Project: Design Lag-Lead and PID Controllers

Abstract

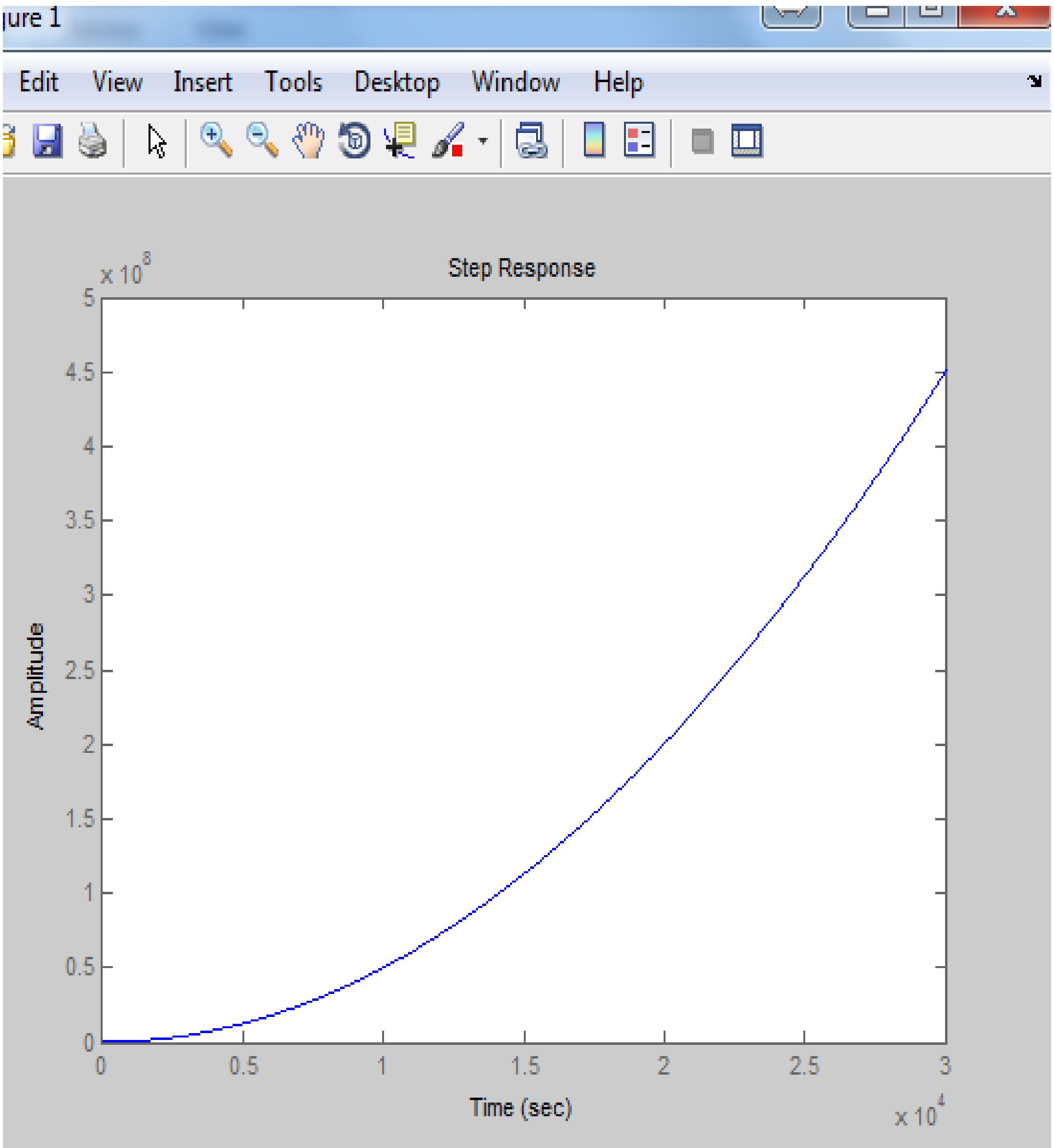
The main objective of this design project is to help students to design different types of controllers such as lag-lead and PID controllers. First of all, we start designing these two controllers by hand using several steps with checking the steady-state error for each controller. Moreover, we verify our design results using MATLAB, SIMULINK and SISOTOOL Design in order to plot the root locus for the open loop system and to plot the step response as well. Ultimately, we apply what we have learned from the lectures in designing our lead-lag and PID controllers for this project.

Table of Contents

MATLAB and SIMULINK Simulations:-

Part I

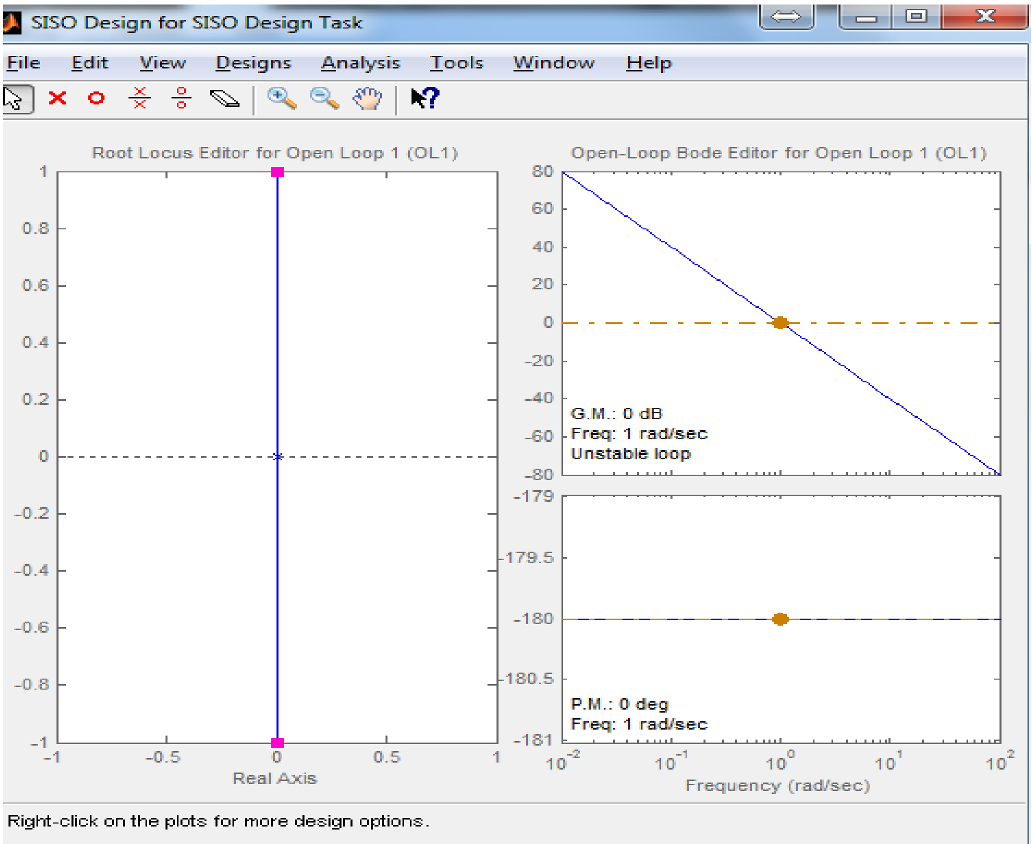
```
>> num  = [1];
>> den = [1 0 0];
>> OL = tf (num,den)
Transfer function:
 1
---
s^2

>> step(OL)
```

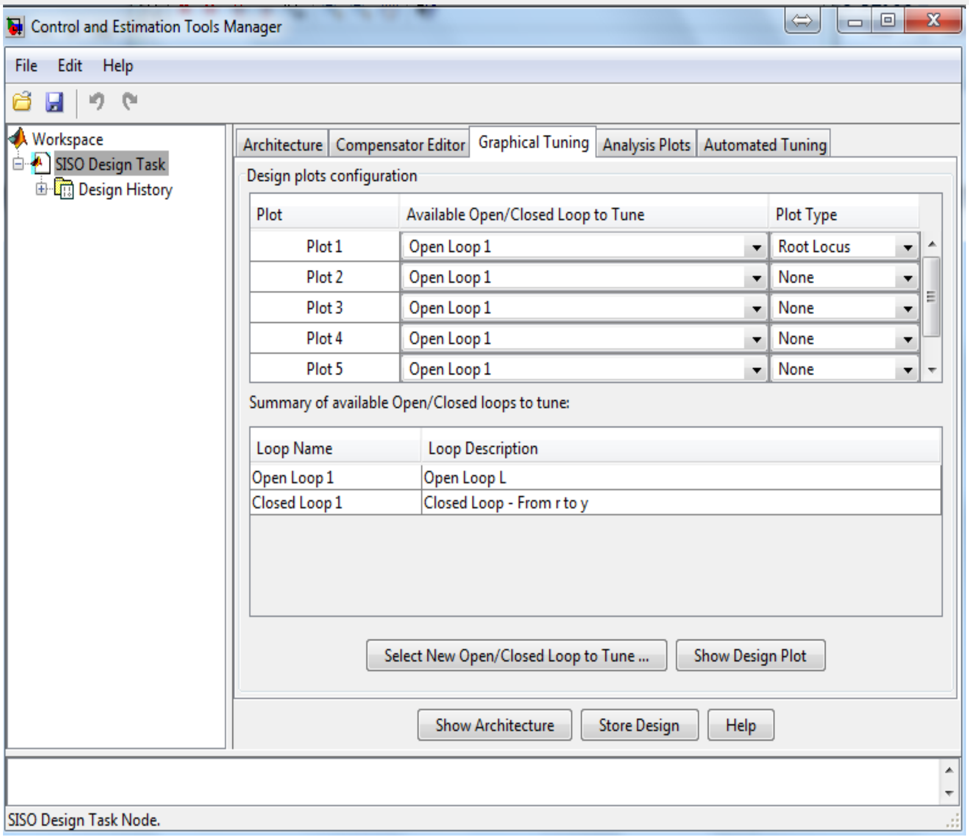


Sisotool Design:

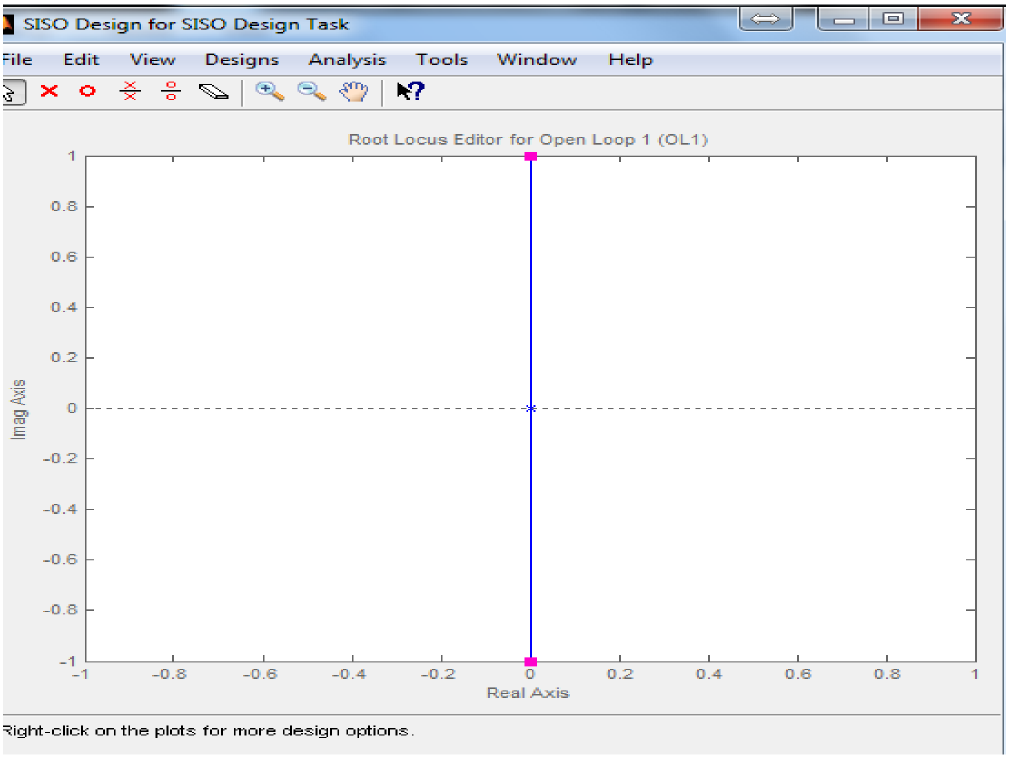
\>> sisotool(OL)

%OL is the open loop transfer function

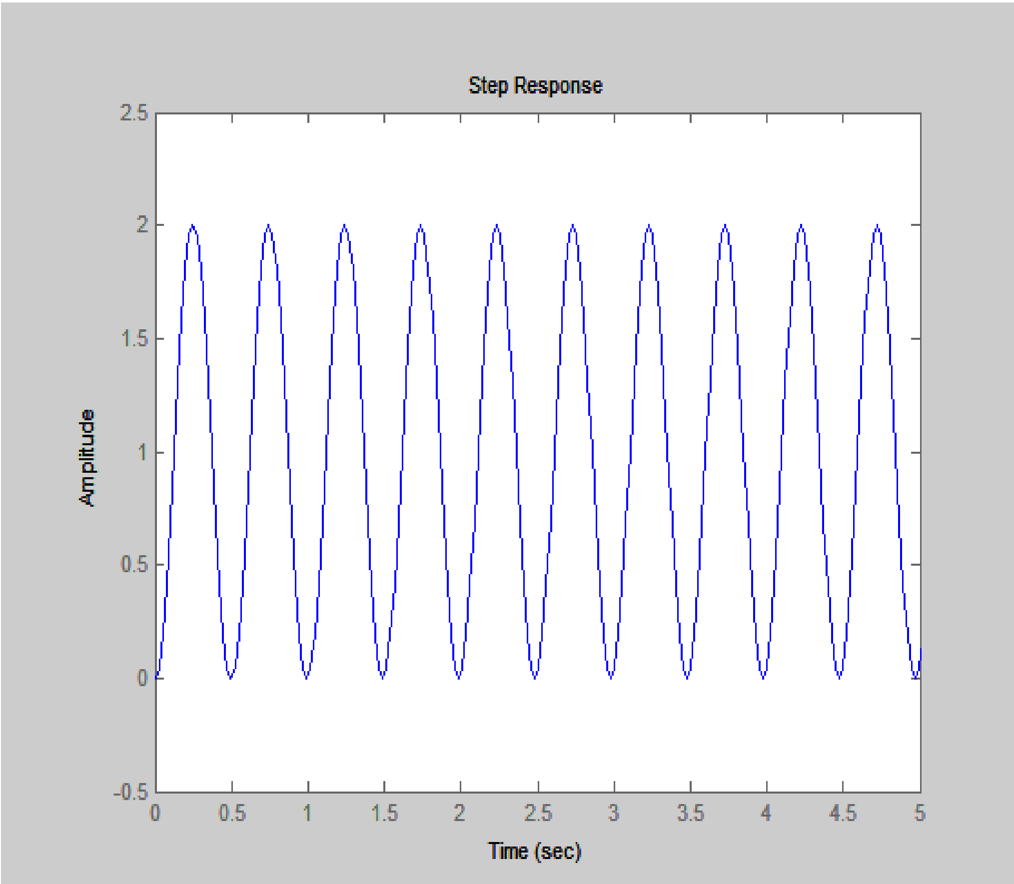


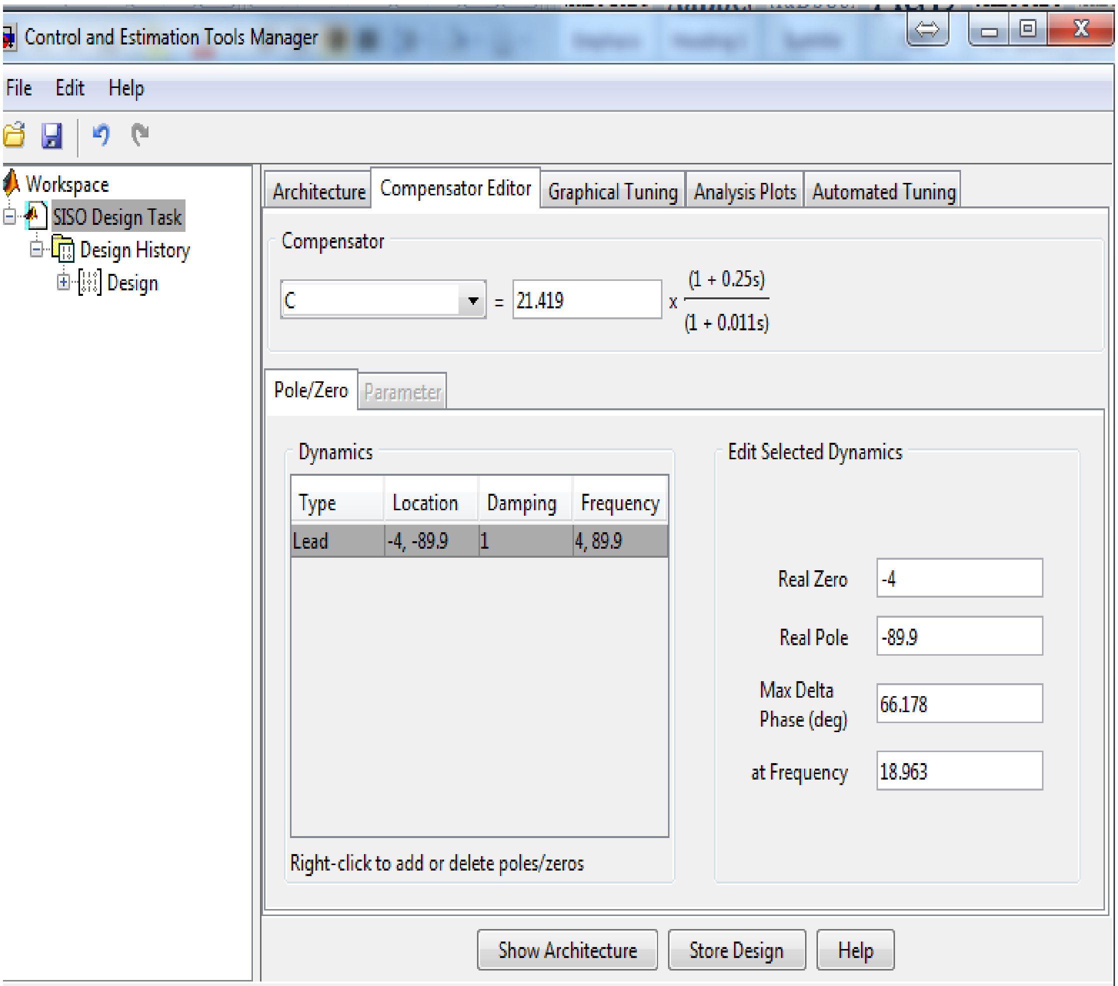


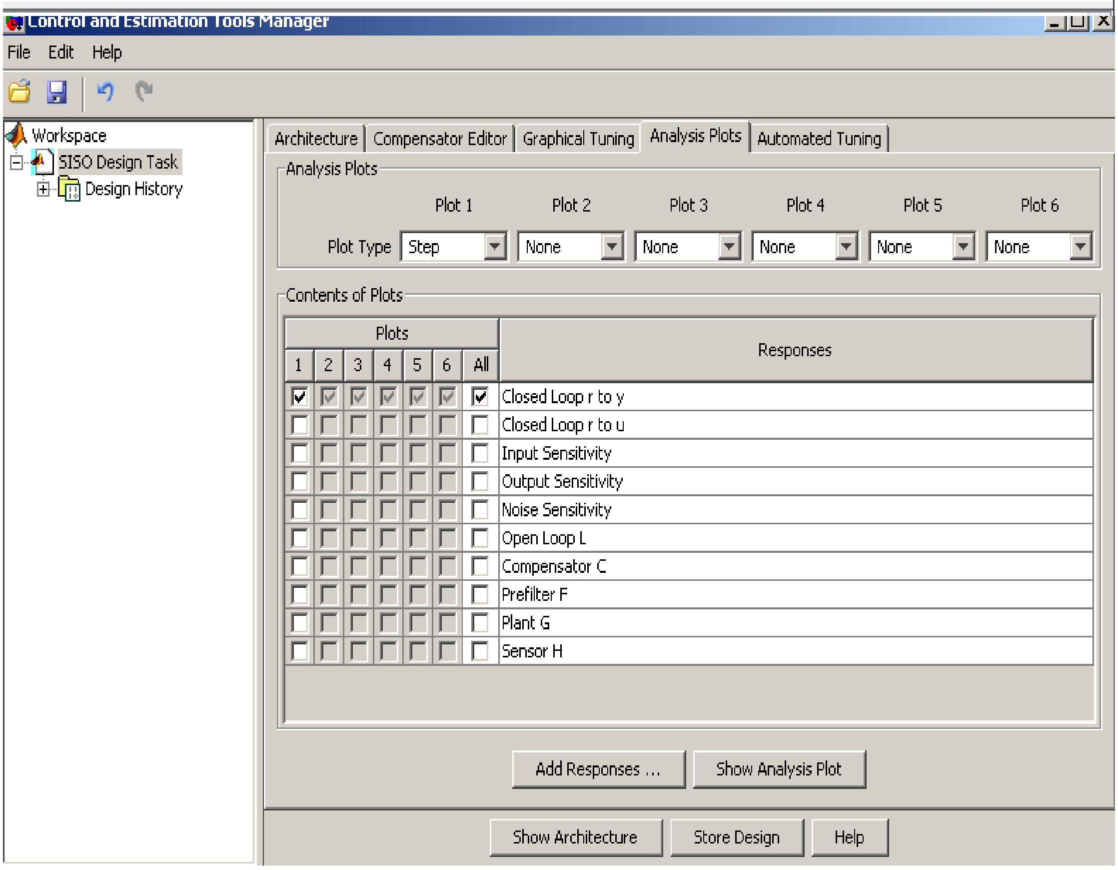
The root locus for the open loop system is shown:



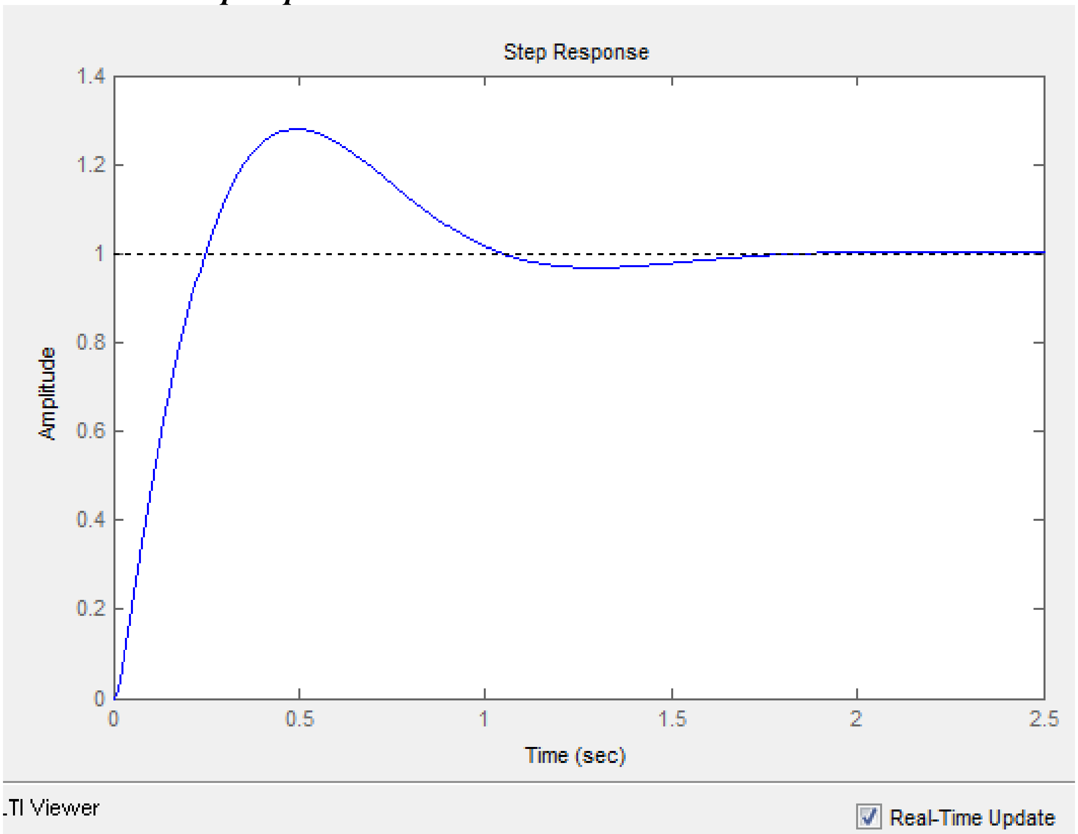
Step response of the original system:

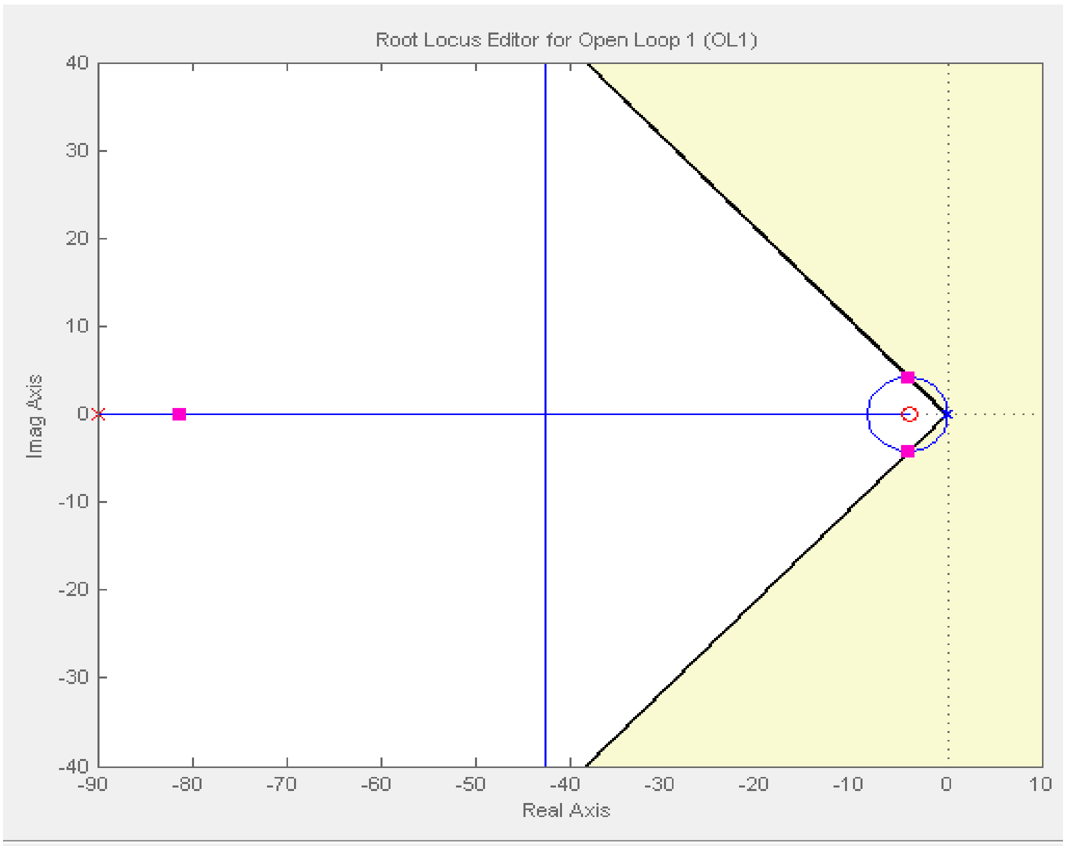


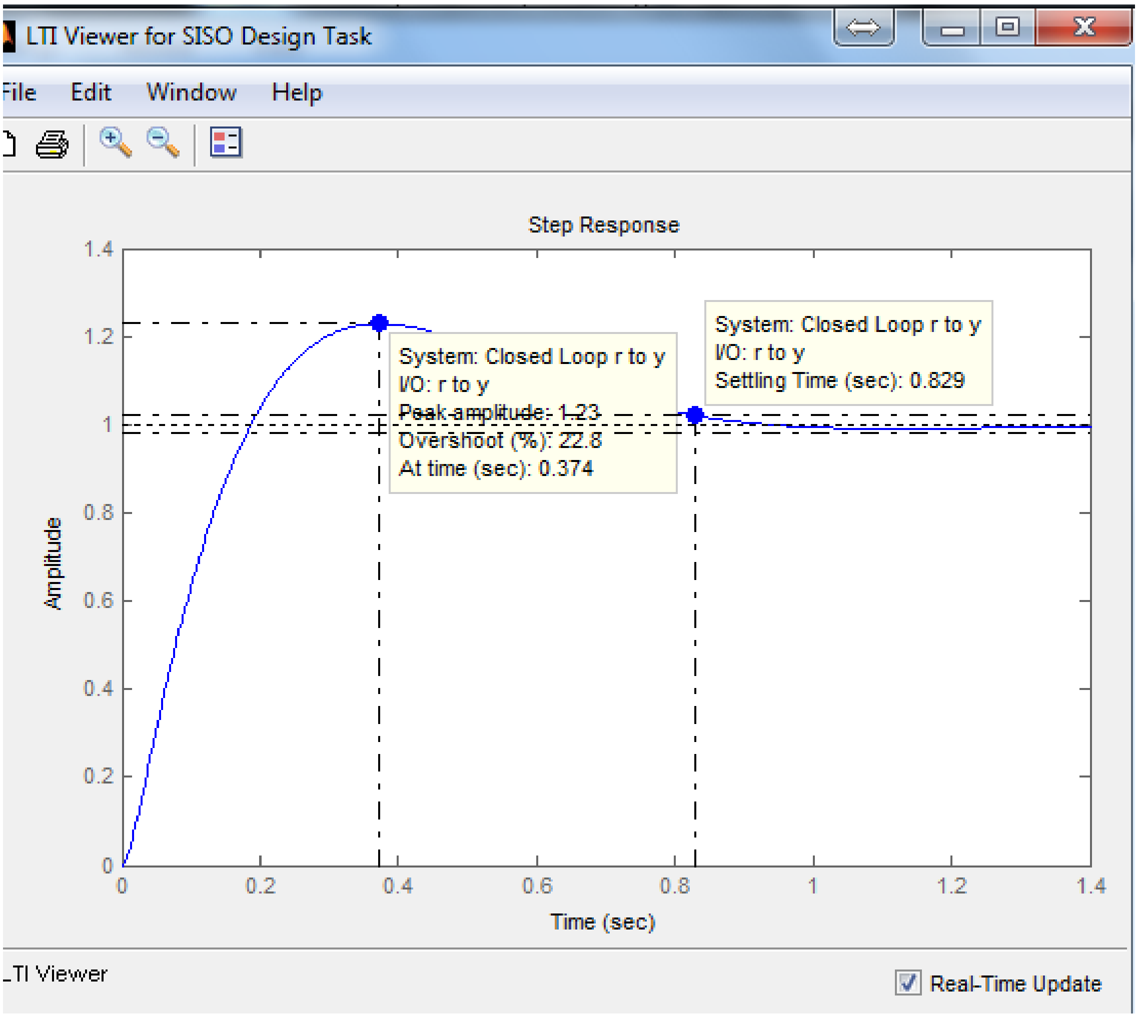

Control and Estimation Tools Manager
File Edit Help
Workspace
SISO Design Task
Design History
Design
Architecture
Compensator Editor
Graphical Tuning
Analysis Plots
Automated Tuning
Compensator
C = 21.419 x (1 + 0.25s) / (1 + 0.011s)
Pole/Zero
Parameter
Dynamics
Type Location Damping Frequency
Lead -4, -89.9 1 4, 89.9
Right-click to add or delete poles/zeros
Edit Selected Dynamics
Real Zero -4
Real Pole -89.9
Max Delta Phase (deg) 66.178
at Frequency 18.963
Show Architecture
Store Design
Help


Control and Estimation Tools Manager
File Edit Help
Workspace
SISO Design Task
Design History
Architecture
Compensator Editor
Graphical Tuning
Analysis Plots
Automated Tuning
Analysis Plots
Plot 1 Plot 2 Plot 3 Plot 4 Plot 5 Plot 6
Plot Type Step None None None None None
Contents of Plots
Plots
1 2 3 4 5 6 All
Responses
Closed Loop r to y
Closed Loop r to u
Input Sensitivity
Output Sensitivity
Noise Sensitivity
Open Loop L
Compensator C
Prefilter F
Plant G
Sensor H
Add Responses ...
Show Analysis Plot
Show Architecture
Store Design
Help

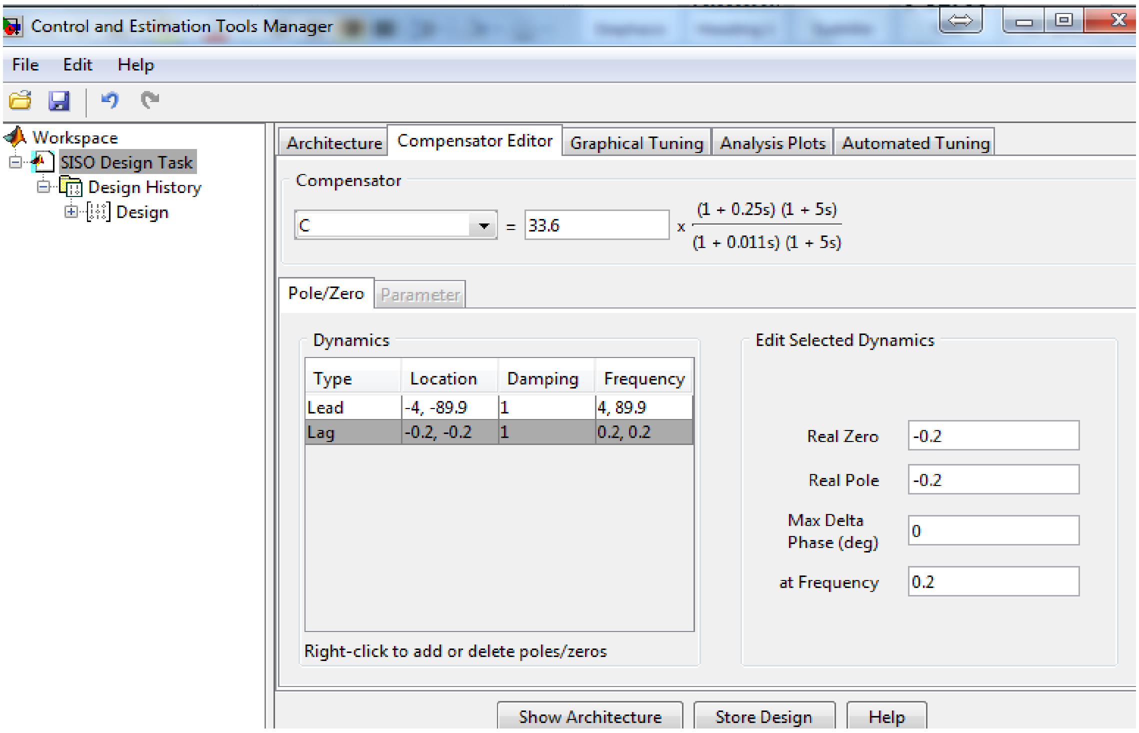
This shows the step response:

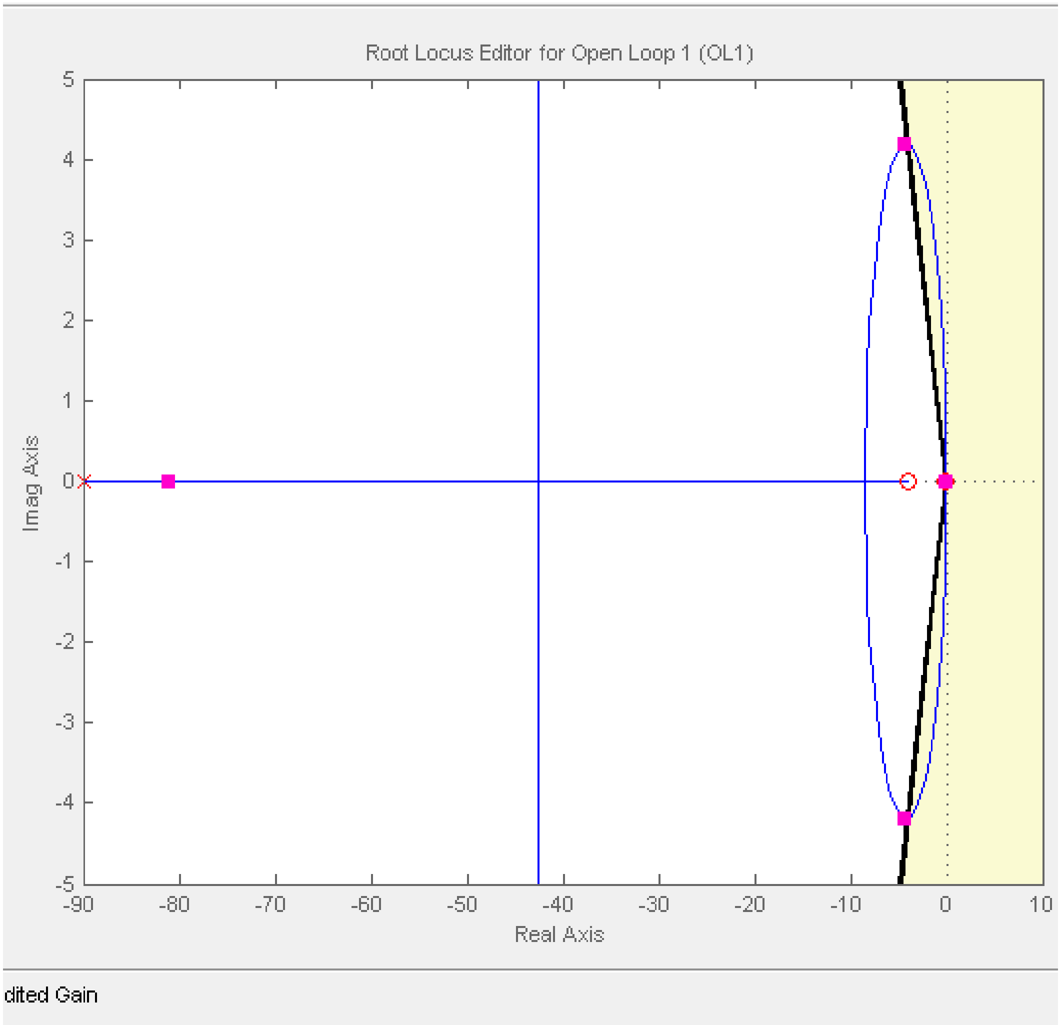


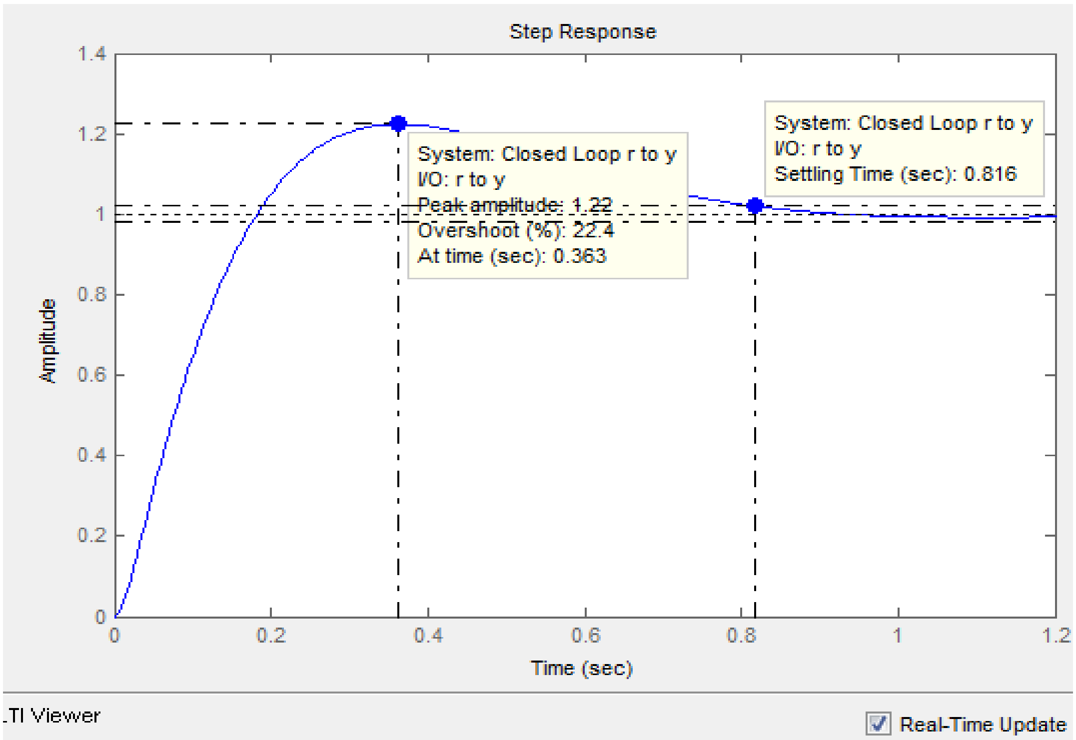




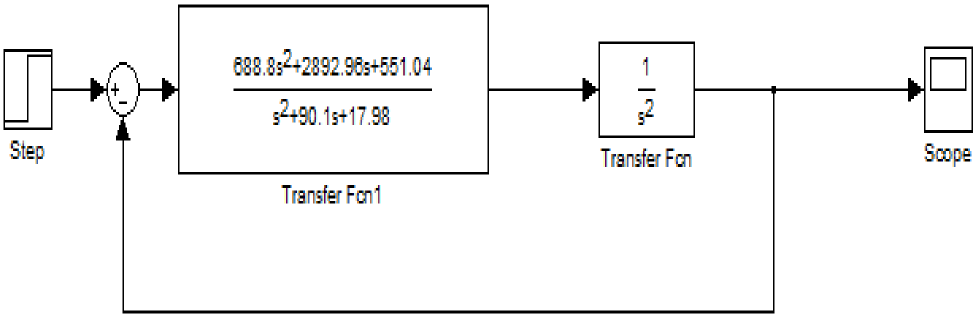
Adding Lag-Controller to the previous design:

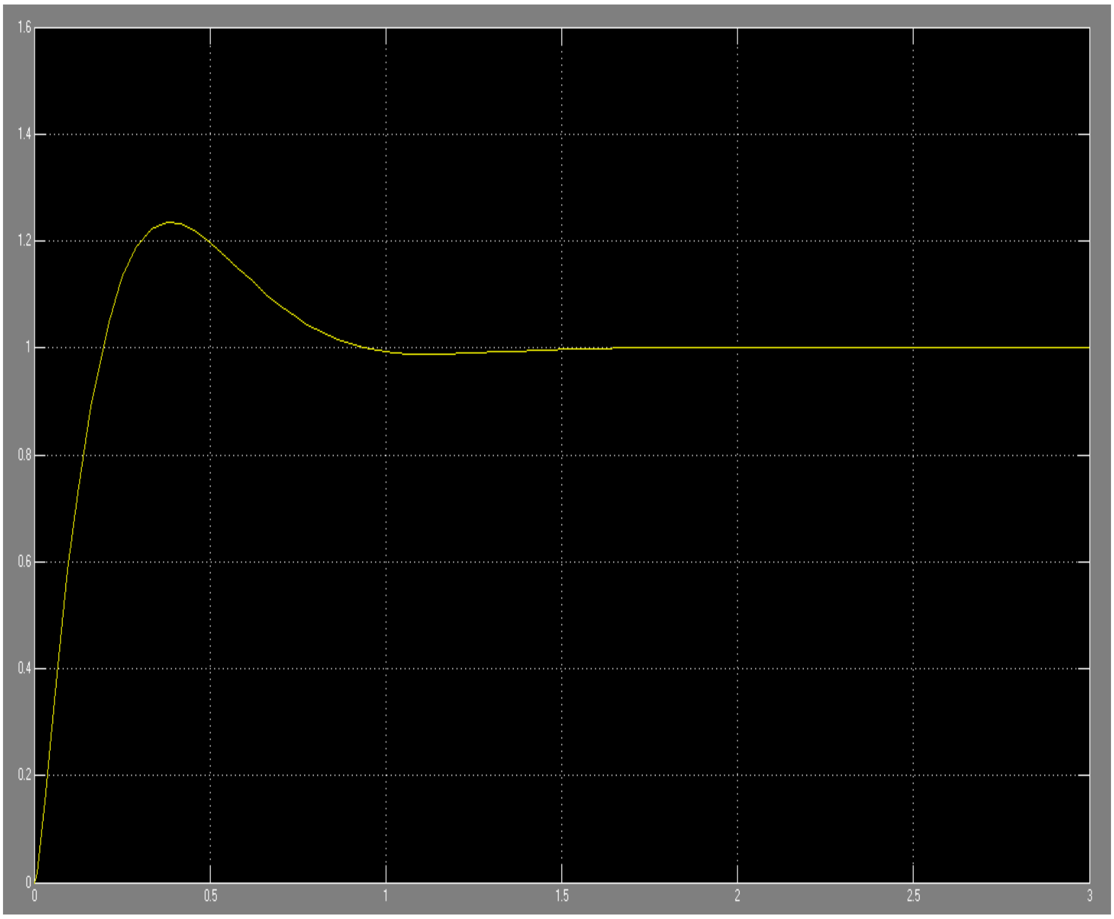



Root Locus Editor for Open Loop 1 (OL1)
Imag Axis
Real Axis
5
4
3
2
1
0
-1
-2
-3
-4
-5
-90
-80
-70
-60
-50
-40
-30
-20
-10
0
10
dited Gain


Step Response
Amplitude
Time (sec)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0.2
0.4
0.6
0.8
1.2
System: Closed Loop r to y
I/O: r to y
Peak amplitude: 1.22
Overshoot (%): 22.4
At time (sec): 0.363
System: Closed Loop r to y
I/O: r to y
Settling Time (sec): 0.816
.TI Viewer
Real-Time Update

SIMULINK:


688.8s2+2892.96s+551.04
s2+90.1s+17.98
1
s2
Step
Transfer Fcn1
Transfer Fcn
Scope


1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0.5
1.5
2
2.5
3

Explanation of the results:

The known is the closer you get to the real axis the smaller the overshoot you get. However, in this case the T.F function of the system is not the same as the general form : $\frac{wn^2}{s^2+z\,wn\,s+wn^2}$, due to the existence of a zero in the target region which increases the percent overshoot. Thus, getting a 5% overshoot will be extremely hard.

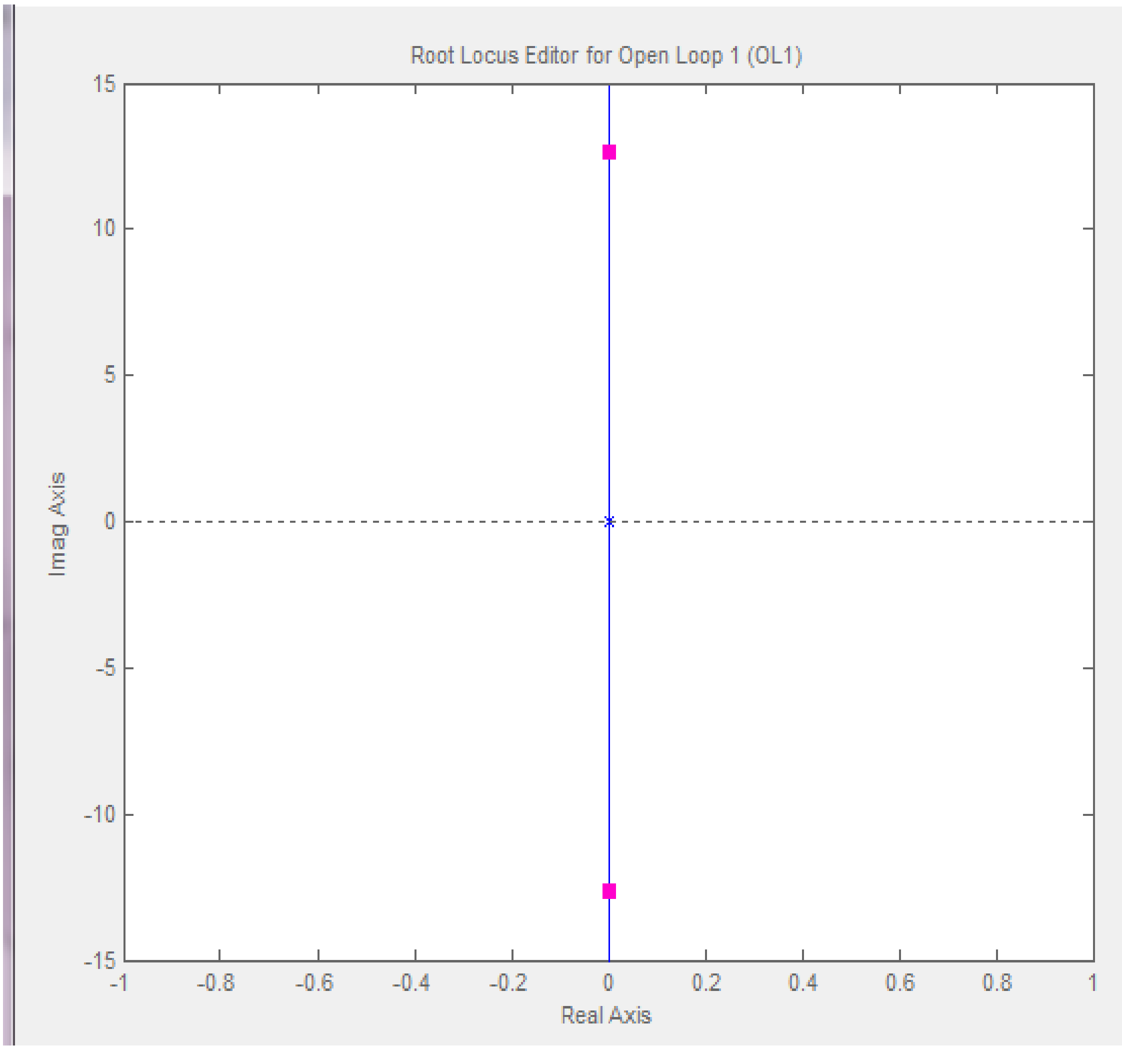
Effect of Disturbance on the System:

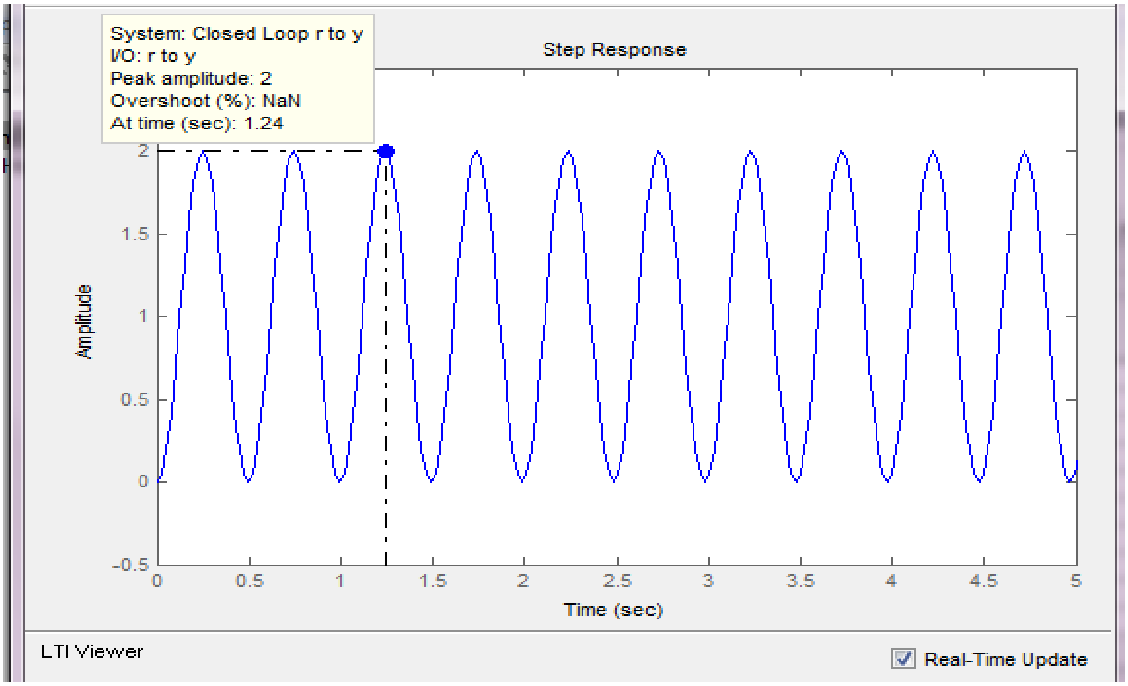
If step disturbance is added to the system after the Lead-Lag controller, it will affect the response of the system by an extremely small amount. Hence, it can be concluded that the Lead-Lag controller can resist the added step disturbance but will not completely eliminate its effect.

Part II

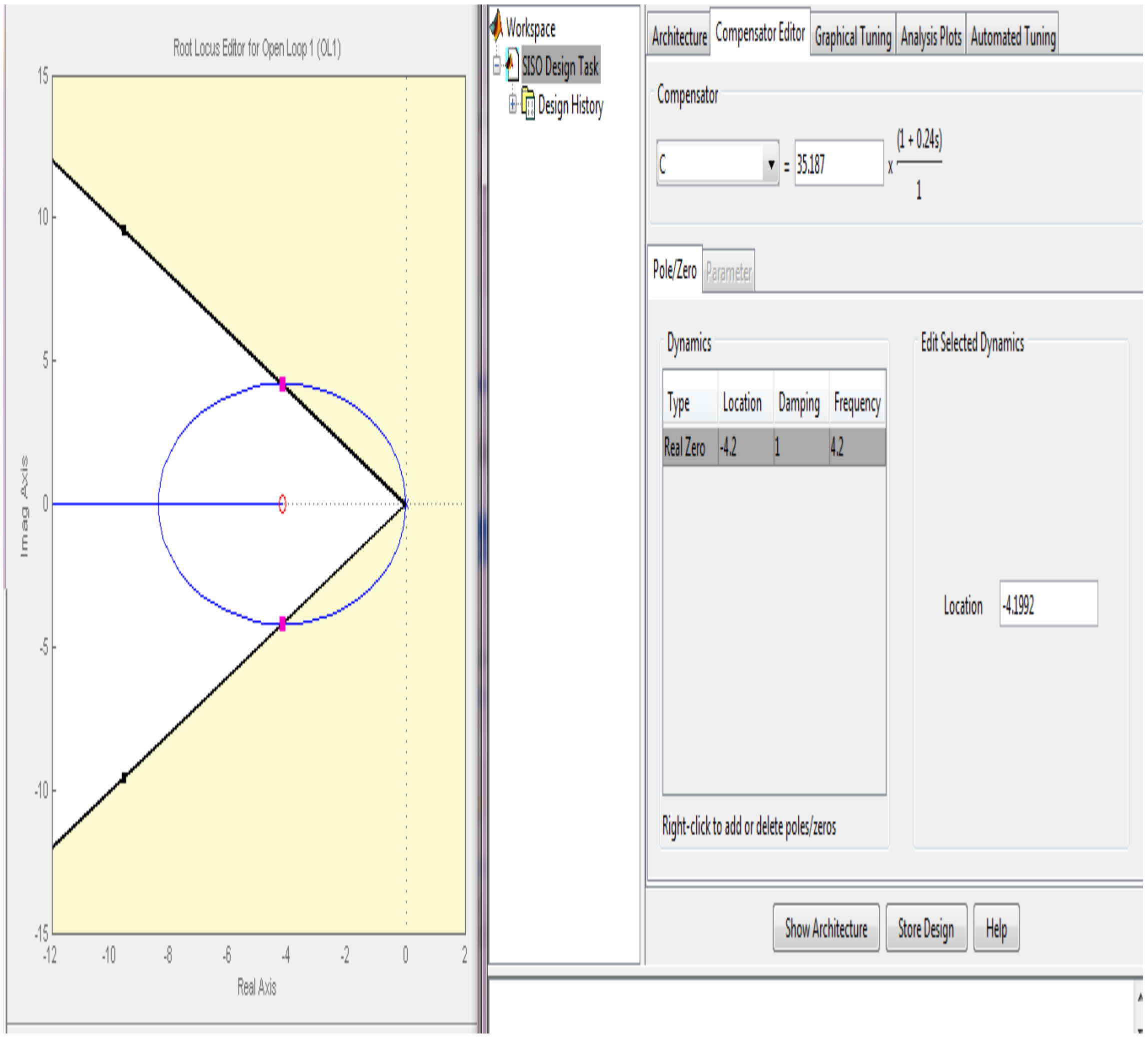
```
>> num=[1];
den=[1 0 0];
OL=tf(num,den)

Transfer function:
1
---
s^2

>>sisotool(OL)
```

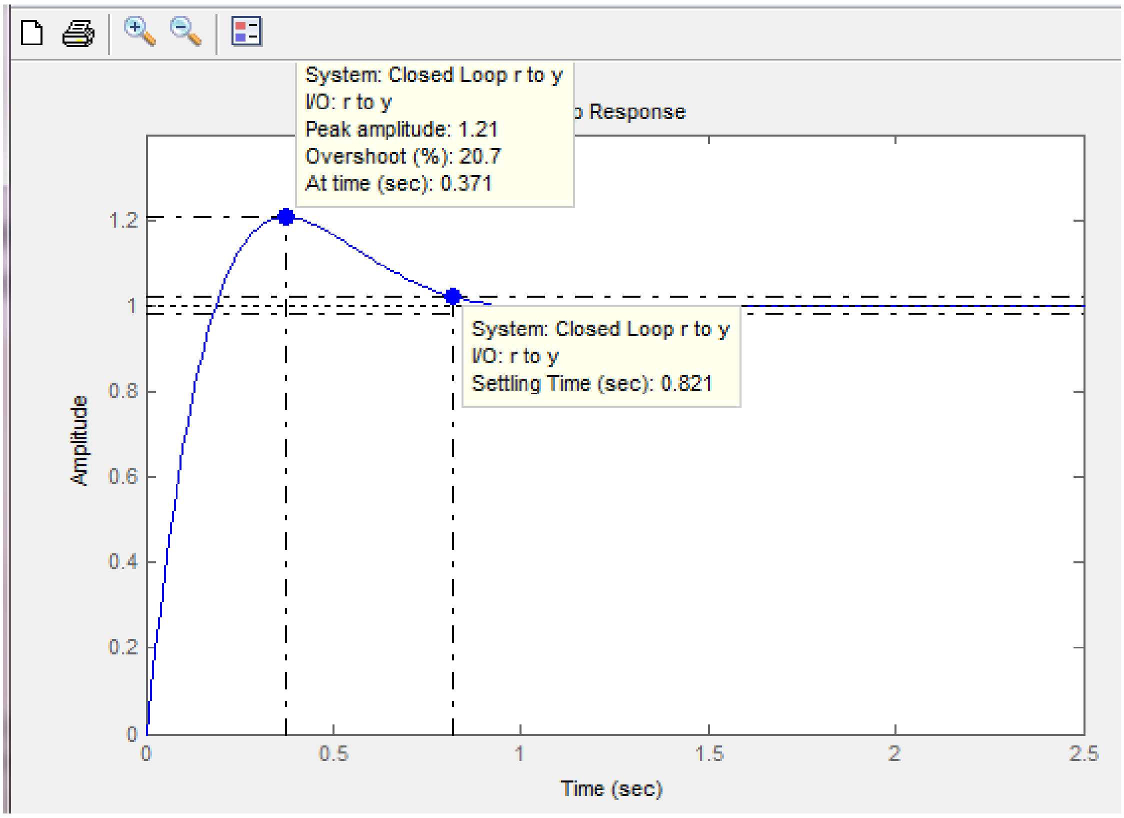
Root Locus for original system



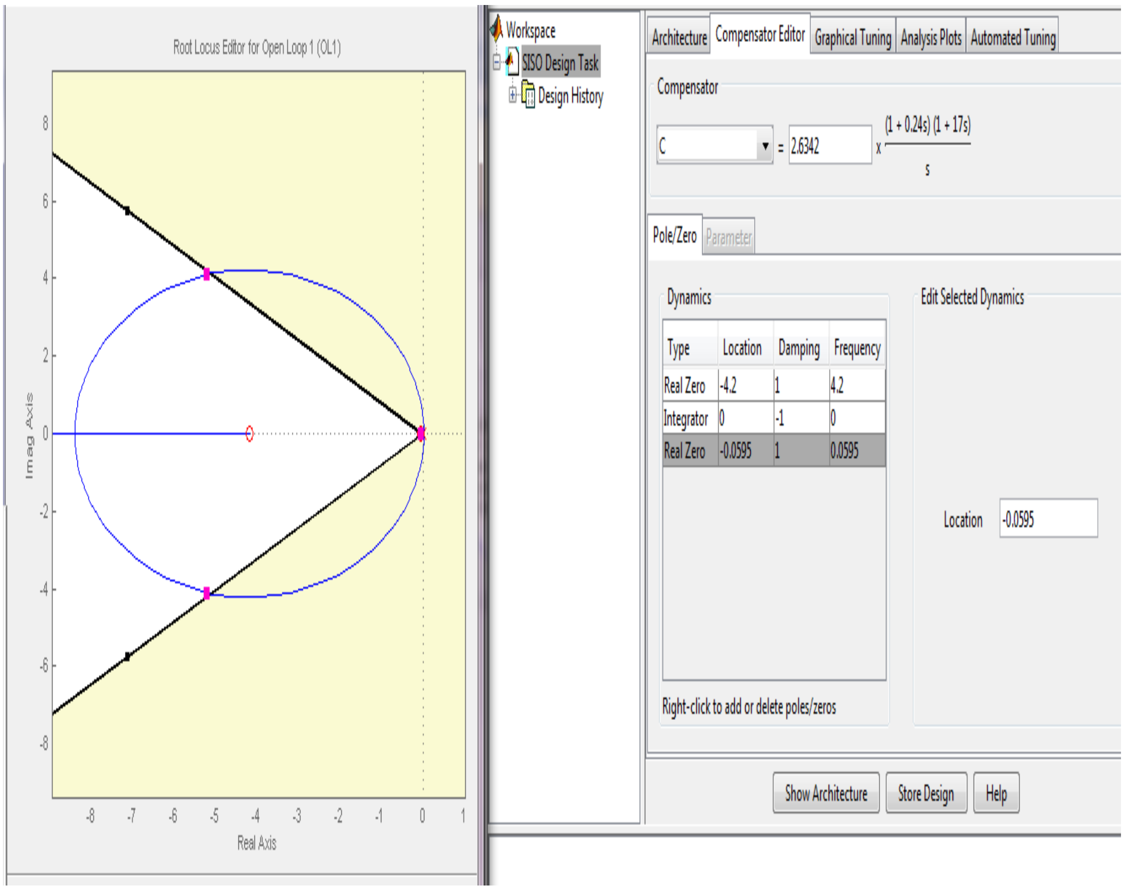


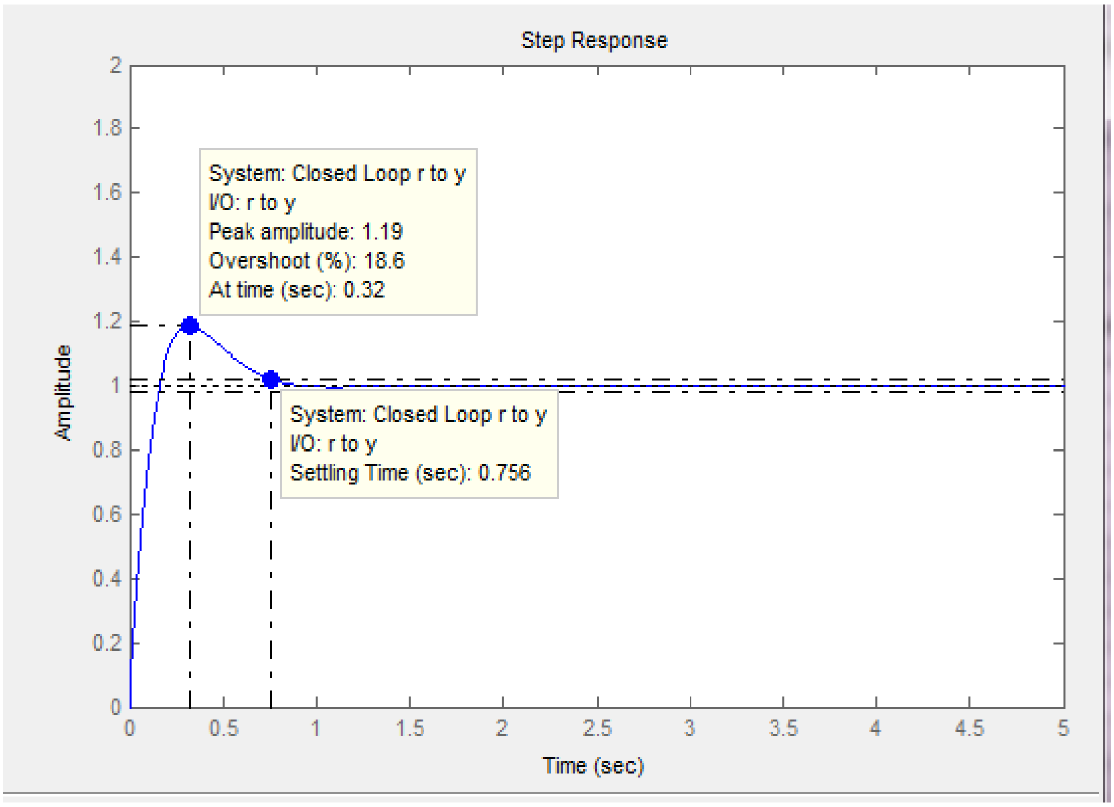
Root Locus after adding a PD controller with the design requirements



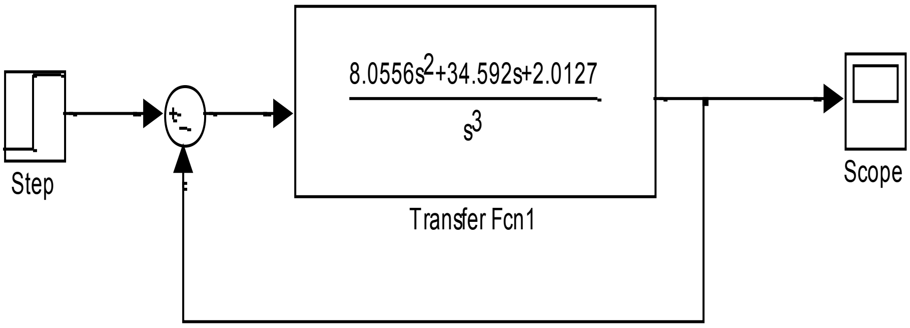


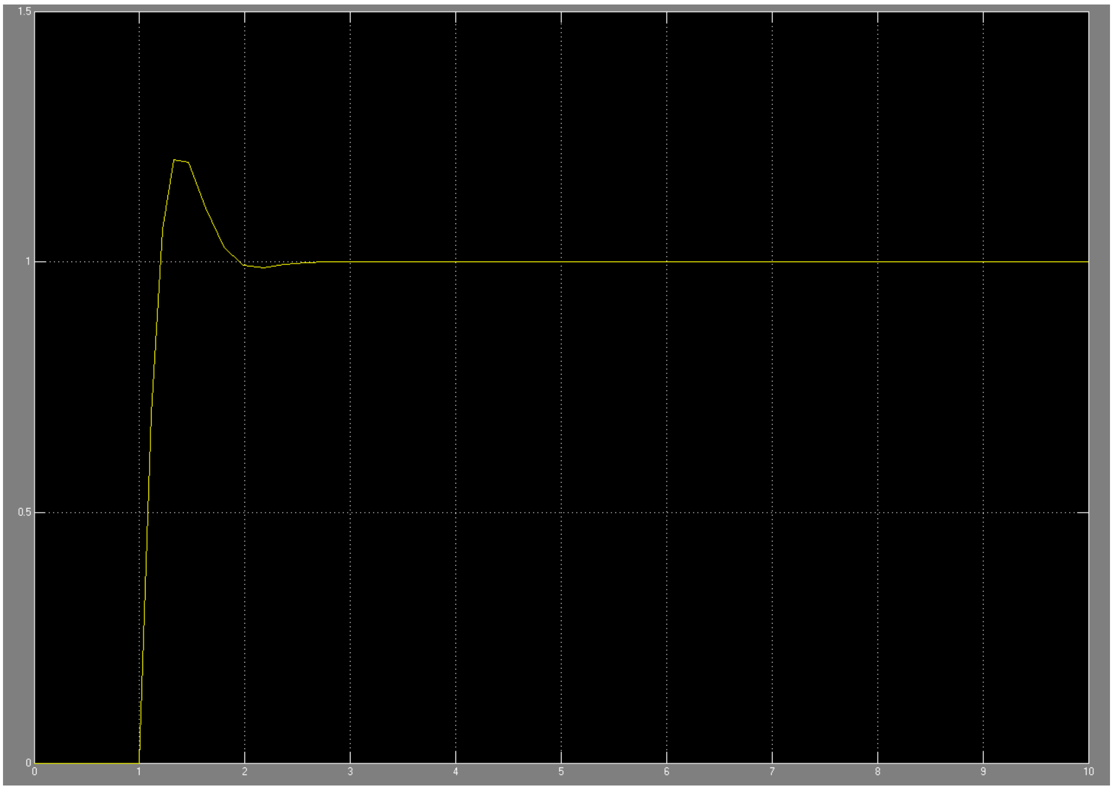
Add a PI controller




Step Response
System: Closed Loop r to y
I/O: r to y
Peak amplitude: 1.19
Overshoot (%): 18.6
At time (sec): 0.32
System: Closed Loop r to y
I/O: r to y
Settling Time (sec): 0.756
Amplitude
Time (sec)
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

SIMULINK:


8.0556s^2+34.592s+2.0127
s^3
Step
Transfer Fcn1
Scope
+
-



Explanation of the results:

The known is the closer you get to the real axis the smaller the overshoot you get. However, in this case the T.F function of the system is not the same as the general form: $\frac{wn^2}{s^2+z\,wn\,s+wn^2}$, due to the existence of a zero in the target region which increases the percent overshoot. Thus, getting a 5% overshoot will be extremely hard.

Effect of Disturbance on the System:

If step disturbance is added to the system after the Lead-Lag controller, it will affect the response of the system by an extremely small amount. Hence, it can be concluded that the PID controller can resist the added step disturbance but will not completely eliminate its effect.

Conclusion

Overall, after completing our project in Control Systems I (ELE353), we were able to utilize MATLAB, SIMULINK and SISOTOOL DESIGN programming to analyze and design different types of controllers such as Lag-Lead and PID controllers. Furthermore, we were able also to design by hand different types of controllers such as Lag-Lead and PID controllers. Moreover, in this project, we applied the team skills where we divided the work between ourselves in order to do the required task in fastest possible time. Ultimately, this project is one of the most important applications in the control systems I area where we can applied our gained skills from this project in designing and analyzing any controller in the future.