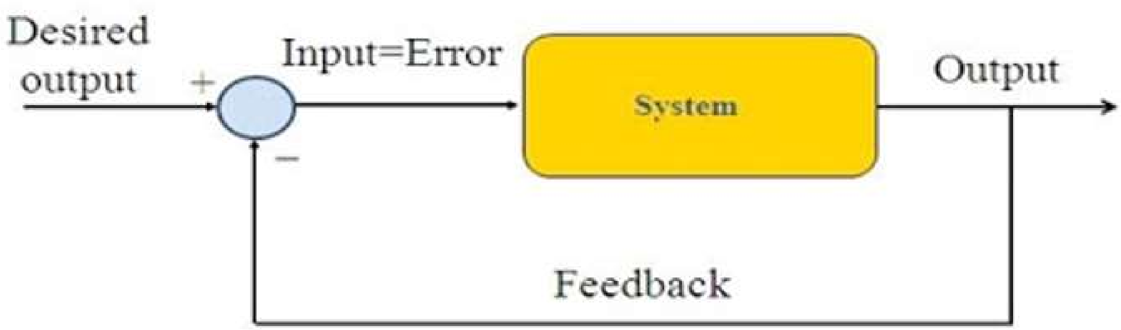
MEEM 4775 Project 3 Report

DESIGN OF A CONTROL SYSTEM USING BODE PLOTS



Created & Submitted by: Vrushaketu Mali

10-Dec-2020

Table of Contents

Chapter 1. Introduction

Most of the dynamic systems which we consider in the practice are having a compensator or controller. The role of controller is to make the system follow a specific command input as well as to avoid any disturbance. When the system is coupled with the controller then the feedback is necessary in order for controller to determine what action it needs to take. For example, a controller may reduce the input if the actual output is more than required output and may increase the input if the actual output is less than the required output. Therefore, a system with the compensator or controller is called as closed loop system with feedback mechanism. There are several types of the control system such as PID, State Feedback, LQR. The role of the control engineer is to design a control system, specifically design the control system gains such that closed loop system will have desired response characteristics.

Chapter 2. Summary

This project aims at developing a control system for the given transfer function of a plant using Bode Plots. Thus, the project involves designing the control system parameters for required time domain & Bode specifications, writing a MATLAB script and building a mathematical model for simulation of the system. The control system is designed using Bode plots with step-by-step approach until the specifications are met.

Accomplishments:

1. In this project, first of all we have analyzed the open loop system in order to gain the information of performance characteristics.
2. Then, we have designed the lead compensator as a controller.
3. As per the initial design, again the changes are made in order to reach the desired specifications.
4. Root locus and Nyquist plot is presented in order to co-relate the design with Bode plot.
5. After observing the performances meeting the specification, we have analyzed the sensor latency for system instability.
6. Finally, the sensor latency analysis done through simulation is verified using Bode plots.

Chapter 3. Project/Problem Statement

Consider a dynamic system whose transfer function is

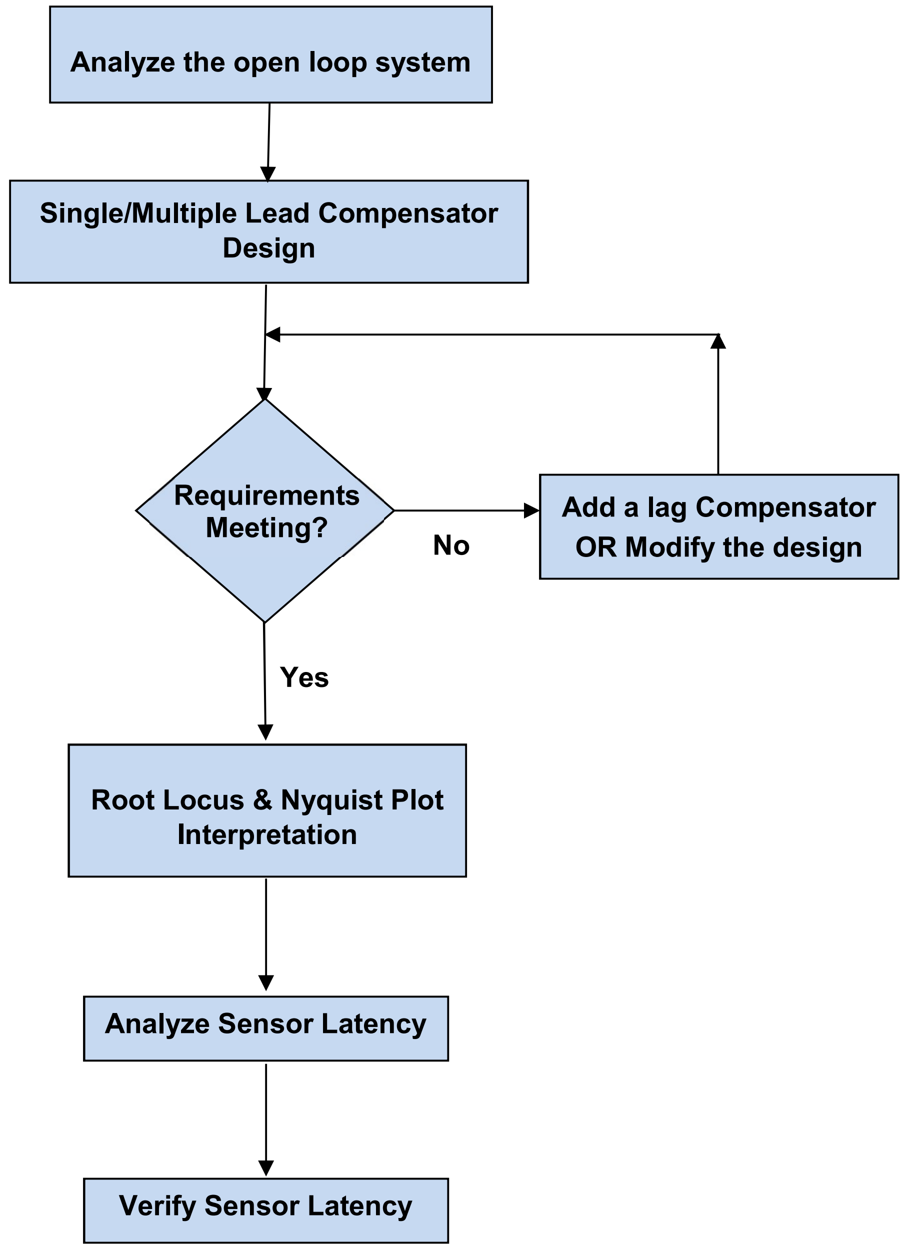
$$\frac{Y}{U} = \frac{50\,(s+4)}{s\,(s+6)\,(s^2+4s+25)}$$

where Y is position in meters and U is force in Newtons. You must design a control system with the following requirements:

1. The only measured quantity is y.
2. Phase margin greater than 40_.
3. Velocity error constant greater than 20
4. Step response 5% settling time less than 1.0 seconds.
5. The force should not exceed 15 N.

At a minimum, your design must be justified using Bode Plot techniques. Root locus and Nyquist interpretation must also be provided along with simulation results illustrating satisfaction of the requirements. When making your Nyquist plot, be sure to use rational scales so that it can be easily interpreted. After designing your control strategy, determine the maximum allowable time delay that can be tolerated on the position feedback. This could represent latency of a wireless sensor. Verify the simulated value using either a Bode or Nyquist plot.

Chapter 4. Project Approach



Chapter 5. Analysis of loop transfer function with controller gain

Before running into designing control system and analyzing the loop transfer function. Also, in the specification requirement, we have been provided the requirement of steady state error in terms of the velocity error constant. Therefore, we will first determine the controller gain required.

5.1 System Type

The plant transfer function is given in the project statement as

$$G(s) = \frac{Y}{U} = \frac{50\ (s+4)}{s\ (s+6)\ (s^2+4s+25)}$$

From the transfer function, we can observe that, the number of zeros present is 1. Therefore, this is type 1 system. For this system, the steady state error for step input is always zero. The steady state error for ramp input is given in terms of velocity error constant.

5.2 Controller gain for steady state error requirement

As per the requirement, the velocity error constant must be greater than 20/3.

The velocity error constant is given as $K_v = \lim_{s \to 0} s\ D\ G$

The transfer function for lead compensator is given as $D = \frac{K_c\ (T\ s+1)}{\propto T\ s+1}$

Therefore, $\frac{20}{3} < \lim_{s \to 0} s\ \frac{K_c\ (T\ s+1)}{\propto T\ s+1}\ \frac{50\ (s+4)}{s\ (s+6)\ (s^2+4s+2\ \ )}$

$$\frac{20}{3} < \frac{K_c\ 4}{3}$$

Therefore, $K_c > 5$

The controller gain must be greater than 5 in order to meet the requirement of velocity error constant.

5.3 Bode analysis of loop transfer function with controller gain

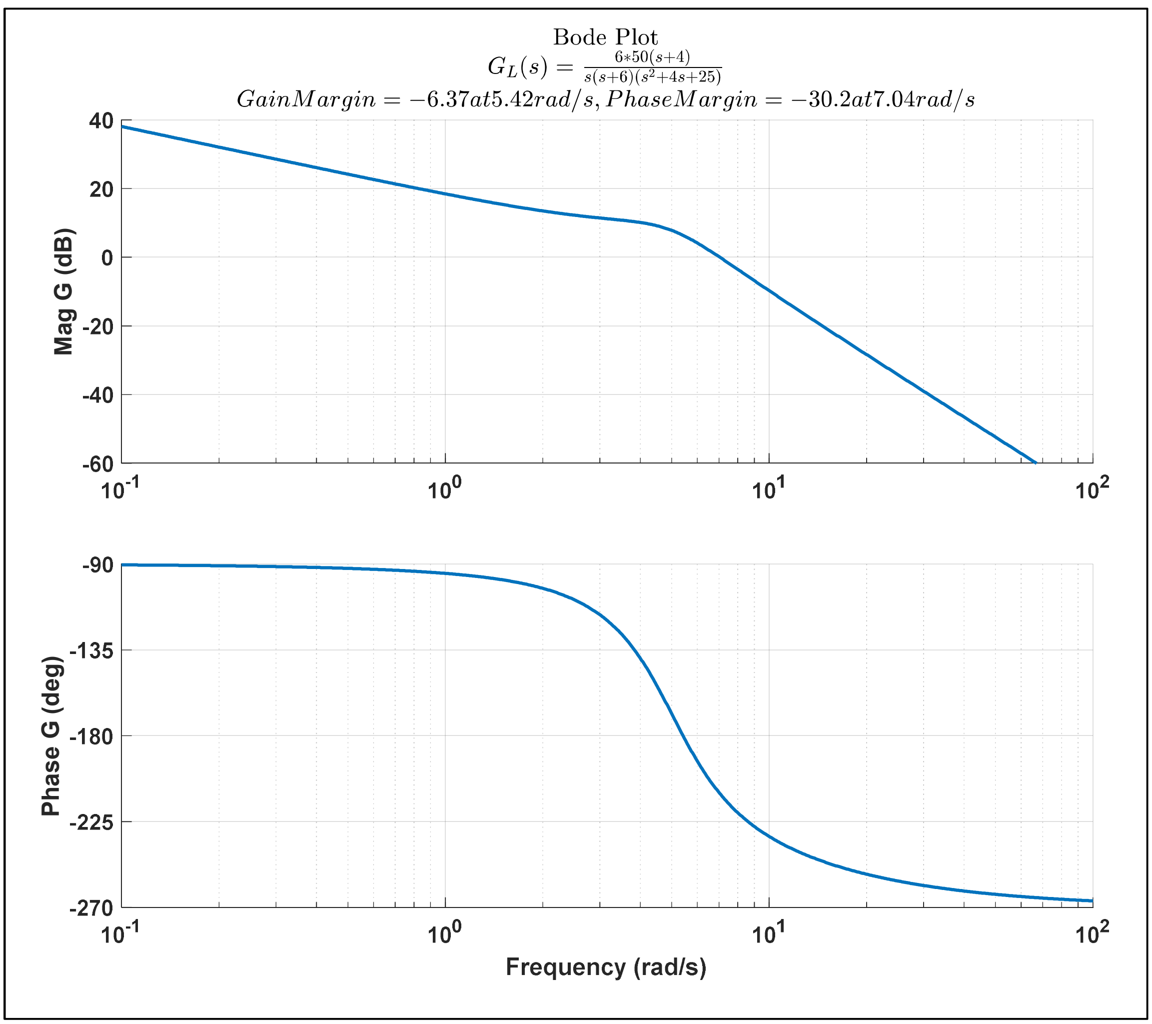
Now, consider the loop transfer function with controller gain and check the existing phase margin so that we can know how much we are offset from the requirement.

Since controller gain needs to be greater than 5, consider the value as 6 and form the loop transfer function as

$$G_L(s) = \frac{6 * 50\ (s+4)}{s\ (s+6)\ (s^2+4s+25)}$$

Below is the bode plot of the loop transfer function with controller gain as 6.

The current phase margin is -30.2 deg which means the system is unstable. We need phase margin more than 40 deg that means, we need to ask for phase margin of at least 70.2 deg during compensator design.



Chapter 6. Control System Design

From the last chapter, we recognized that the system is unstable. The phase margin is way beyond the spec. Since required change in phase margin is more than 50°, we need two lead compensators as single lead compensator can be used for maximum 50°.

6.1 First lead compensator design

The transfer function for the lead compensator is given by $D = \frac{(T\ s+1)}{\propto T\ s+1}$

a) For the first lead compensator, we would ask for 45°.

b) Compute $\propto$ for 45° phase.

$$\propto_1 = \frac{1 - \sin \emptyset_1}{1 + \sin \emptyset_1} = \frac{1 - \sin 45}{1 + \sin 45} = 0.1716$$

c) Compute the value of frequency, at which we want to put the phase.

For this, we need to check what will be the magnitude drop for asking this phase and then check that magnitude in original loop transfer function bode plot.

The magnitude drop corresponding to this $\propto_1$ value is -7.65 dB.

From looking at figure 1, the magnitude -7.65 dB is at 9.3 rad/s.

Therefore, we would put the phase at $\omega_{m1} = 9.3\ rad/s$

d) Compute T_1 from $\propto_1$ & ω_{m1}

$$T_1 = \frac{1}{\omega_{m1}\sqrt{\propto_1}} = \frac{1}{9.3\sqrt{0.1716}} = 0.2596$$

e) Transfer function of first lead compensator design

From above values, the transfer function of first lead compensator would be

$$D_1 = \frac{(T_1\ s + 1)}{\propto_1 T_1\ s + 1} = \frac{0.26\ s + 1}{0.044\ s + 1}$$

6.2 Second lead compensator design

a) For the second lead compensator, we need to ask for remaining phase which is 25.2°

b) Compute $\propto$ for 25.2° phase.

$$\propto_2 = \frac{1 - \sin \emptyset_2}{1 + \sin \emptyset_2} = \frac{1 - \sin 25.2}{1 + \sin 25.2} = 0.4027$$

c) Compute the value of frequency, at which we want to put the phase.

For this, we need to check what will be the magnitude drop for asking this phase and then check that magnitude in $\mathrm{G} * D_1$ transfer function bode plot.

The magnitude drop corresponding to this $\propto_2$ value is –3.95 dB.

From looking at figure 2, the magnitude –3.95 dB is at 11.1 rad/s.

Therefore, we would put the phase at $\omega_{m2} = 11.1\ rad/s$

d) Compute T_2 from $\propto_2$ & ω_{m2}

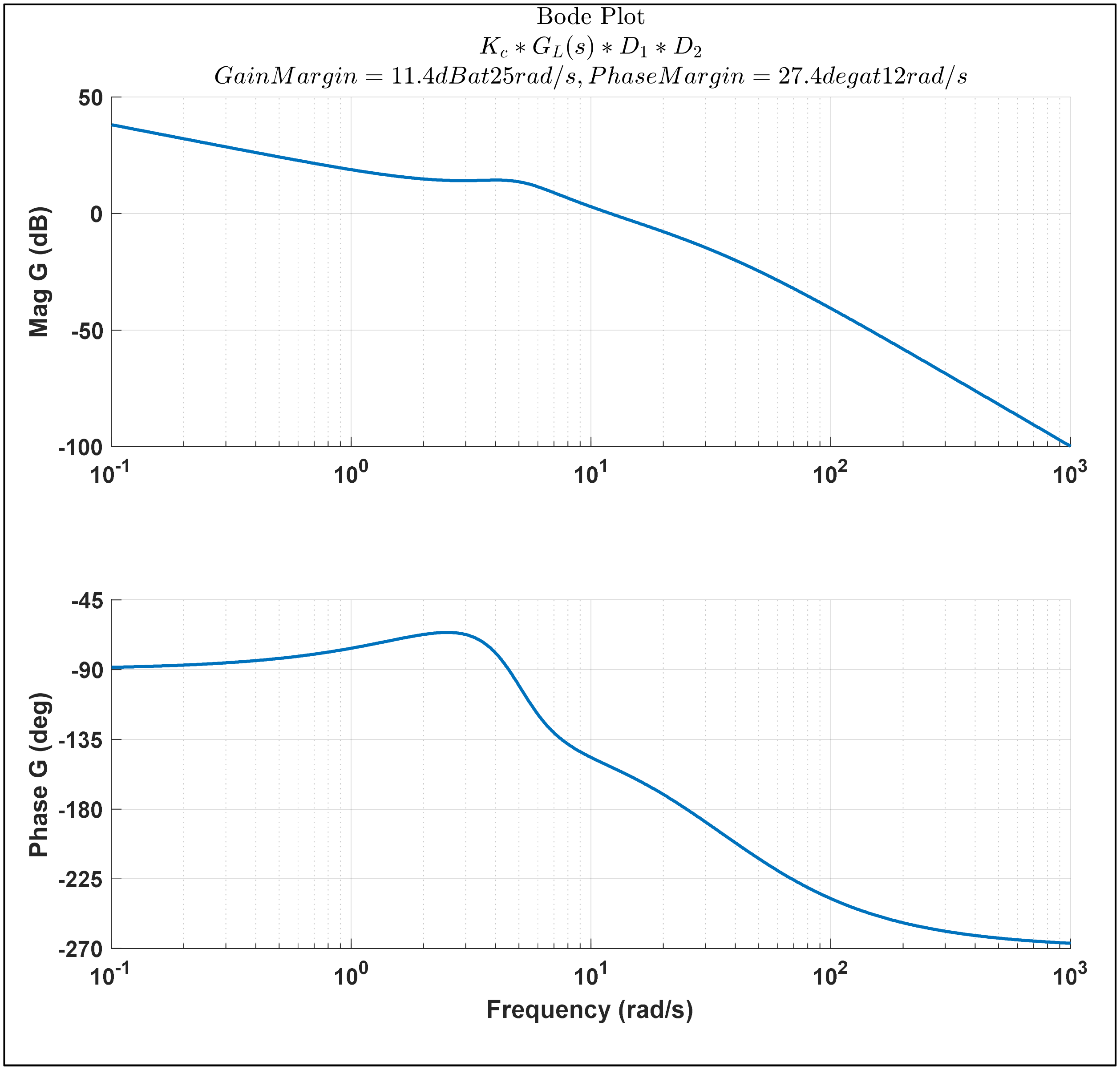
$$T_2 = \frac{1}{\omega_{m2} \sqrt{\propto_2}} = \frac{1}{11.1 \sqrt{0.4027}} = 0.142$$

e) Transfer function of second lead compensator design

From above values, the transfer function of first lead compensator would be

$$D_2 = \frac{(T_2\, s + 1)}{\propto_2 T_2\, s + 1} = \frac{0.142\, s + 1}{0.024\, s + 1}$$

f) Bode Plot of loop transfer function after second lead compensator design



If we see above bode plot of the loop transfer function considering two lead compensator designs, the phase margin is still not meeting the requirement. This is because, while shifting the magnitude plot for asking required phase, there was some phase loss happened which causes the phase margin behind what we needed. Therefore, we need to either ask for more phase margin in second lead compensator design or design a new compensator.

However asking for more phase margin may not meet the settling time requirement and adding another lead compensator would have effect as described in next section, so we will design a new lag compensator for meeting the requirements.

6.3 Lag compensator design

From figure 3, we analyzed that two lead compensators is still not meeting the performance requirement in terms of phase margin. If we add another lead compensator, we might meet the phase margin requirement but the settling time requirement may not be satisfied because it causes faster response. Because of faster response, the actuator size required for lead compensator is also more. However, we are limited with the actuator force. Therefore, we would explore lag compensator design which after adding to previous two lead compensators design will create combined lead lag compensator.

a) Determine the gain cross-over frequency ω_{gd} based on the amount of phase required.

From previous figure, we analyzed that the phase margin is not more than 40°
Therefore, we will ask for required phase margin and perhaps more in lag compensator because of some phase loss. We will target for 45°
For 45° phase margin, the phase needed is 135° and at that phase the gain cross-over frequency ω_{gd} is 7.6 rad/s

b) Determine the amount (M) that the magnitude plot must shift in order to achieve the desired gain crossover frequency.

From the magnitude plot, the magnitude at 7.6 rad/s frequency is M = 7.51 dB.

c) Compute the design parameter β

The parameter β in lag compensator design is calculated as 10^(M/20);
Thus, $\beta = 2.3741$

d) Compute T

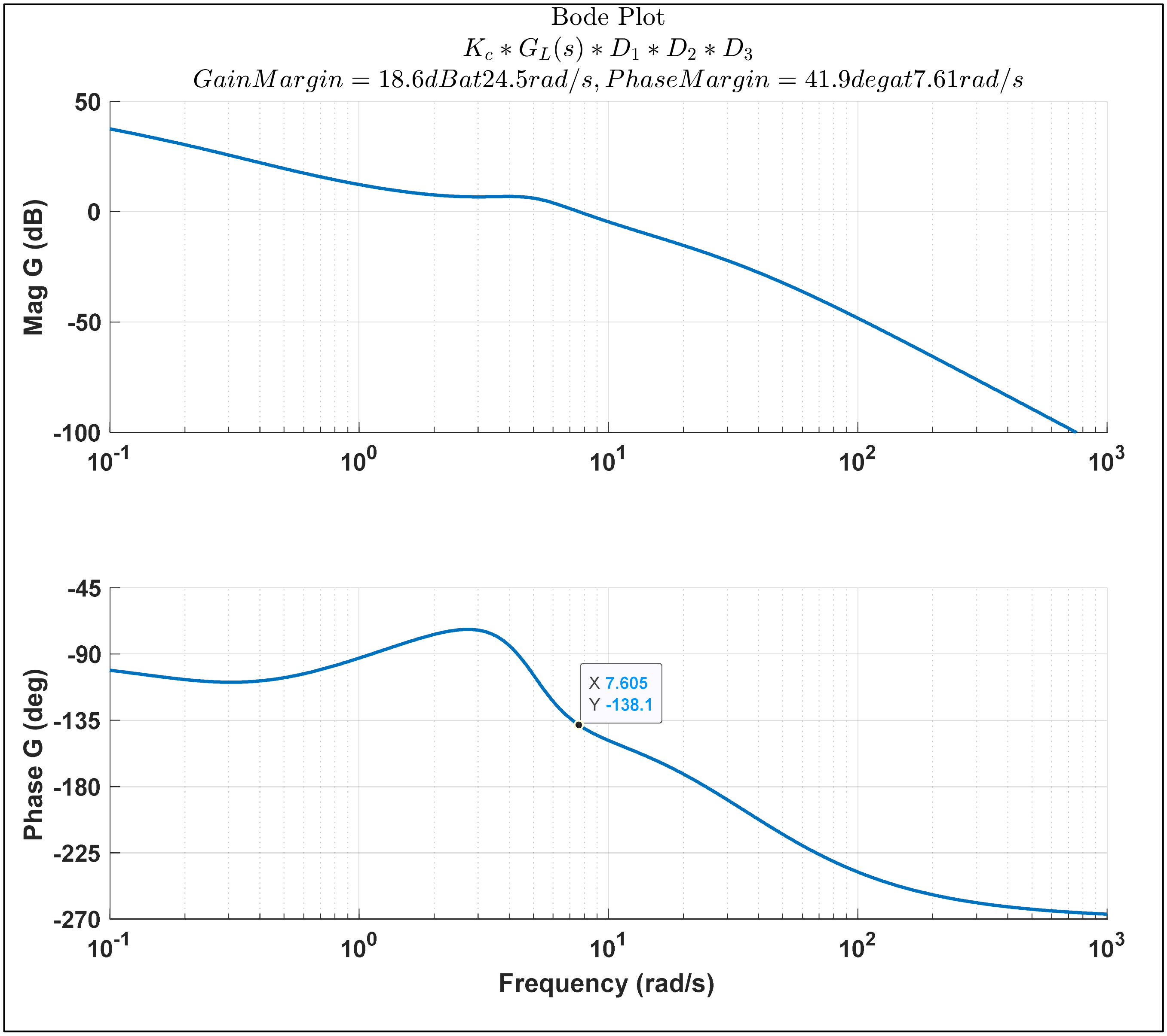
T can be calculated as $10/\omega_{gd}$. However, we can increase the parameter 10 in order to adjust the phase bubble. We will take the parameter as 13.

$$T = \frac{13}{\omega_{gd}} = \frac{13}{7.6} = 1.7105\ s$$

e) Transfer function of lag compensator design

$$D_3 = \frac{(T\ s + 1)}{\beta\ T\ s + 1} = \frac{1.711\ s + 1}{4.061\ s + 1}$$

6.4 Bode Plot of loop transfer function after lead lag compensator design



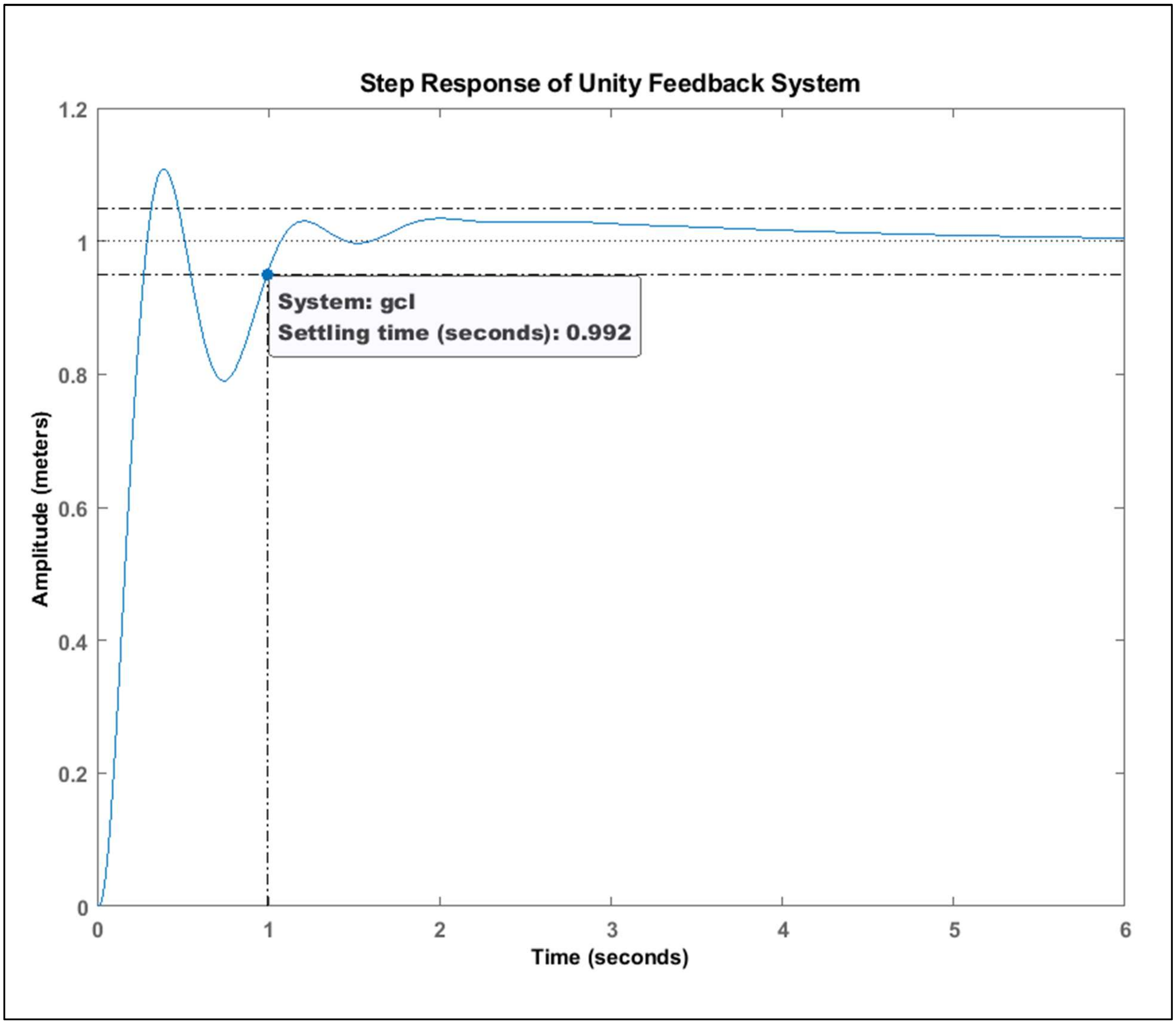
If we see above bode plot of the loop transfer function considering two lead compensators and a lag compensator design then we are meeting the phase margin requirement of 40°. The phase margin achieved after implementing the controller design is **41.9°**

The transfer function for controller is then given as $D = D_1 * D_2 * D_3$

$$D = \frac{(s + 6.92)(s + 3.92)(s + 0.58)}{(s + 49.05)(s + 20.69)(s + 0.24)}$$

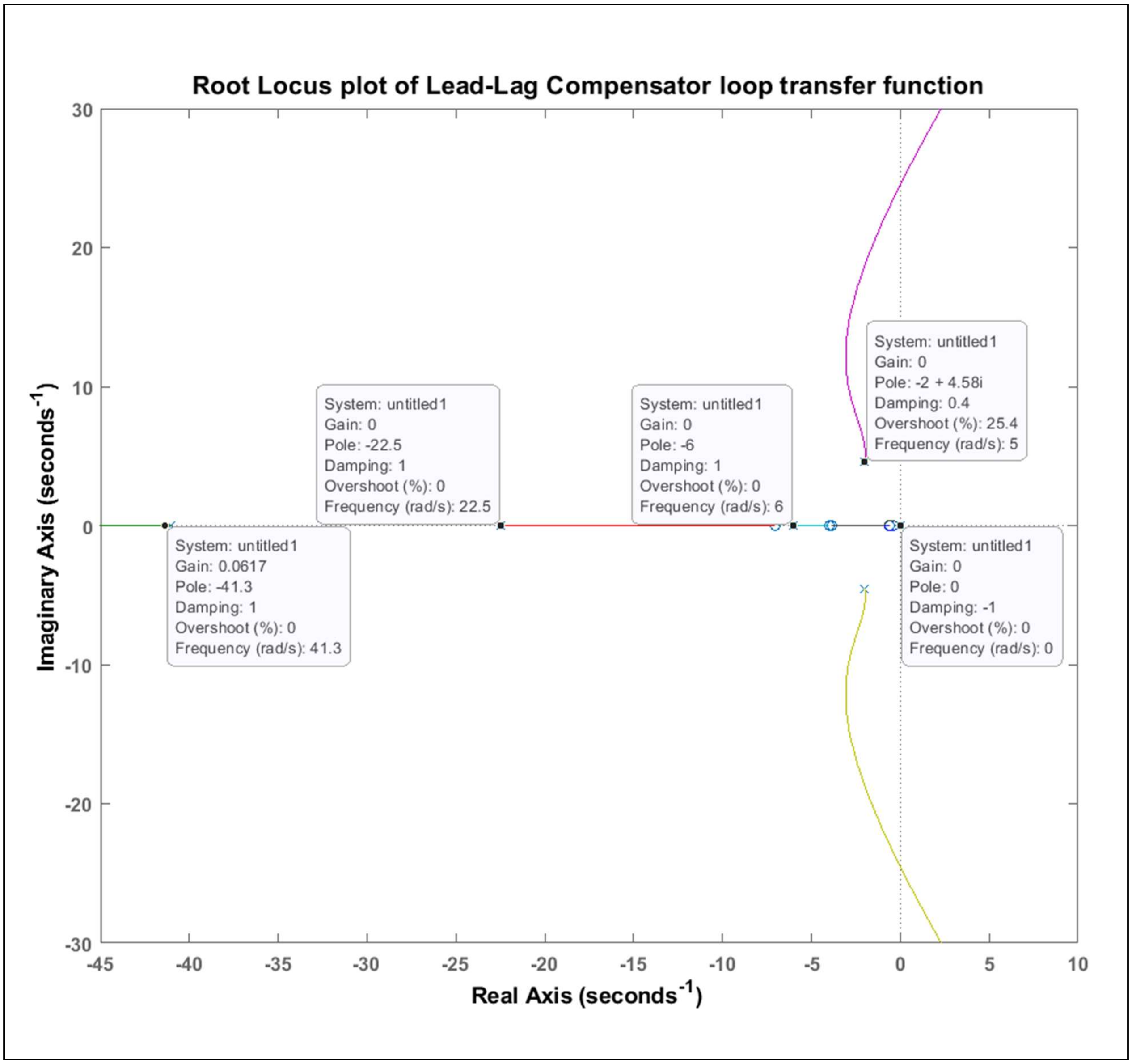
6.5 Step response of unity feedback system

As one of the requirements of the controller design is that the step response of the unity feedback system must have settling time less than 1 seconds for 5% criteria. Therefore, let's form the transfer function of the unity feedback system having our controller design and analyze the step response. Please refer appendix for MATLAB code.



Above figure is the step response plot of the unity feedback system which incorporates the controller we have designed. The data point shows that for 5% settling criteria, the settling time is 0.992 seconds which is less than 1 second and thus we are meeting the requirement.

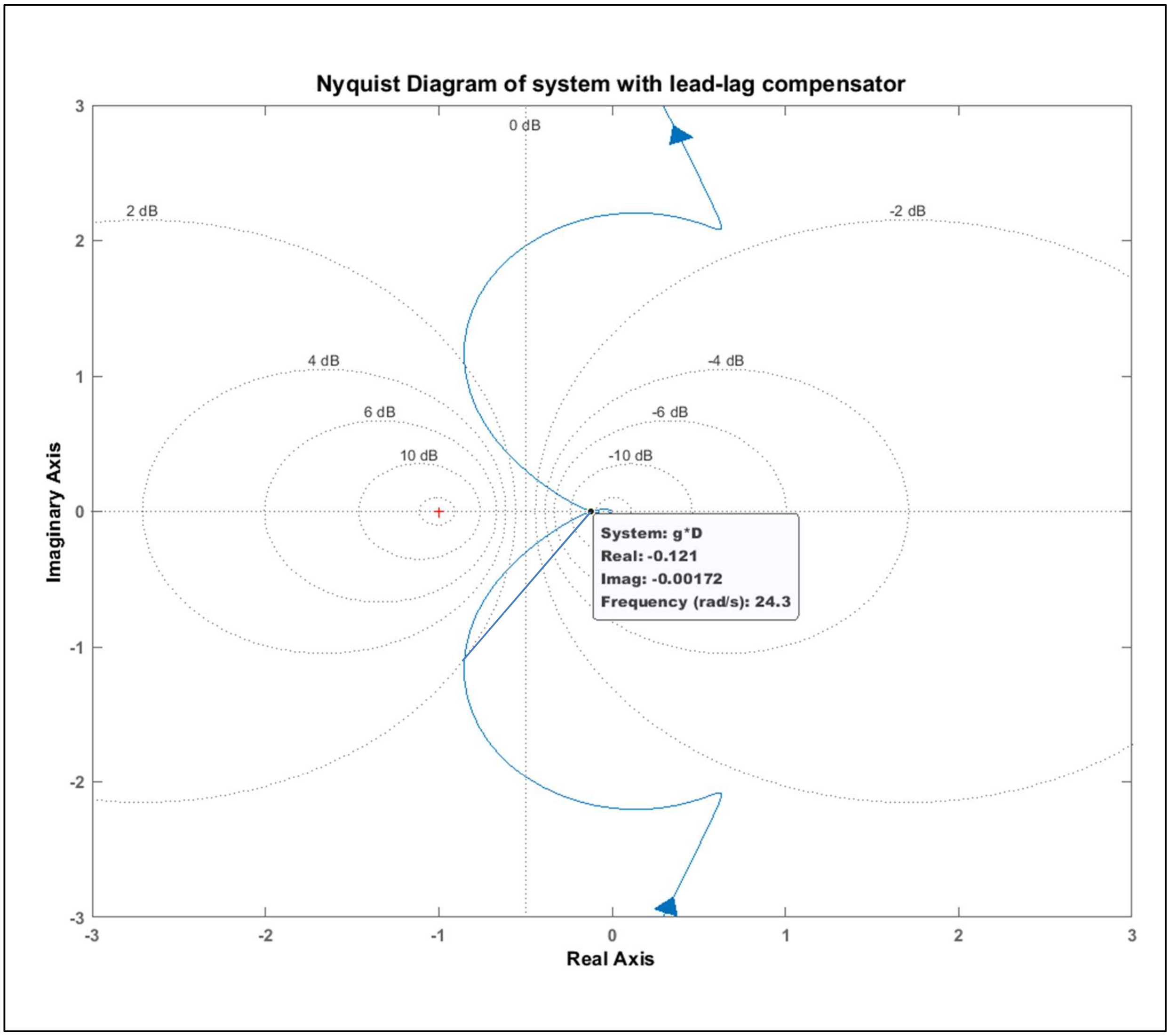
6.6 Root Locus Interpretation



The above figure shows the root locus plot of the system at gain 6. If we evaluate the poles of the loop transfer function g*D by using pole command, then we get 0, -41.05, -22.45, -6, -2.0 ± 4.5j , -0.24

These poles are same as those shown in the root locus plot.

6.7 Nyquist Plot Interpretation



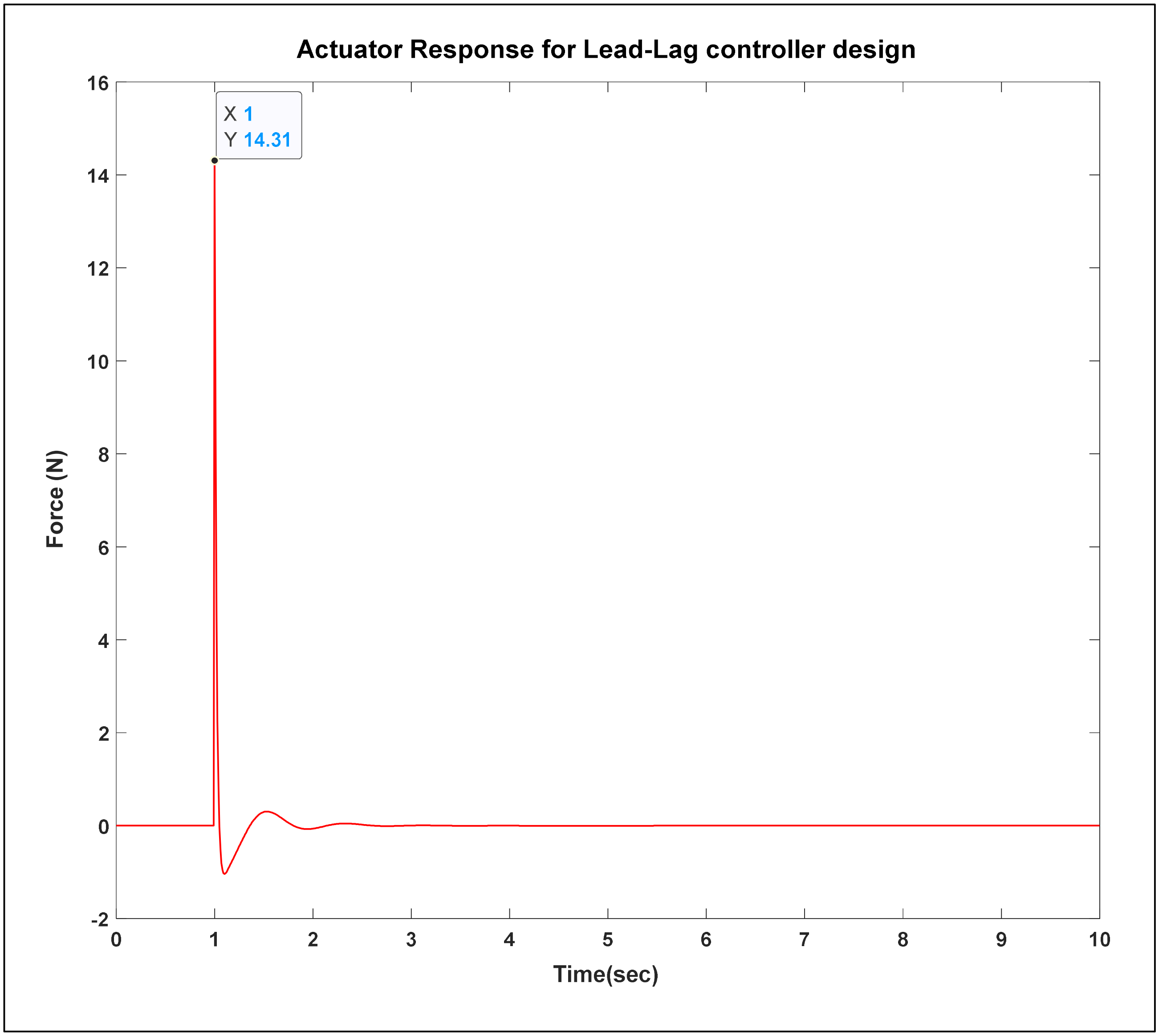
Above figure is the Nyquist plot of the system with lead-lag compensator. The Nyquist plot intersects real axis at 0.121. From this we can calculate the gain margin and compare with the gain margin achieved in Bode plot.

The gain margin from real axis intersection will be $-20 * \log_{10} 0.121 = \ 18.34\ dB$

The gain margin from Bode plot is 18.62 dB.

The phase margin will be the angle between the line shown from the real axis intersection to circle intersection and the horizontal axis. Looking at the figure this is roughly 42°

6.8 Actuator Response for lead-lag compensator design



The above figure shows the actuator response after implementing the lead and lag controller and giving unit step reference command. The maximum actuator force is 14.3 N which is below the saturation limit of 15 N and this is achieved without using saturation block. Thus, it is beneficial as it is not inducing any non-linearity because of saturation.

6.9 Performance Summary of controller design

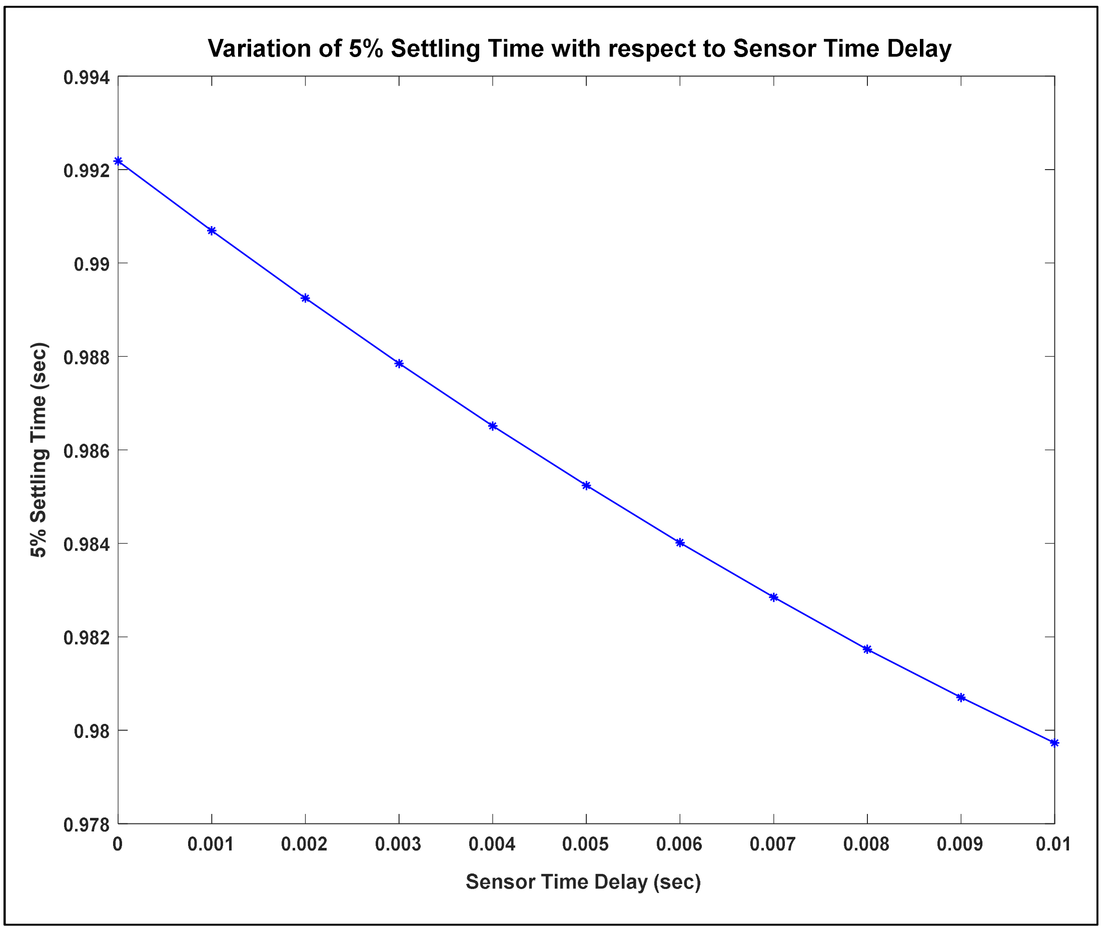
As we progressed through some finite iterations of controller design, we would tabulate the performance at each iteration. Below table shows the characteristics at each iteration.

Design Iteration	Design Type	Phase Margin	Settling Time	Velocity error constant	Remark
1	Single Lead	**-4.09^{o}**	Not required to check	8	**Design Fail**
2	Two Lead	**27.4^{o}**	Not required to check	8	**Design Fail**
3	Two lead and lag	**41.9^{o}**	**0.99 seconds**	8	**Design Pass**

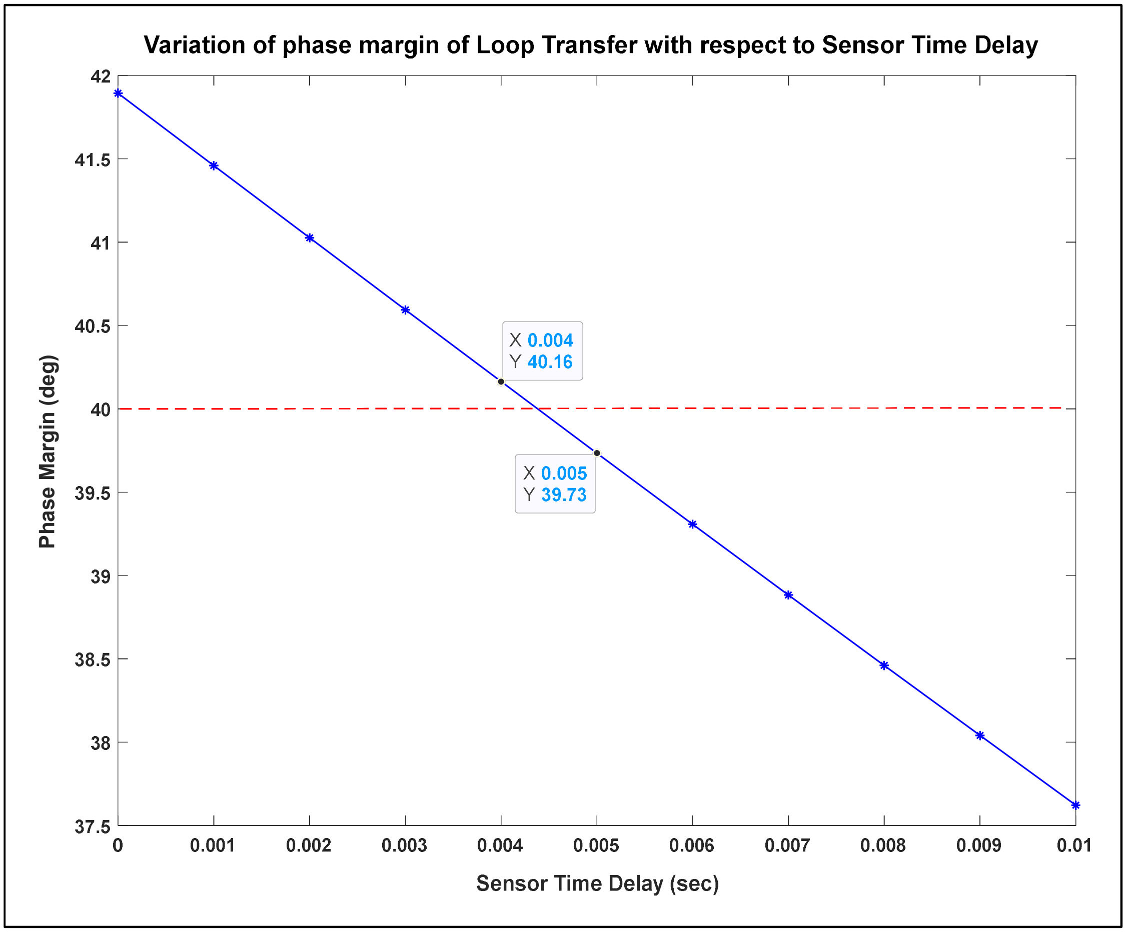
Chapter 7. Wireless Sensor Latency

7.1 Sensor Latency for out of spec performance

Now as we have controller design in place, we analyzed the performance with unity feedback. With unity feedback, the phase margin of loop transfer function is more than 40^o. Now, we would analyze the effect of the time delay in the sensor on position feedback on the performance in terms of phase margin and the settling time. Therefore, we will do certain iterations for sensor transfer function with different time delay and then will import new transfer function every time to check its effect.
We will plot two figures – first would be the variation of settling time with respect to sensor time delay and second would be the variation of phase margin with respect to sensor time delay.



The above figure shows that as sensor time delay is going to increase, the settling time is going to decrease therefore we will not have any problem on the settling time requirement with sensor time delay. Now we will check the effect on phase margin.

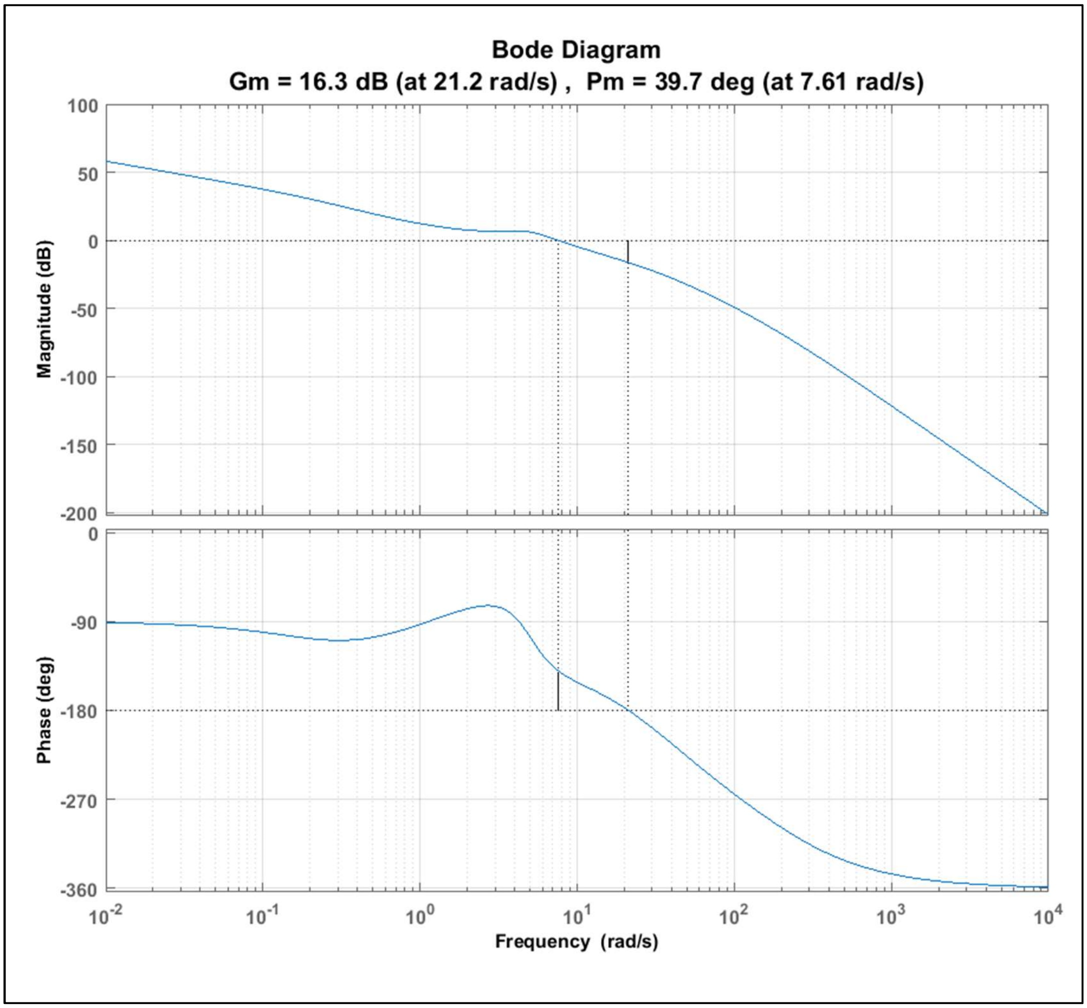


The above figure shows the variation of phase margin of loop transfer with respect to sensor time delay. It can be seen that till 4 ms of delay, the phase margin requirement is met however, once the sensor time delay is made 5 ms then the phase margin goes below the requirement and becomes 39.73°

Thus, maximum time delay before system performance goes out of spec is 4 milli-seconds.

7.2 Verification of Sensor Latency through Bode plot

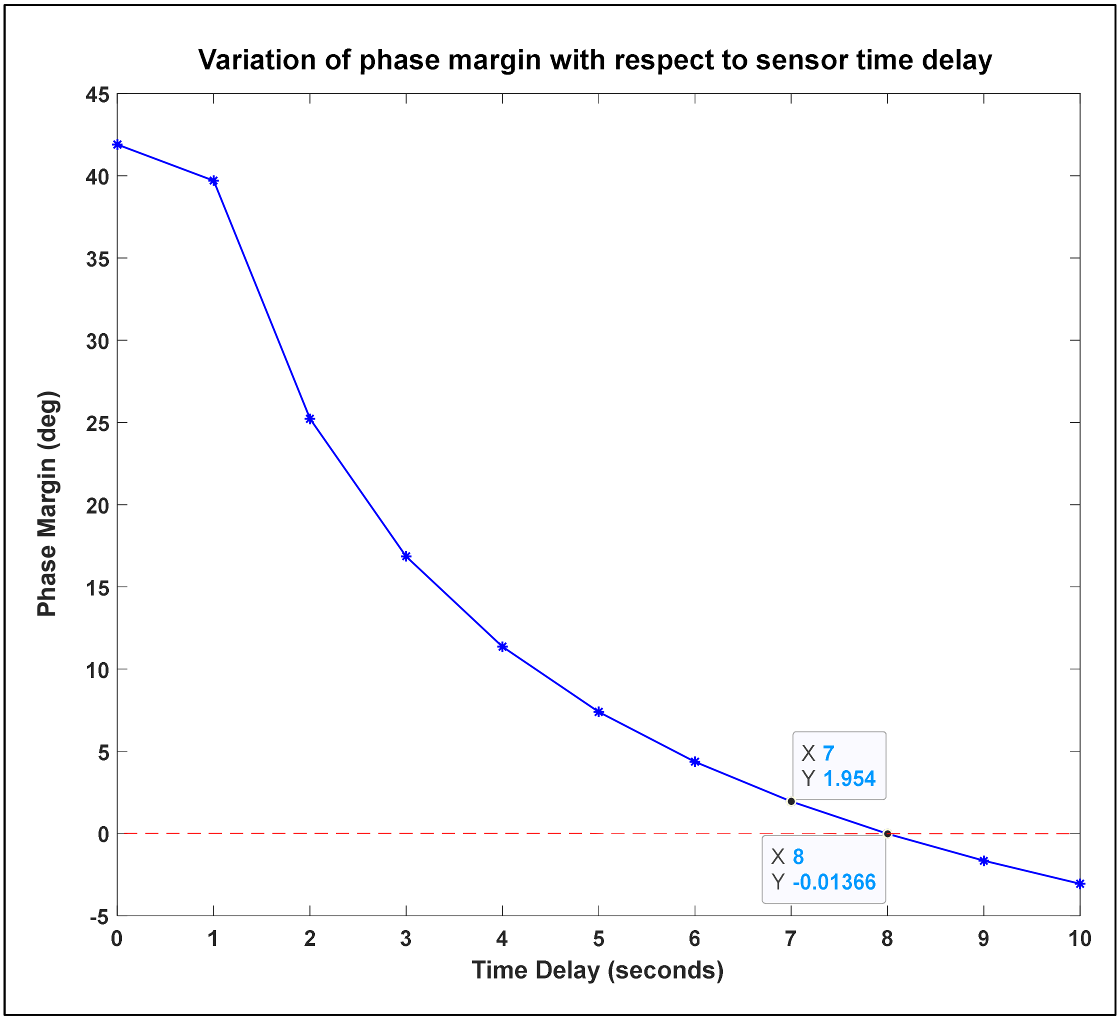
Now we will verify the sensor latency analyzed above through bode plot. For that, we can form the transfer function of sensor with 5 ms time delay and construct the bode plot.



The above bode plot of the loop transfer function having sensor with time delay of 5 ms verifies that the phase margin drops below 40°

7.3 Sensor Latency for system instability

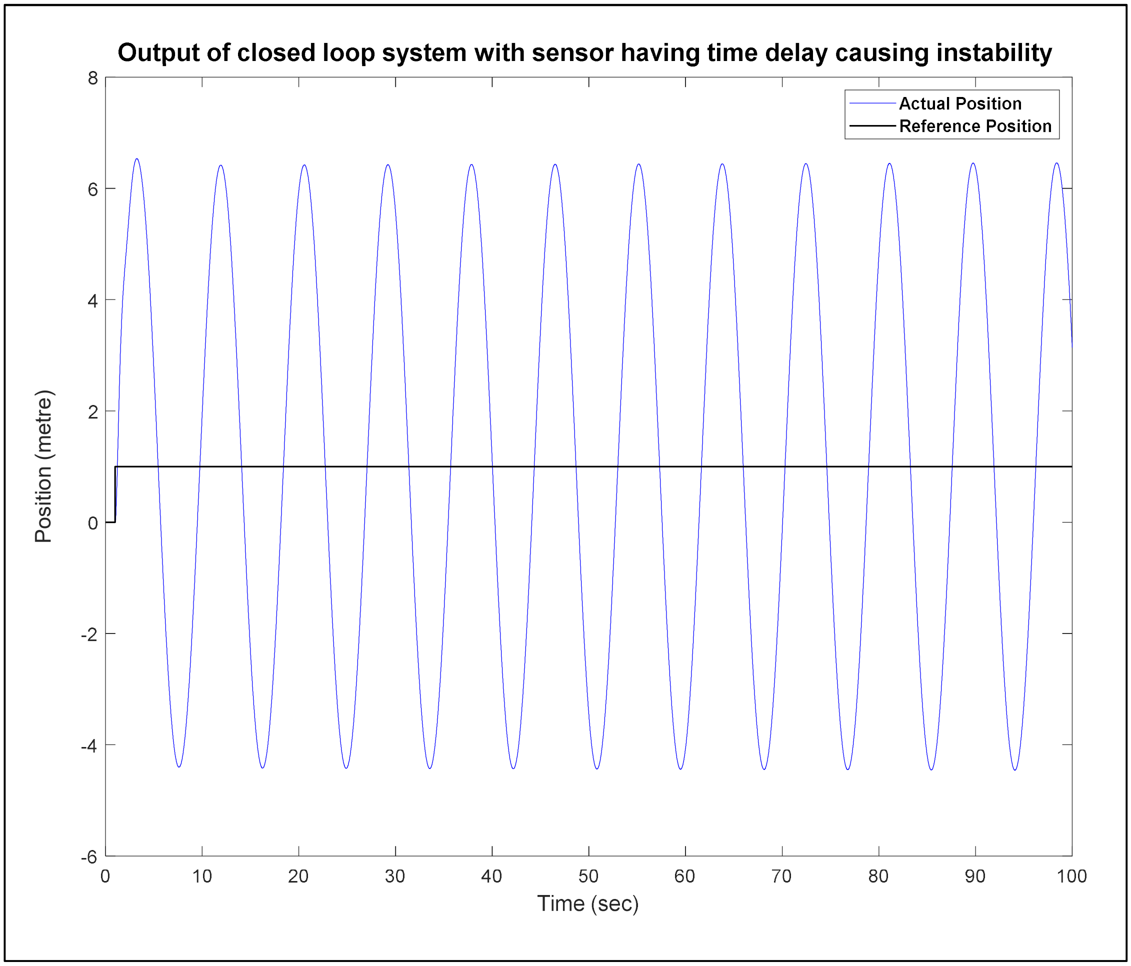
As we have seen that increase the sensor time delay decreases the phase margin, there would be a value of time delay where the phase margin would become negative. The negative phase margin illustrates that the closed loop system is unstable. Therefore, we will find out the time delay value at which system becomes unstable by plotting the variation of phase margin with respect to time delay and simulating the model in Simulink at that time delay and observing the output. The output should be unstable.



The above figure shows the variation of phase margin with sensor time delay. It can be seen that up to 7 seconds of time delay the system is stable and 8 seconds system becomes unstable as phase margin becomes negative.

7.4 Simulation of model at sensor time delay causing instability

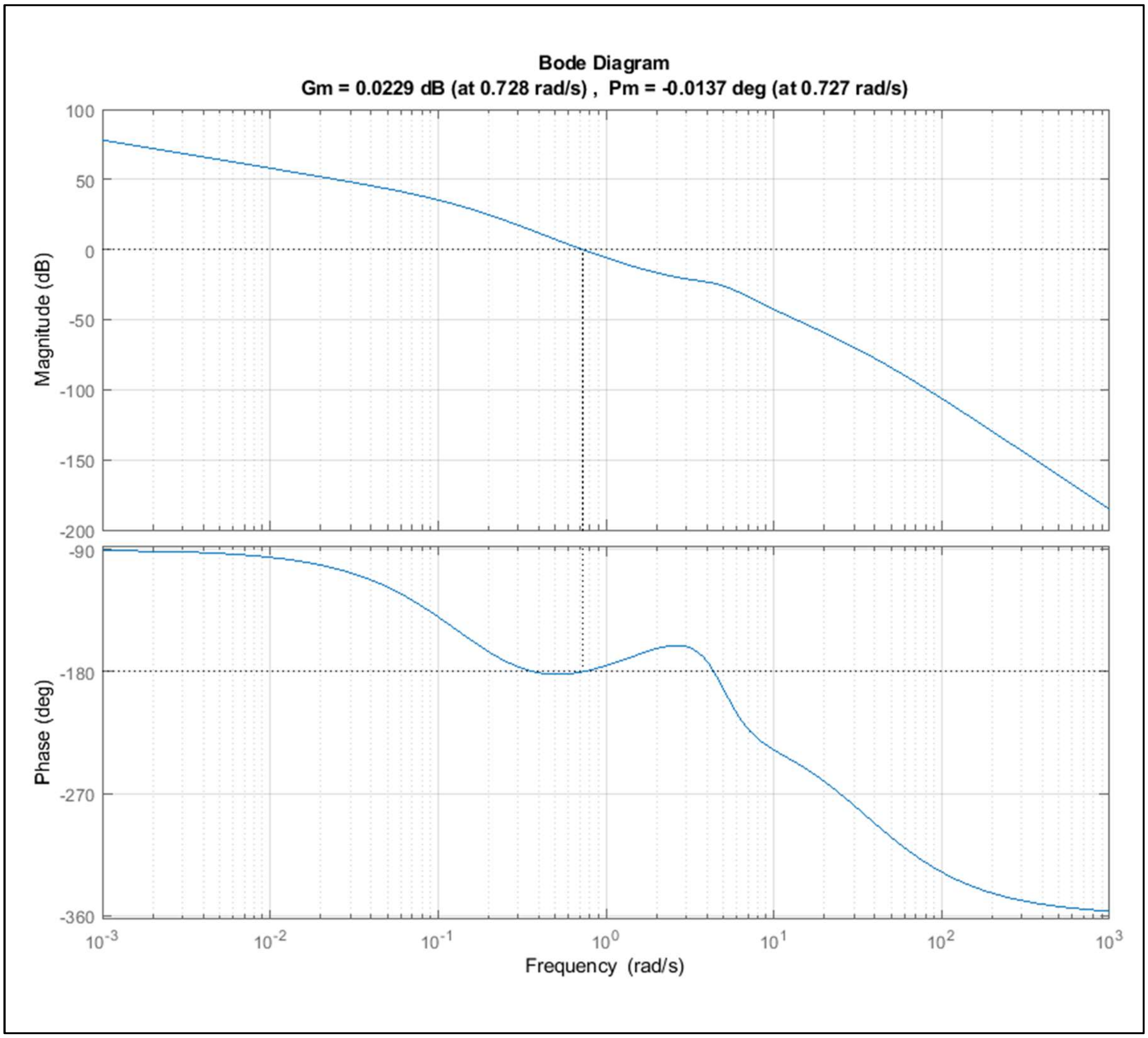
We have derived the value of sensor time delay which would cause the system instability. We will simulate the Simulink model having this sensor transfer function in the feedback path and see the output of the model.
Please refer appendix for the Simulink model.



The above figure shows that the output of the closed loop system is unstable when the sensor time delay is 8 seconds.
Therefore maximum time delay before system becomes unstable is 7 seconds.

7.5 Verification of sensor time delay for system instability

Now we will verify the sensor latency analyzed above through bode plot. For that, we can form the transfer function of sensor with 7 seconds time delay and construct the bode plot.



The above bode plot of the loop transfer function having sensor with time delay of 8 ms verifies that the phase margin drops below 0° and thus system becomes unstable.

Chapter 9. What will be in the command window after running the MATLAB script

After running the MATLAB script provided in the folder, you will see below information in the command window. It mentions the performance characteristics and based on that concludes whether controller design is good. It also mentions the sensor latency for two cases we discussed.

Command Window

```
Below are the performance characteristics of closed loop system
------------------------------------------------------------------------------
The phase margin is greater than 40 deg. Requirement is Satisfied.
------------------------------------------------------------------------------
The settling time is less than 1 second. Requirement is Satisfied.
------------------------------------------------------------------------------
The actuator force is less than 15N. Requirement is Satisfied.
------------------------------------------------------------------------------
Controller Design is meeting all the requirements.
------------------------------------------------------------------------------
The maximum delay in position sensor before performance goes out of spec is 4.000000 milliseconds.
------------------------------------------------------------------------------
The maximum delay in position sensor before system becomes unstable is 7.000000 seconds.
```

Chapter 10. Conclusion

The project walked us through the approach to design a lead and lag compensator using bode plots. It also showed us the way lead and lag compensator affect the closed loop system performance.

Lead compensators make the response faster thus compromises on the settling time requirement and also the actuator needed are of larger size because of instant pick in response.

The lag compensators make the response sluggish improving the settling time requirement and the actuators needed are of smaller size. In order to achieve optimum response characteristics, one has to choose wisely between two or both can be implemented in combination.

This project also explained the Nyquist plot interpretation about system gain and phase margin which can be an alternative to bode plot.

References

1. Canvas Notes-MEEM 4775 – Dr.Gordon Parker, Michiagn Technological University

2. *Feedback Control of Dynamic Systems* by Gene Franklin, J David-Powell, Abbas Emami-Naeini

APPENDIX A: MATLAB Code for setting up, configuring and simulating the Simulink model

Clean up

```
clearvars  % Clear the workspace
close all  % Close all open figures
clc   % Clear the command window
Simulink.sdi.clear;  % Clear the simulink data inspector
```

Simulation Parameters

```
% Solver
simPrm.solTyp = 'Fixed-step';  % Solver type
simPrm.sol    = 'ode3';  % Solver type 2
simPrm.dt  = 0.01; % Integration step size
simPrm.tEnd = 10; % Simulation end time

% Input
Input.Ts = 1;  % Start time of input step
Input.A = 1;   % Amplitude of input step

% Sensor
Tau = 0.005;  % Sensor Time delay
H = tf(1,[Tau 1]);  % Sensor Transfer function

Tau2 = 8;  % Sensor time constant for instability
H2 = tf(1,[Tau2 1]);  % Instability Sensor transfer function
```

Design Requirements

```
PM = 40;  % Lower limit of phase margin
Threshold = 5;  % Settling Time percentage threshold
```

Anonymous functions for Bode plot characteristics

```
pAng    = @(X) (1-sind(X))/(1+sind(X));
magDrop = @(X) 10*log10(X);
evalM   = @(g,X) 20*log10(abs(evalfr(g,X*1j) ) );
evalA   = @(g,X) 180/pi*angle( evalfr(g,X*1j) );
```

Error Constant and gain

```
Kc = 6;  % Set Gain as 6 more than 5
Kv = Kc*200/(25*6); % Calculate Velocity error constant
```

Original Loop Transfer function

```
g = tf(Kc*50*[1 4],[1 10 49 150 0]);% Create loop transfer function

% figure(1);margin(g);grid;  % Create bode plot of original loop transfer
function

pm0 = -30.2; wm0 = 7.04;  % Current phase margin and gain cross over frequency

pmC = (PM - pm0);  % Required change in phase margin
```

Lead & Lag Compensator Design

```
% Lead Compensator 1 Design
pmC1 = 45;  % Target phase margin for first compensator design
alpha1 = (1-sind(pmC1))/(1+sind(pmC1));  % Calculate alpha
MGDrop1 = magDrop(alpha1);  % Calculate Magnitude drop

wm1 = 9.3;  % Frequency at Magnitude drop

T1 = 1/wm1/sqrt(alpha1);

D1 = tf([T1 1],[alpha1*T1 1]); % Transfer function of first compensator

%---------------------------------------------------------------------------%
% Lead Compensator 2 Design
pmC2 = pmC-pmC1;   % Remaining phase margin for second lead compensator

alpha2 = (1-sind(pmC2))/(1+sind(pmC2));  % calculate alpha
MGDrop2 = magDrop(alpha2);  % Calculate Magnitude drop

wm2 = 11.1;  % Frequency at this magnitude drop

T2 = 1/wm2/sqrt(alpha2);

D2 = tf([T2 1],[alpha1*T2 1]); % Transfer function of second compensator

%---------------------------------------------------------------------------%
% Lag Compensator Design

w1 = 7.6; %7.6; 7.69 Frequency at required phase margin

T = 13/w1; %13/w1;

M = 7.51; %7.51; 7.29

Beta = 10^(M/20);

D3 = tf([T 1],[Beta*T 1]); %  transfer function of third compensator
```

```
D = D1*D2*D3;   % Transfer function of entire compensator design

gcl = feedback(g*D,1);  % Transfer function of closed loop system with unity
feedback
```

Open and configure the Simulink model

```
open_system('p3Sim.slx');  % Open the simulink model
set_param('p3Sim','SolverType',simPrm.solTyp);  % set this solver type in
simulink parameters
set_param('p3Sim','Solver',simPrm.sol);     % set this integration method in
simulink solver
```

Simulate the model with unity feedback

```
set_param('p3Sim/SW', 'sw', '1');  % Connect unity feedback line to switch input
SimOut1 = sim('p3Sim','SignalLoggingName','sdata');  % Simulate the model and
save results in sdata
```

Extract the results for plotting and calculation

```
%Handy constants
Out = 2;
U = 1;
In = 3;

Results.Y = SimOut1.sdata{Out}.Values.Data(:,1);
Results.U = SimOut1.sdata{U}.Values.Data(:,1);
Results.In = SimOut1.sdata{In}.Values.Data(:,1);
```

Response Characteristics

```
TSe = stepinfo(gcl,'SettlingTimeThreshold',Threshold/100).SettlingTime;  %
Settling Time
[Gm,Pm] = margin(g*D);  % Phase Margin
Umax = max(abs(Results.U)); % Maximum Actuator Force
```

Conclude the controller design

```
mycontroller(TSe,Pm, Umax,Tau,Tau2);  % Conclude my controller performance
```

Time delay in Sensor on position feedback Before performance go out of spec

```
gcl2 = feedback(g*D,H); % Transfer function of closed loop system with Sensor
```

Simulate the model with Sensor feedback

```
 set_param('p3Sim/SW', 'sw', '0');  % Connect unity feedback line to switch input
 SimOut2 = sim('p3Sim','SignalLoggingName','sdata');  % Simulate the model and
save results in sdata

 Results.U2 = SimOut2.sdata{U}.Values.Data(:,1);  % Actuator force
 Results.Y2 = SimOut2.sdata{Out}.Values.Data(:,1);  % Unstable output

 TSe2 = stepinfo(gcl2,'SettlingTimeThreshold',Threshold/100).SettlingTime;  %
Settling Time
 Umax2 = max(abs(Results.U2)); % Maximum Actuator Force
```

Plot figures

```
 %% Plot figures
 % figure(1)
 open('documents/figures/KcGL.fig');

 %figure(2)
 open('documents/figures/KcGLD.fig');

 % figure(3);
 % myplot(SimOut1.tout, Results.Y,'Step response of Closed loop system with Unity
Feedback',1,'b','Time(sec)','Position (m)')
 % hold on
 % myplot(SimOut1.tout, Results.In,'Step response of Closed loop system with Unity
Feedback',1,'k','Time(sec)','Position (m)')
 % legend('Output Response','Input Step');yticks(0:0.1:1.2);
 % hold off
 open('documents/figures/step.fig');

 % figure(4);
 % rlocus(g*D);axis equal;xlim([-45 10]);ylim([-30 30]);  % Root Locus of closed
loop system
 % title('Root Locus plot of Lead-Lag Compensator loop transfer function');
 open('documents/figures/rlocus.fig');

 % figure(5);
 % % nyquist(g*D);grid;axis equal;xlim([-3 3]);ylim([-3 3]);  % Nyquist Plot
 % % title('Nyquist plot of Lead-Lag Compensator loop transfer function');
 open('documents/figures/Nyq.fig');
 %
 % figure(6)
 % % myplot(SimOut1.tout, Results.U,'Actuator Response for Lead-Lag controller
design',1,'r','Time(sec)','Force (N)')
 open('documents/figures/Force.fig');

 % margin(g*D*H);grid;
```

```
 % title('Bode Plot of system at time delay causing out of spec performance'); %
Bode plot for verifying out of spec
 open('documents/figures/BPOutSpec.fig');

 % figure(8);
 open('documents/figures/Unstable.fig');hold off  % Bode plot of G*D1*D2*D3

 % figure(9);
 % margin(g*D*H2);grid;
 % title('Bode Plot of system at time delay causing instability');  % Bode plot
for verifying Instability
 open('documents/figures/BPUnstable.fig');
```

APPENDIX B: MATLAB function for plotting the figures and concluding the controller design

```
function myplot(x,y,ttl,LW,color,xlbl,ylbl)
plot(x,y,color,'LineWidth',LW);
xticks(0:1:10);  % Give the x axis ticks
title(ttl); % Give the title.
xlabel(xlbl);
ylabel(ylbl);
end

function mycontroller(Tse,PMargin, Umax,Tau,Tau2)
fprintf('<strong>Below are the performance characteristics of closed loop
system</strong>\n');
if Tse < 1 && PMargin > 40 && Umax < 15
    fprintf('<strong>------------------------------------------------------------
------------------</strong>\n');
    fprintf('The phase margin is greater than 40 deg. Requirement is
Satisfied.\n');
    fprintf('<strong>------------------------------------------------------------
------------------</strong>\n');
    fprintf('The settling time is less than 1 second. Requirement is
Satisfied.\n');
    fprintf('<strong>------------------------------------------------------------
------------------</strong>\n');
    fprintf('The actuator force is less than 15N. Requirement is Satisfied.\n');
    fprintf('<strong>------------------------------------------------------------
------------------</strong>\n');
    fprintf('<strong>Controller Design is meeting all the
requirements.</strong>\n');
    fprintf('<strong>------------------------------------------------------------
------------------</strong>\n');
    fprintf('<strong>The maximum delay in position sensor before performance goes
out of spec is %f milliseconds.</strong>\n',(Tau*1000)-1);
    fprintf('<strong>------------------------------------------------------------
------------------</strong>\n');
    fprintf('<strong>The maximum delay in position sensor before system becomes
unstable is %f seconds.</strong>\n',Tau2-1);
else
    fprintf('<strong>------------------------------------------------------------
------------------</strong>\n');
    fprintf(2,'<strong>One of the requirements is not satisfied.Please redesign
the controller.</strong>\n');
end
end
```

APPENDIX C: Simulink Model