ACS214

**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:

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.

The original plant is

**To get the 15% overshoot, the following equations will be used during the calculation, the maximum overshoot can be obtained by setting to get ** by using Equation 1, andthen 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.



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



Equation 2: Phase Margin of current plant by damping ratio

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

**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

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**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ** | 0.05 | 0.1 | 0.2 | 0.24 | 0.3 | 0.5 | 1 |
| *G dB* | 13 | 10 | 7 | 6 | 5 | 3 | 0 |
| *m*  (deg.) | 65 | 55 | 42 | 38 | 33 | 19 | 0 |

Table

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.

Equation

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)

Equation

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 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)

This shows the good resoponse of step input with 2 Klead and a propotional compensator

Task B

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

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

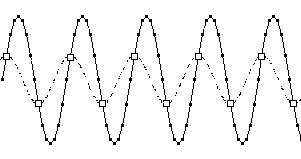


Figure the sampling frequency and time difference

(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.

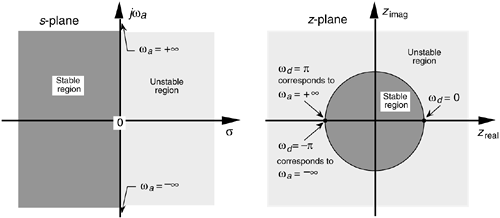
Compensator design should be similar to Task A, however, due to discrete system approximation, the response will be very different in general or dramatically changes at critical point, therefore, a new intermedium domain cross continuous and discreet domain is necessary 

Figure 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

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)

The design

Gw=Gz2Gw(Gz,Ts)

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)

Gwc=feedback(Gw,1)

figure(2)

bode(Gw)

figure(3);

step(Gwc);

[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);

[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)

Conclusion