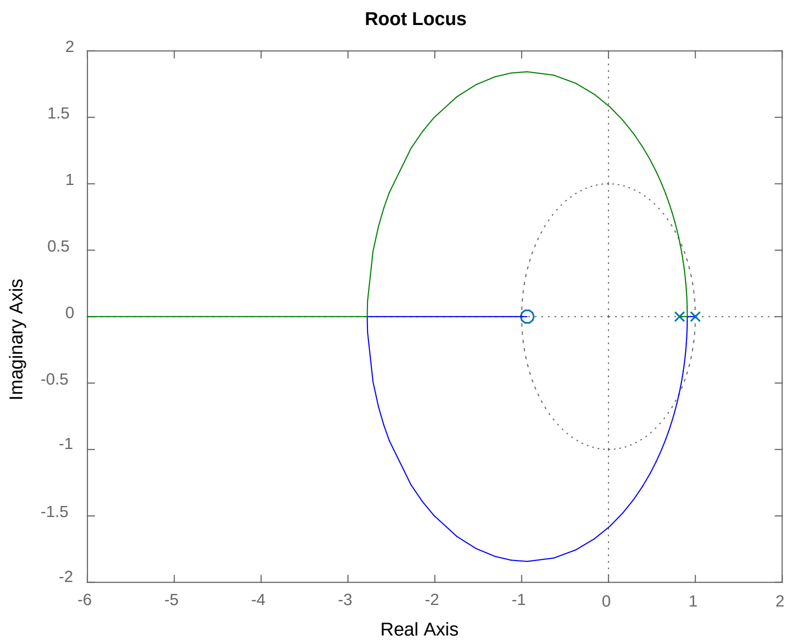


Figure 1.10: Root locus of G (z)

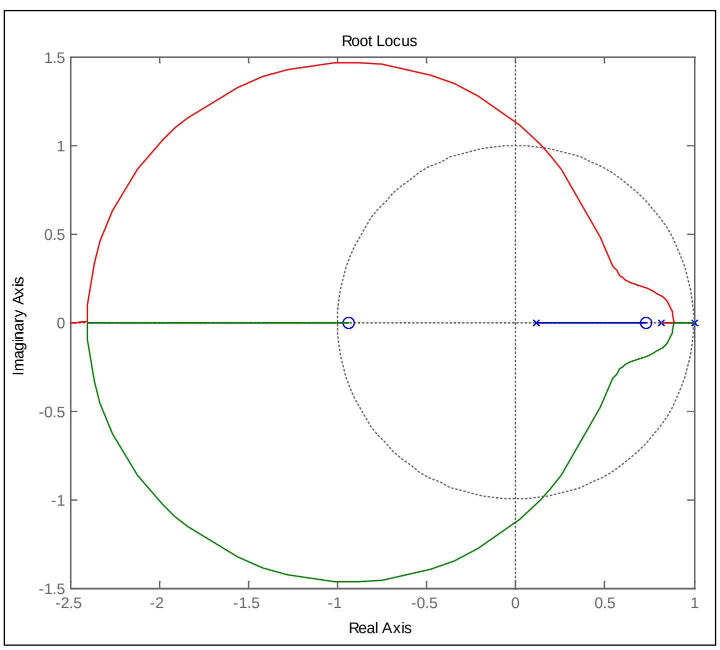


Figure 1.11: Root locus of G (z)$G_d(z)$*

Let's investigate the poles and zeros of the systems $G(z) * G_d(z)$ and G(z);

```
In[37]:= Solve[Numerator[Gz] == 0, z]

Out[37]= {{z → -0.9355}}

In[33]:= Solve[Denominator[Gz] == 0, z]

Out[33]= {{z → 0.8187}, {z → 1.}}

In[35]:= Solve[Denominator[Gz AchievedGdz] == 0, z]

Out[35]= {{z → 0.350554}, {z → 0.8187}, {z → 1.}}

In[36]:= Solve[Numerator[Gz AchievedGdz] == 0, z]

Out[36]= {{z → -0.9355}, {z → 0.796329}}
```

Inside unit circle is where the system is stable. By adding the zeros and poles to the system with the controller, the system is made more stable and the overshoot and settling time were decreased.

d.

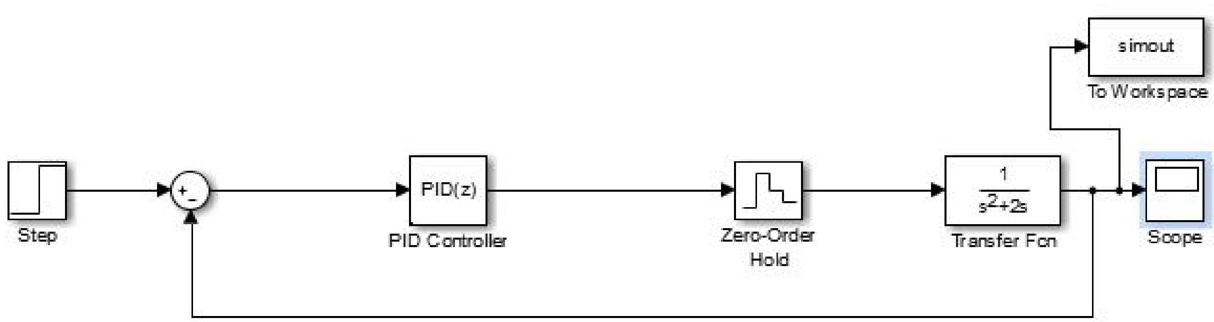


Figure 1.12: Simulink model for step response

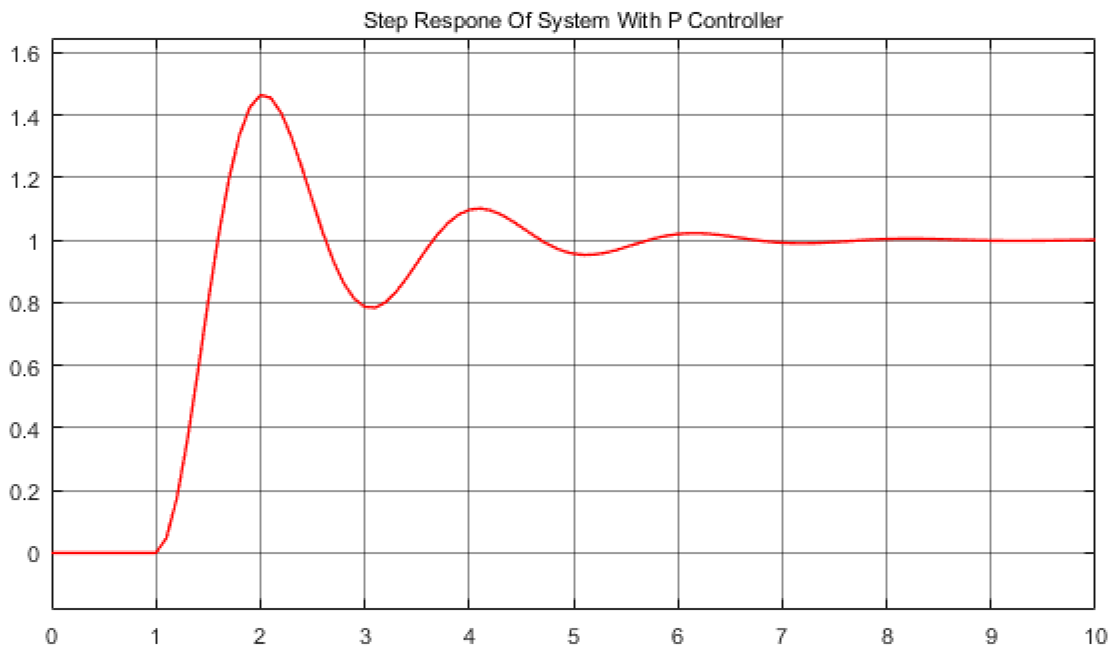
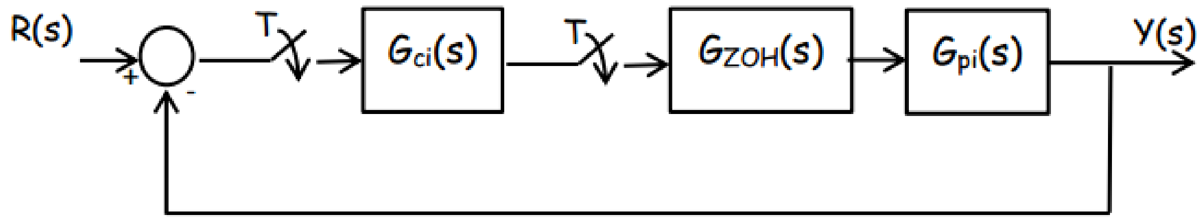


Figure 1.15: Step Response of System with P Controller

The unit step response of the most updated system is in the figure above. As can be seen from the figure, the overshoot and the settling time of the system is higher compared to the system graph designated in a. In addition there was some oscillation before the system is finally settled. Therefore, controller K did not satisfy the set parameters.

2. System responses has been found for 5 different sampling periods with Tustin transform.



Since we are team 4,

$$G_{pi}(s) = \frac{5}{s+8} \qquad G_{ci}(s) = \frac{35.5(s+8.5)}{s(s+19)}$$

First, we find the pmax as the biggest magnitude of the poles of the continuous closed loop system.

In[3]:=

```
Gpi = 5/(s + 8);
Gci = 35.5 (s + 8.5)/(s (s + 19));
Ts = Together[Gpi Gci/(1 + Gpi Gci)]
```

$$\frac{1508.75 + 177.5\,s}{1508.75 + 329.5\,s + 27.\,s^2 + s^3}$$

In[6]:=

```
PolesSISO[Ts]
```

{-9.01445, -8.99277 - 9.30055 i,
-8.99277 + 9.30055 i}

```
In[7]:= pmax = Sqrt[8.99^2 + 9.3^2]

12.9348
```

We found the pmax= 12.9348

T1	$\frac{2}{P_{max}}$	$\cong 154\ ms$
T2	$\frac{1}{P_{max}}$	$\cong 77\ ms$
T3	$\frac{1}{2\ P_{max}}$	$\cong 38.65\ ms$
T4	$\frac{1}{5\ P_{max}}$	$\cong 15.4\ ms$
T5	$\frac{1}{10\ P_{max}}$	$\cong 7.7\ ms$

Figure 2.1: Table of Sampling Periods

When we apply Tustin transform on MATLAB.

```
>> Gci=35.5*(s + 8.5)/(s*(s + 19));

>> Gc1z=c2d(Gci,0.154,'tustin')

Gc1z =

  1.836 z^2 + 1.453 z - 0.3834

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

    z^2 - 0.812 z - 0.188

 Sample time: 0.154 seconds

>> Gc2z=c2d(Gci,0.077,'tustin')

Gc2z =

 1.048 z^2 + 0.5166 z - 0.531

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

    z^2 - 1.155 z + 0.1551

Sample time: 0.077 seconds
```

Discrete-time transfer function.

```
>> Gc3z=c2d(Gci,0.03865,'tustin')

Gc3z =

  0.5842 z^2 + 0.1649 z - 0.4194
  ------------------------------
      z^2 - 1.463 z + 0.4629

 Sample time: 0.03865 seconds
```

Discrete-time transfer function.

```
>> Gc4z=c2d(Gci,0.0154,'tustin')

Gc4z =

  0.2541 z^2 + 0.03121 z - 0.2229
  -------------------------------
      z^2 - 1.745 z + 0.7447

 Sample time: 0.0154 seconds
```

Discrete-time transfer function.

```
>> Gc5z=c2d(Gci,0.0077,'tustin')

Gc5z =

  0.1315 z^2 + 0.008336 z - 0.1232
  --------------------------------
       z^2 - 1.864 z + 0.8637

 Sample time: 0.0077 seconds
```

Discrete-time transfer function.

When we apply these to the Simulink;

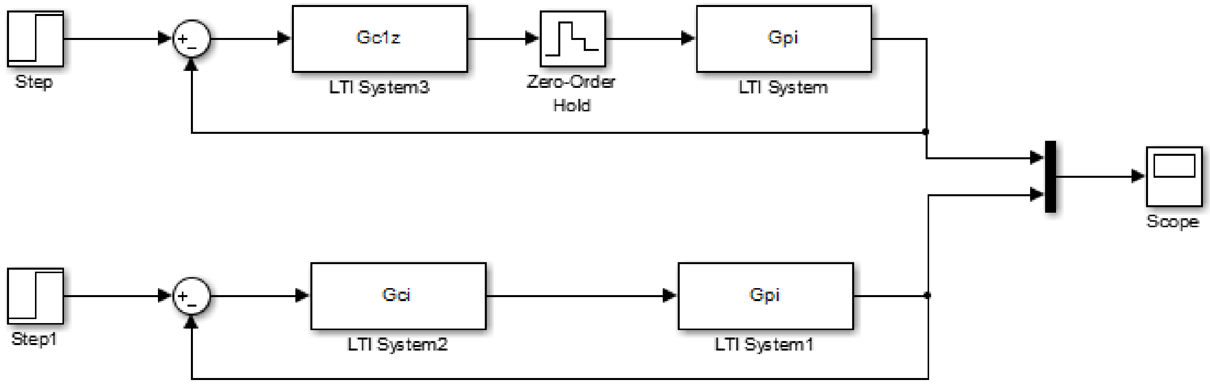


Figure 2.2: Simulink Blocks

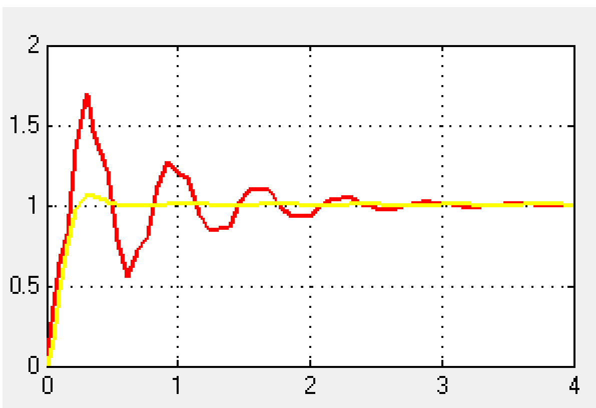


Figure 2.3: Step Responses at T=0.154 seconds (red->sampled yellow->continuous)

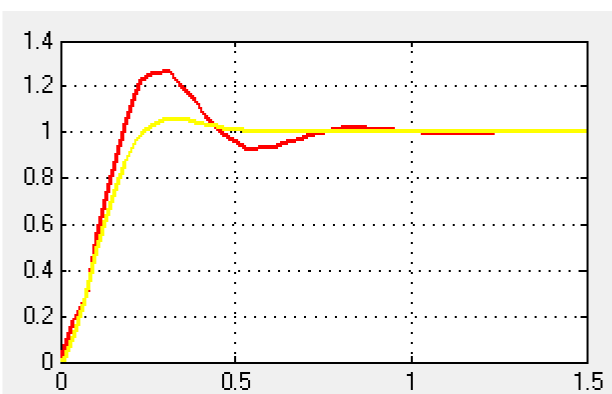


Figure 2.4: Step Responses at T=0.077 seconds (red->sampled yellow->continuous)

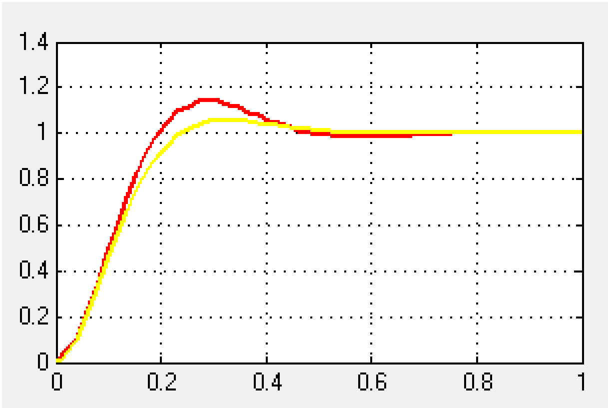


Figure 2.5: Step Responses at T=0.03865 seconds (red->sampled yellow->continuous)

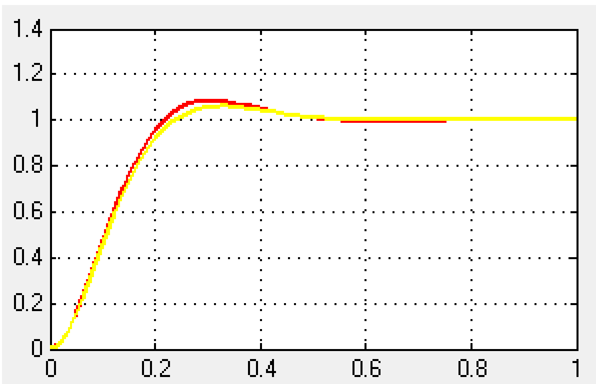


Figure 2.6: Step Responses at T=0.0154 seconds (red->sampled yellow->continuous)

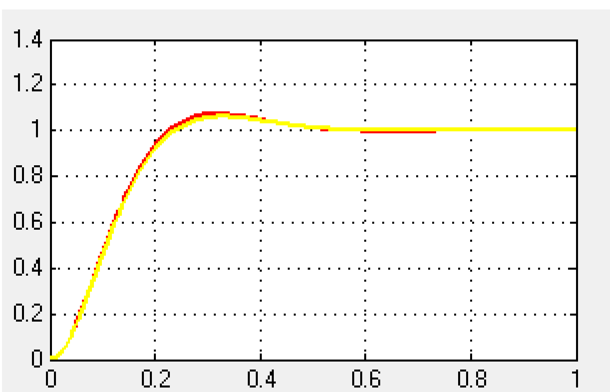


Figure 2.7: Step Responses at T=0.0077 seconds (red->sampled yellow->continuous)

As we can see, when we had less sampling periods, sampled system response came closer to the continuous one.