Computer Aided Control Systems Analysis and Design Using MATLAB

Introduction

The design of the continuous domain transfer function can be achieved in multiple ways to achieve design criteria. In this report, the main method used is frequency response method. From the given transfer function, which is critical damped in the s domain.
The unity feedback system has a plant with the transfer function given by:

$$G(s) = \frac{684}{s(s^2+12s+57)}$$

The design of compensators should satisfy requirements given below:
i. Percentage overshoot < 15%.
ii. Rise-time < 100 msec.
iii. Settling time < 500 msec.
iv. Zero steady-state error to a step.

To get the 15% overshoot, the following equations will be used during the calculation, the maximum overshoot can be obtained by setting $Omax$ to get ζ by using Equation 1, and then work out the phase margin using the equation 2, and use the bode diagram to find out the gain crossover frequency for that phase margin. plug in the ω inside the equation to work out the K. And the K is required proportional gain compensator.

$$O_{\max} = e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}}$$

Equation 1: Overshoot/safety factor of plant's damping ratio

$$\phi_M \cong \tan^{-1}\left\{\frac{2\zeta}{(\sqrt{1+4\zeta^4}-2\zeta^2)^{0.5}}\right\}$$

Equation 2: Phase Margin of current plant by damping ratio

In practical, the $Omax$ **is considered a bit lower than the requirement, in this case, it was set to 10% overshoot, which is 0.10 in equivalent of** $Omax$**,**

In this lab practice, to fit the purpose of computer aid design, the function was made by using Matlab. The function is shown below

```
function [ r ] = dampingratio( po )
%OVERSHOOT transfer the desired overshoot to damping ratio
r=sqrt((log(po/100)^2)/(pi^2+(log(po/100)^2)))
%   r is the damping ratio, and po is the overshoot input.
return
```

```
end
```

and to use this function, the overshoot of 10% was chosen for design purpose,

```
damp=over2damp(10)
```

This shall return required damping ratio required, which is 0.5912,

For the design purpose the Ts (settling time) was chosen at 0.2

$$\omega_B = \frac{4}{T_s \zeta} \sqrt{(1 - 2\zeta^2) + \sqrt{4\zeta^4 - 4\zeta^2 + 2}}$$

Equation 3

TASK A

This design is intend to add lead-lag compensator to increase phase-margin in order to obtain desired rise speed and overshoot.

The script used in the design process is shown below with detailed explanation

```
G=tf(684,[1 12 57 0])
po=10;
[dp]=dampingratio(po)
figure(1)
bode(G)
[Gm,Pm,Wcg,Wcp] = margin(G);
```

These code is to generate the original plant and initial plot of bode diagram to find plant phase margin and gain.

To find a lead compensator, the process starts from looking for required phase margin. The ideal phase margin is 60degree in general, therefore the required phase margin is 60 due to the plant's phase margin is0.

```
Pm=60-Pm
```

The alpha value was chosen from table below

Phase-Lead Table

α	0.05	0.1	0.2	0.24	0.3	0.5	1
G dB	13	10	7	6	5	3	0
φ_m (deg.)	65	55	42	38	33	19	0

Table 1

```
alpha=1/0.05
```

From the table, 0.05 is the value for this phase margin and because of different equation was used in later calculation, the alpha is the inverse of alpha on the table.

$$20\log|G(j\omega)| = 20\log\frac{1}{\sqrt{\alpha}}$$

Equation 4

The gain of future plant can be derived from equation 4, this gives the clue of magnitude needed.

```
GM=1/sqrt(alpha)
```

From the bode diagram, magnitude of the obtained frequency is straightforward, however, to satisfy the computer participation, a method was developed to squeeze out the value of the bode and match them with known variables. The phase and frequency can be matched from our known magnitude.

```
[MAG,PHASE,W] = bode(G)
phase= interp1( squeeze(MAG), squeeze(PHASE), GM)
w= interp1( squeeze(MAG), W, GM)
```

$$Z = \frac{w}{\sqrt{\alpha}}$$
$$P = \alpha \times Z$$

Equation 5

Now, the lead compensator can be written down according to equation 5

```
Z=w/sqrt(alpha)
P=alpha*Z
```

And because the magnitude shift, another alpha is the coefficient for this lead compensator

```
Klead=tf([1 Z],[1 P])*alpha
GKlead=G*Klead
figure(2)
GKleadc=feedback(G*Klead,1);
step(GKleadc)
[Gm2,Pm2,Wcg2,Wcp2] = margin(GKlead)
figure(4)
bode(GKlead)
```

From the plot above, the performance of this lead compensator does not meet the requirement and the phase margin is still way below the desired value, hence another lead compensator should be considered. In order to extract the peak values for value check, following equations are used to check peak values, and they are not a part of main body.

```
GKleadcs=step(GKleadc)
figure(3)
findpeaks(GKleadcs)
PKS=findpeaks(GKleadcs);
```

This method of finding peaks can store the peak value into variable, which is a good way to monitor the system performance.

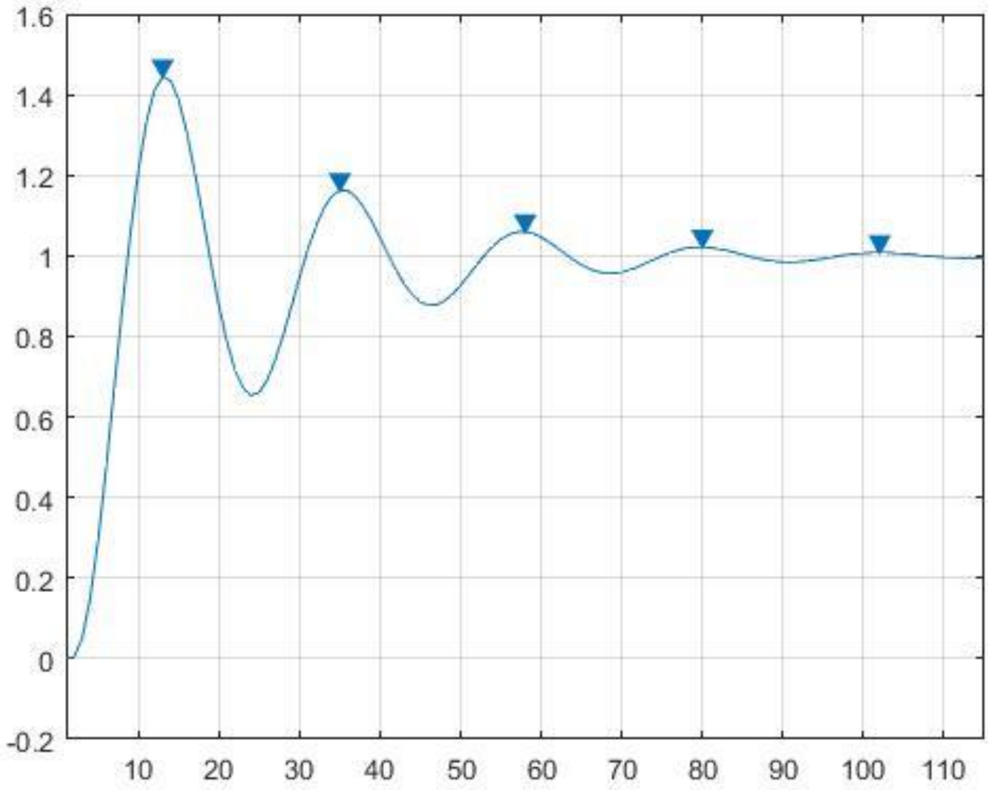

Figure 1

This is the design of second compensator to increase phase margin and other specification

```
Pm=60-Pm2;
alpha2=3.33
[MAG,PHASE,W] = bode(GKlead)
GM2=(1/sqrt(alpha2))
phase2= interp1( squeeze(MAG), squeeze(PHASE), GM2)
w2= interp1( squeeze(MAG), W, GM2)
Z2=w2/sqrt(alpha2)
P2=alpha2*Z2
Klead2=tf([1 Z2],[1 P2])*alpha2
G2Klead=GKlead*Klead2;
% % % % % % % % % % % %
G2Kleadc=feedback(G2Klead,1)
figure(5)
step(G2Kleadc)
figure(6)
bode(G2Klead)
figure(7)
G2KleadK=GKlead*Klead2*(1/1.28);
G2KleadKc=feedback(G2KleadK,1);
step(G2KleadKc)
```

Table 2: Closed Loop feedback performance of Compensated Plant

Specification	GKlead1Klead2K
Percentage Overshoot	12
Settling Time	0.4867s
Rise Time	0.1145s

This shows the good response of step input with 2 Klead and a proportional compensator

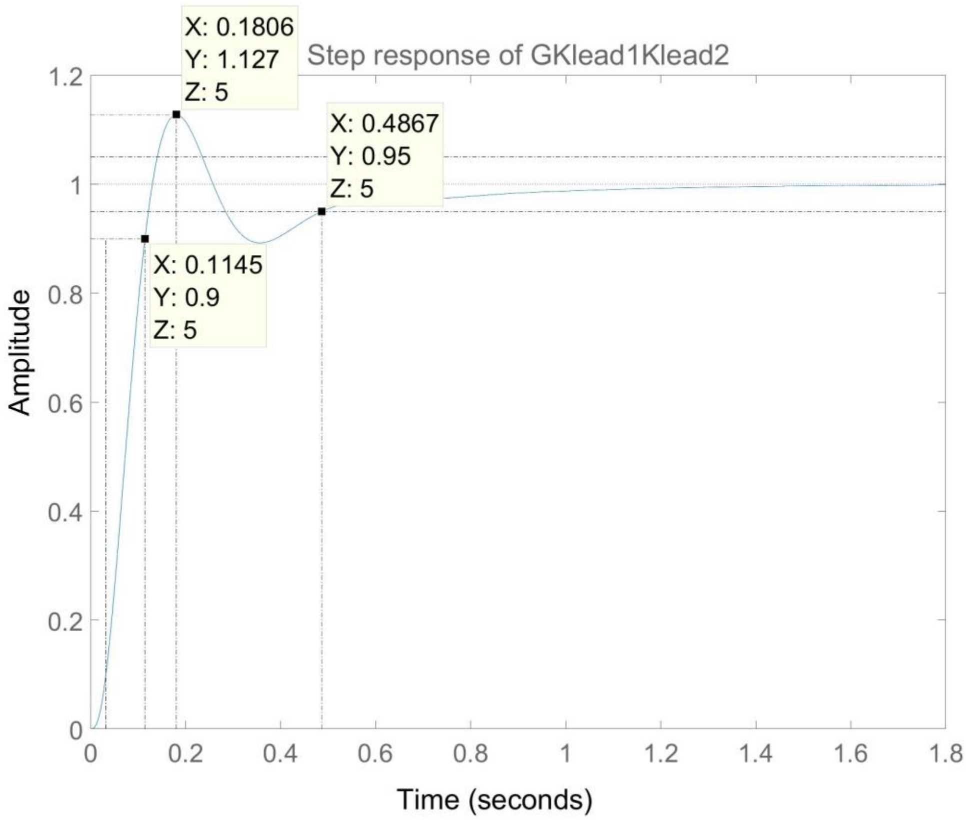

Figure 2

Task B

And by Shannon -Nyquist theory, the sampling frequency is 20 times bandwidth frequency,

$$\omega_s = 20\omega_B$$

The sampling frequency should be neither too big or too small, due to the compromise of performance and interference. A big sampling time could result lack of detail which leads to lack of accuracy, and the small value will result interference with noise of the system, which also leads to losses of information

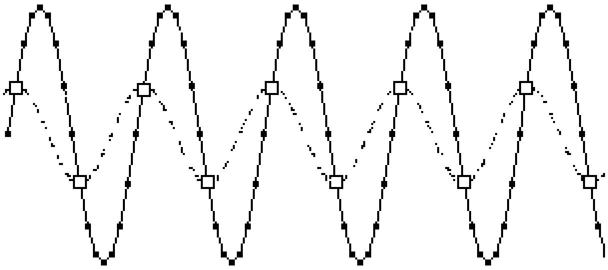

Figure 3 the sampling frequency and time difference

(http://www.daqarta.com/alias1.PNG,2016)

The Shannon-Nyquist theorem provides a reasonable sampling frequency and time.

By given damping ratio and settling time, bandwidth can be obtained from equation3 and sampling frequency as well. Therefore, sampling time can be calculated by using following equation.

```
Ts=(2*pi)/Ws
```

To have good result of the discreet system, reasonable accurate parameters are required, hence the number format is changed to long to have more digits in order to improve accuracy.

```
format long
```

This command changes single to double-precision accuracy.

Compensator design should be similar to Task A, however, due to discrete system approximation, the response will be very different in general or dramatically changes at critical point, therefore, a new intermedium domain cross continuous and discreet domain is necessary

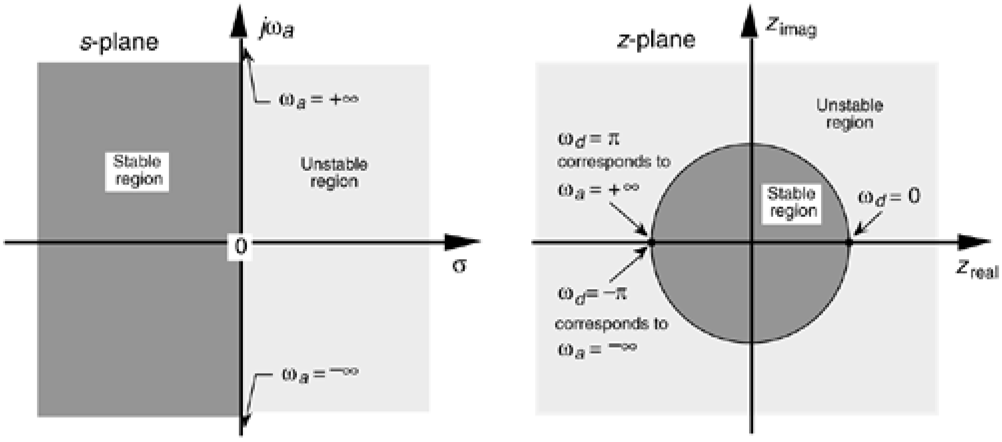

Figure 4 Difference of stability region of s-plane and z-plane (http://flylib.com/books/2/729/1/html/2/images/0131089897/graphics/06fig31.gif,2016)

To obtain a suitable compensator for z-domain, a intermedium domain is introduced called w-domain. This can be achieved by using a self-made function Gz2Gw

```
function [ Gw ] = Gz2Gw( Gz,Ts )
%UNTITLED Summary of this function goes here
%   Detailed explanation goes here
denz=Gz.den{1,1}
numz=Gz.num{1,1}
dent=denz.*[-1 1 -1 1]
numt=numz.*[-1 1 -1 1]
[numx denx]=bilinear(numt,dent,0.5)
% Gx=tf(numx,denx)
xpara=(Ts/2)*-1
numw=[numx].*[xpara^3 xpara^2 xpara 1]
denw=[denx].*[xpara^3 xpara^2 xpara 1]
Gw=tf(numw,denw)

End
```

Below is this self-made function explanation

```
% denz=Gz.den{1,1}
% numz=Gz.num{1,1}
% dent=denz.*[-1 1 -1 1]
% numt=numz.*[-1 1 -1 1]
% [numx denx]=bilinear(numt,dent,0.5)
% Gx=tf(numx,denx)
% xpara=(Ts/2)*-1
% numw=[numx].*[xpara^3 xpara^2 xpara 1]
% denw=[denx].*[xpara^3 xpara^2 xpara 1]
% Gw=tf(numw,denw)
```

It is designed for this plant only but can be quickly transfer into a new plant with minimum modification

This is initial setup of original plant, which is same procedure as section3(Task A), by using code below

```
G=tf(684,[1 12 57 0])
Gc=feedback(G,1)
```

```
damp=over2damp(10)
T=0.2
Wb=(4/(T*damp))* sqrt((1-2*damp^2 )+sqrt(4*damp^4-4*damp^2+2))
Ws=20*Wb
Ts=(2*pi)/Ws
format long
Gz=c2d(G,Ts,'zoh')
Gzc=feedback(Gz,1)
figure(1)
step(Gzc)
hold on
step(Gc)
```

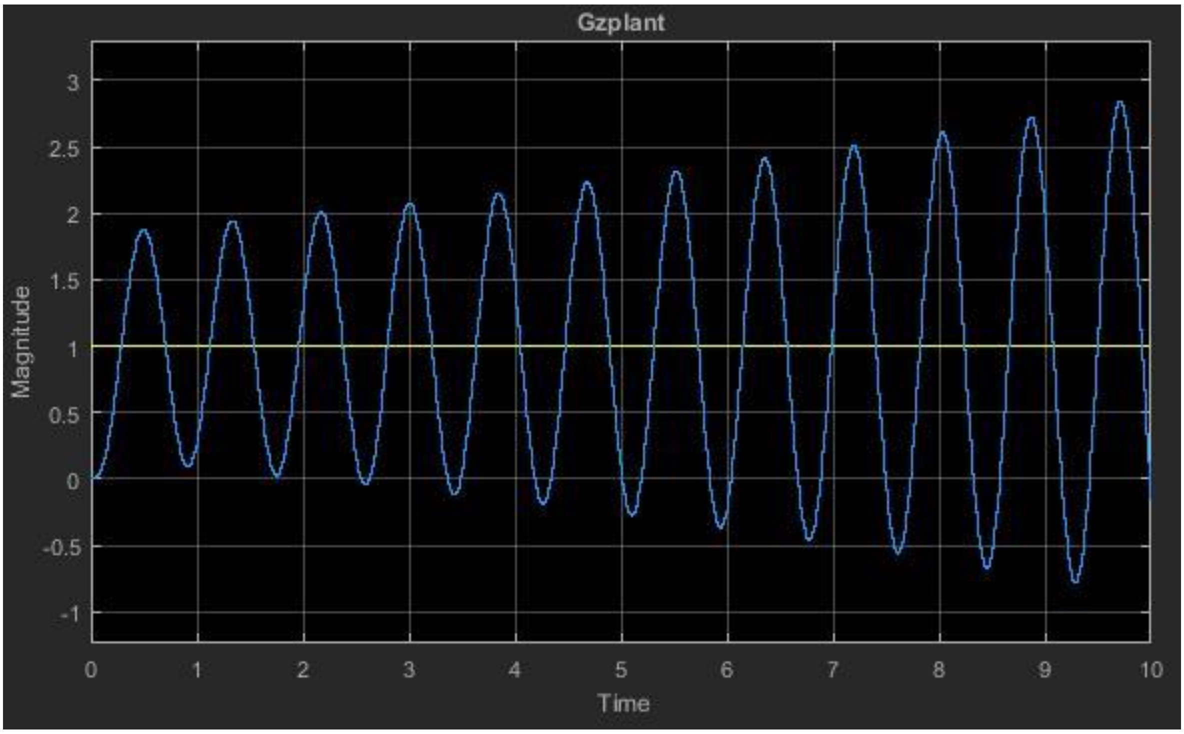

Figure 5

The design of compensator in w-domain is similar to s-domain, however, due to the error caused by sampling, the plant is not exactly same as before, and from figure 5, Gz plant shows divergent behaviour compared to original G plant

```
Gw=Gz2Gw(Gz,Ts)
Gwc=feedback(Gw,1)
figure(2)
bode(Gw)
```

The bode diagram is used to identify the system and make sure the system's error is in considerable range, and from figure above, it is a good match of original system. After the w-domain transfer function is calculated, the same way of finding lead and lag can be applied to w-domain

```
figure(3);
step(Gwc);
```

This figure shows that uncompensated G in w domain, and process below calculates the lead compensator

```
[Gm,Pm,Wcg,Wcp] = margin(Gw);
Pm=(60-Pm)+0.1*(60-Pm)
alpha=1/0.05
GM=1/sqrt(alpha)
[MAG,PHASE,W] = bode(Gw)
phase= interp1( squeeze(MAG), squeeze(PHASE), GM);
w= interp1( squeeze(MAG), W, GM);
Z=w/sqrt(alpha)
P=alpha*Z
Klead=tf([1 Z],[1 P])*alpha
GwKlead=Gw*Klead
figure(2)
GwKleadc=feedback(Gw*Klead,1);
step(GwKleadc)
GwKleadcs=step(GwKleadc);
figure();
findpeaks(GwKleadcs);
PKS=findpeaks(GwKleadcs);
```

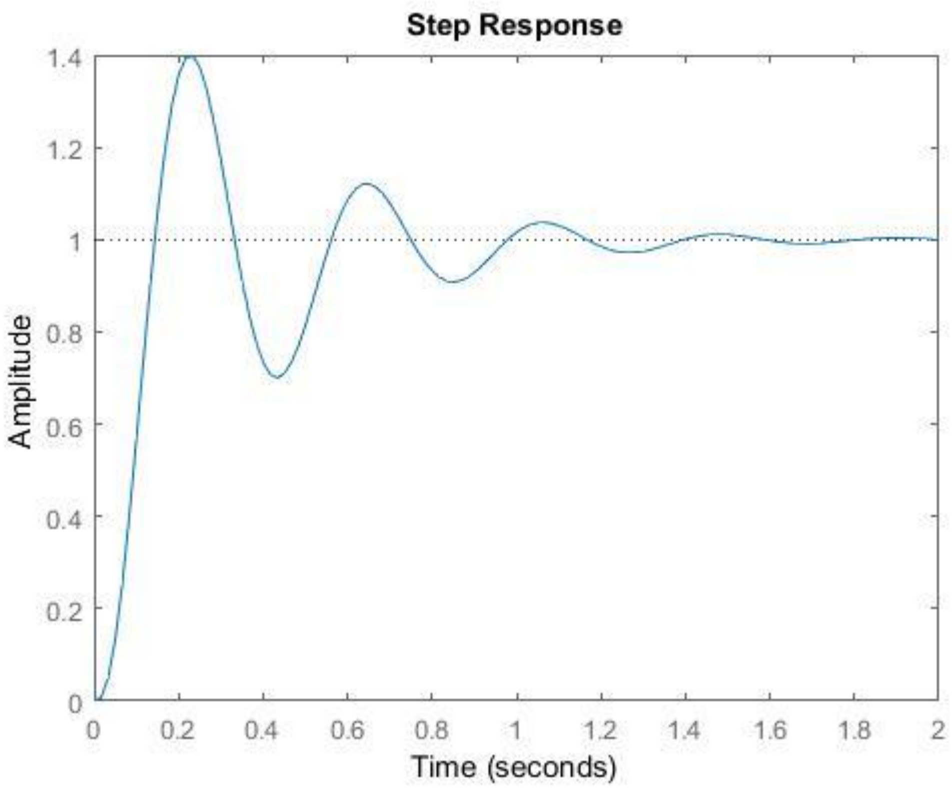

Figure 6: G*Klead close loop feedback

However, from the response above does not meet design criteria hence another lead compensator is required to increase both settling time and phase margin

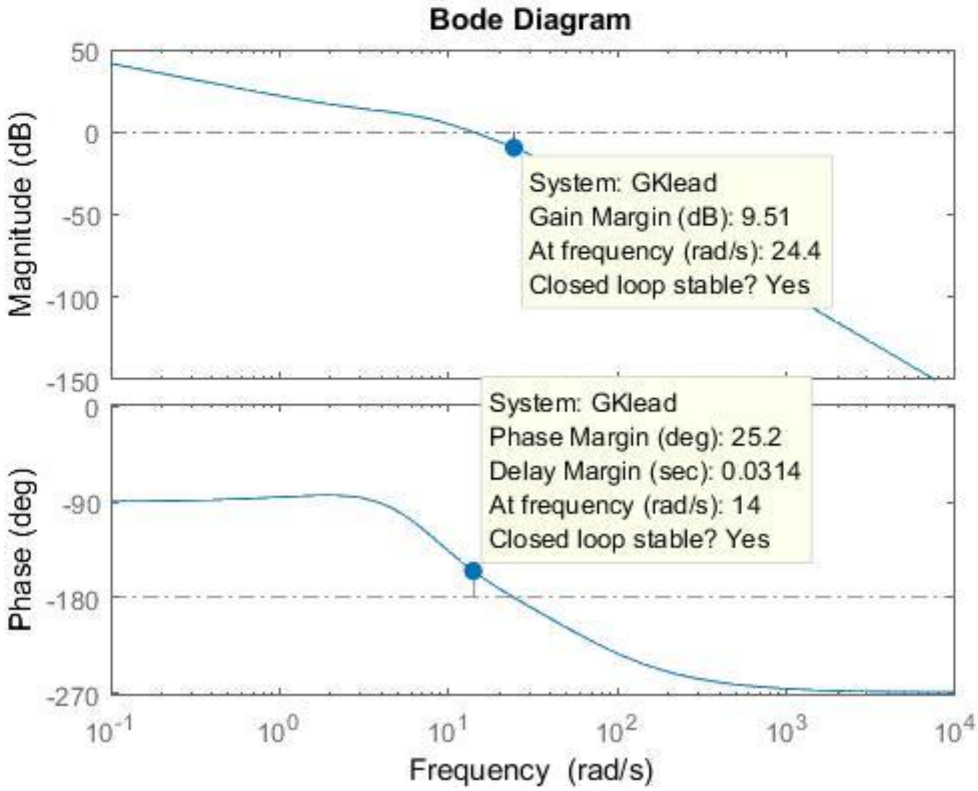

Figure 7

```
[Gm2,Pm2,Wcg2,Wcp2] = margin(GwKlead)
figure(4)
bode(GwKlead)
Pm=(60-Pm2)+0.1*(60-Pm2)
alpha2=1/0.3
[MAG,PHASE,W] = bode(GwKlead);
GM2=(1/sqrt(alpha2))
phase2= interp1( squeeze(MAG), squeeze(PHASE), GM2)
w2= interp1( squeeze(MAG), W, GM2)
Z2=w2/sqrt(alpha2)
P2=alpha2*Z2
Klead2=tf([1 Z2],[1 P2])*alpha2
Gw2Klead=GwKlead*Klead2;
Gw2Kleadc=feedback(Gw2Klead,1)
figure(5)
step(Gw2Kleadc)
figure(6)
bode(Gw2Klead)
figure(7)
K1=1/1.385
Gw2KleadK1=GwKlead*Klead2*K1;
Gw2KleadKc=feedback(Gw2KleadK1,1);
step(Gw2KleadKc)
Gz2KleadK1=c2d(Gw2KleadK1,Ts,'zoh')
Gz2KleadK1c=feedback(Gz2KleadK1,1)
figure(8)
step(Gz2KleadK1c)
K2=1/0.99
```

```
Gz2KleadK1K2=GwKlead*Klead2*K1*K2;
Gz2KleadK1K2c=feedback(Gz2KleadK1K2,1)
% % % % % % % % % % % %
```

By using 2 lead compensator and 2 proportional compensators to adjust response, the plant in w-domain meets all the requirement, however, to verify if it is the same performance, the inverse transfer of w-domain is needed.

To achieve that, $W = \frac{2}{Ts}\left[\frac{z-1}{z+1}\right] = \frac{2}{0.008}\left[\frac{z-1}{z+1}\right] = 250\left[\frac{z-1}{z+1}\right]$

$$Klead1 \times Klead2 \times K1 \times K2 = 1.01 \times 0.722 \times \frac{20w + 63.02}{w + 63.02} \times \frac{3.333w + 35.15}{w + 35.15} =$$

$$\frac{5000(z-1) + 63.02(z+1)}{250(z-1) + 63.02(z+1)} \times \frac{833.25(z-1) + 35.15(z+1)}{250(z-1) + 35.15(z+1)} =$$

$$Klead\ in\ z = \frac{20w + 63.02}{w + 63.02} = \frac{5063.02z - 4936.98}{313.02 - 186.98}$$

Equation 6

$$\text{Klead2 in z} = \frac{3.333w + 35.15}{w + 35.15} = \frac{868.4z - 798.1}{285.15z - 214.85}$$

Equation 7

```
Gz2KleadK1K2=c2d(Gw2KleadK1K2,Ts,'zoh')
Gz2KleadK1K2c=feedback(Gz2KleadK1K2,1)
Kzlead1=tf([5063.02 -4936.98],[313.02 -186.98],Ts)
Kzlead2=tf([868.4 -798.1],[285.15 -214.85],Ts)
GzKzlead1Kzlead2=Gz*Kzlead1*Kzlead2
GzKzlead1Kzlead2c=feedback(GzKzlead1Kzlead2*K1*K2,1)
figure(8)
step(Gw2KleadK1K2c,Gz2KleadK1K2c,GzKzlead1Kzlead2c)
```

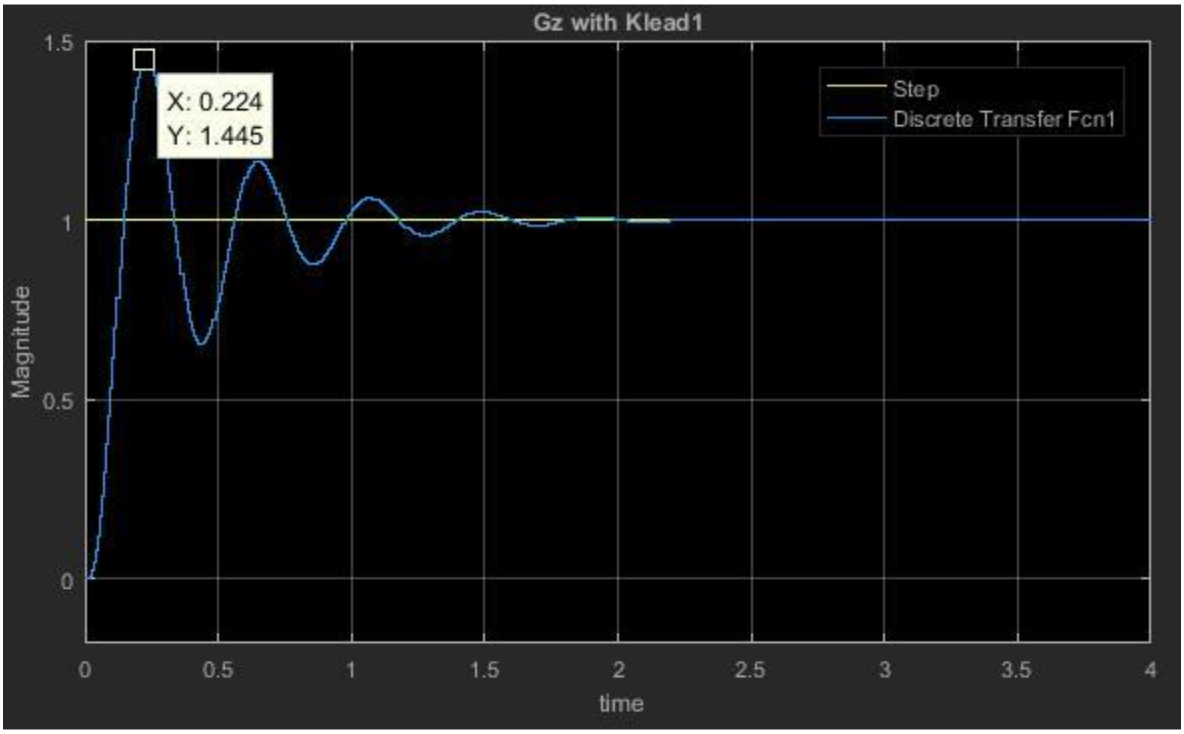

Figure 8

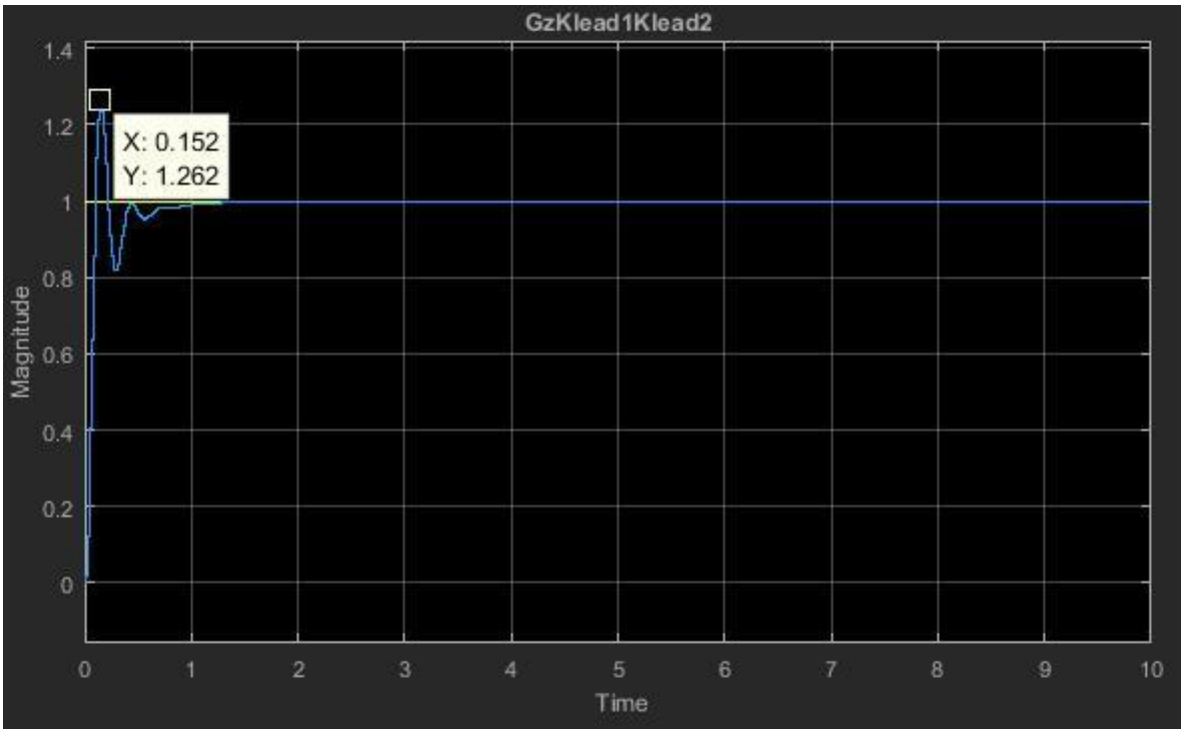

Figure 9

Type equation here.

Conclusion

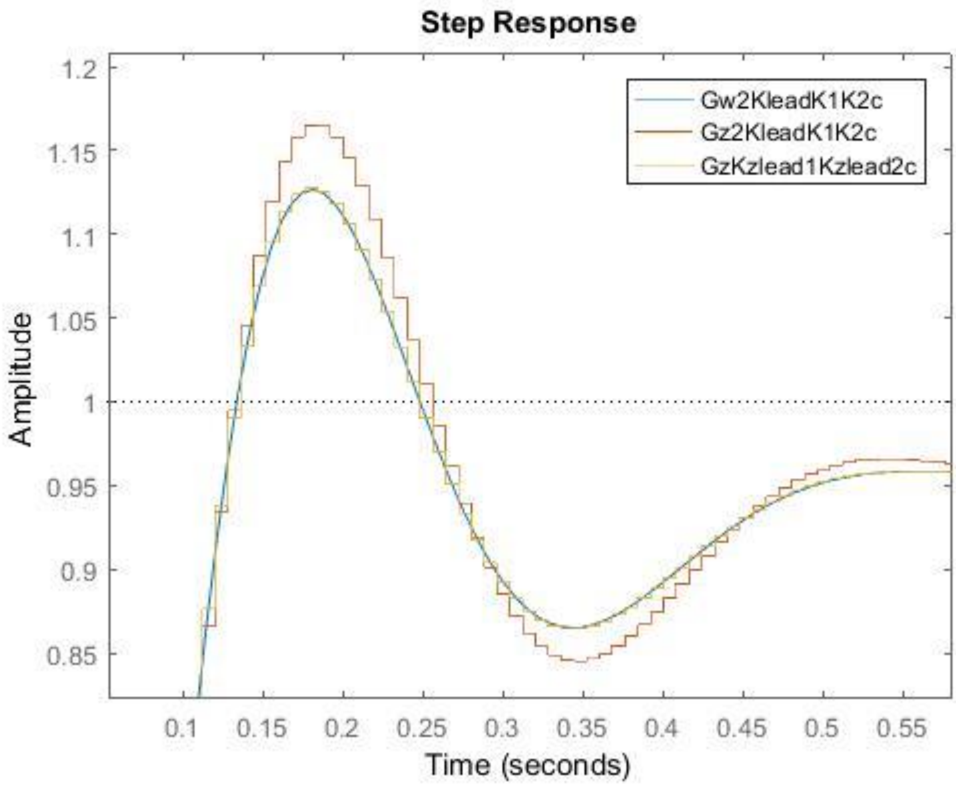

Figure 10

This picture shows the comparison of G in w-domain and compensated by 2 lead compensators and 2 K, and the matched z-domain transfer by using two different methods. The result shows the embedded function, c2d, actually gives inaccurate result of the response, and it also induces higher percentage overshoot. The inverse w-domain transform is an accurate way of connecting discreet and continuous system

Table 3

Specification	w-domain	C2d method	Inverse w-domain
Percentage Overshoot	12.7	16.5	12.8
Rise Time	0.0812	0.0783	0.0816
Settling Time	0.494	0.475	0.489
Offset	0	0	0

The digital system is alternative form of the analogue system, which provides better energy efficacy, and it is much easier for transforming without losing lots of detail if the resolution is suitable for current analogue system. Also, the digital system is more reliable under electromagnetic interference, its signal is more distinguishable than the analogue signal, however less detail if the sampling time is too long. The short sampling time can provide much detail of analogue plant but may induce noise when overlapping happened.