Introduction:

The purpose of this project is to apply the design and analysis methods in controls engineering, including all phases of modeling, implementation and testing. Furthermore I modeled everything in Matlab to showcase my experience with the powerful software package. Here I used two control designs, one in time domain and the other in the frequency domain. After the design aspect I tested my calculations using the Servo trainer. This is to show an understanding of control theory and how systems could react in industry.

Theory:

Part 1

For this project the plant (system) transfer function was given and I designed the open and closed loop functions when Gc=1. I then analyzed the time-domain behavior of the uncompensated system by plotting the step response and determining the transient and steady-state properties. Next I found the damping ratio, natural and undamped frequency taking into consideration the overall properties of the system, checking the overall stability, steady state error, overshoot and rise time using Matlab and Simulink.

Part 2

In the second part I used a given (pre-determined) damping ratio and natural frequency (assuming steady state error is zero) and then created a gain compensation Gc(s) = K and used an RH test determine the acceptable range of K for my closed loop stability. I did this because in many cases in industry certain aspects of a system might be known and other parameters might be needed. I compared the behavior of the gain-compensated system with that of the uncompensated system (from Part 1). I designed a first-order controller of the form $Gc(s) = Kc (1+\tau 1s)/(1+\tau 2s)$ where Kc, τ1 and τ2 are free parameters (to emulate a design process of a system).

Part 3

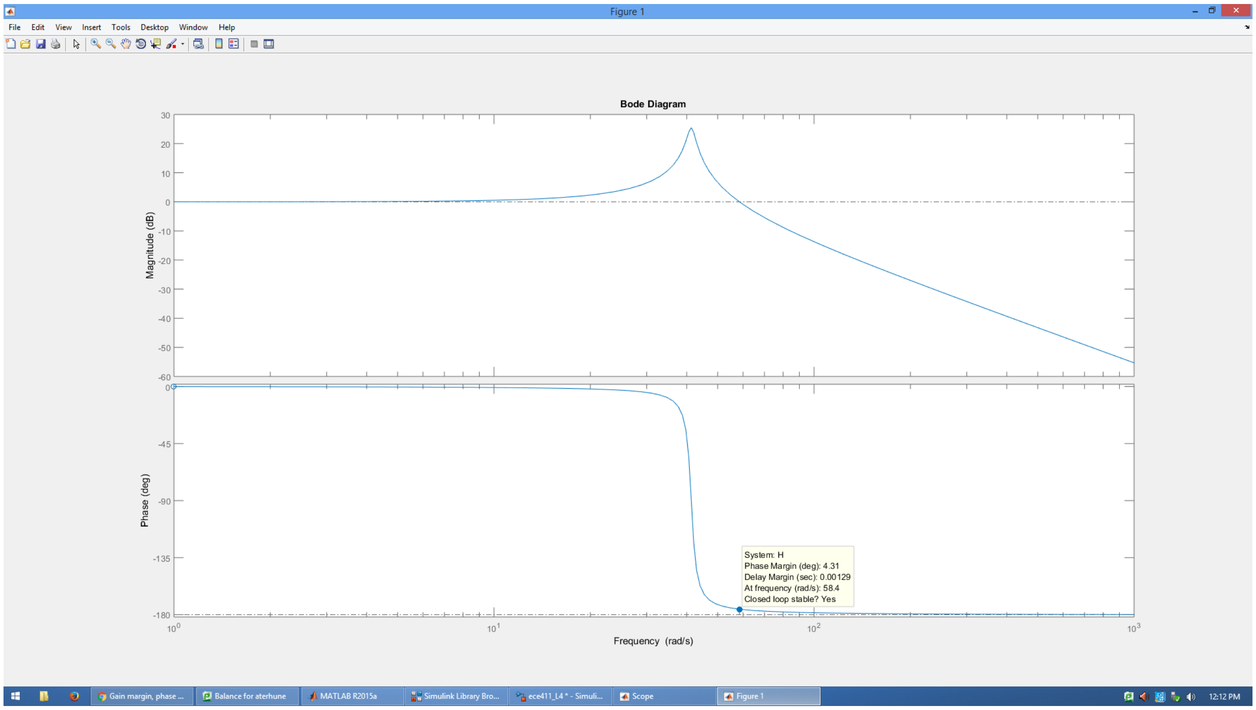
For the third part I began with a simple gain compensation Gc(s) = K and used the RH test to determine the range of K for closed loop stability. After choosing the gain (K) I generated the Bode Plots (see Results and Discussion) and step response of the gain-compensated system and calculated GM and PM. All modeling was done in Simulink.

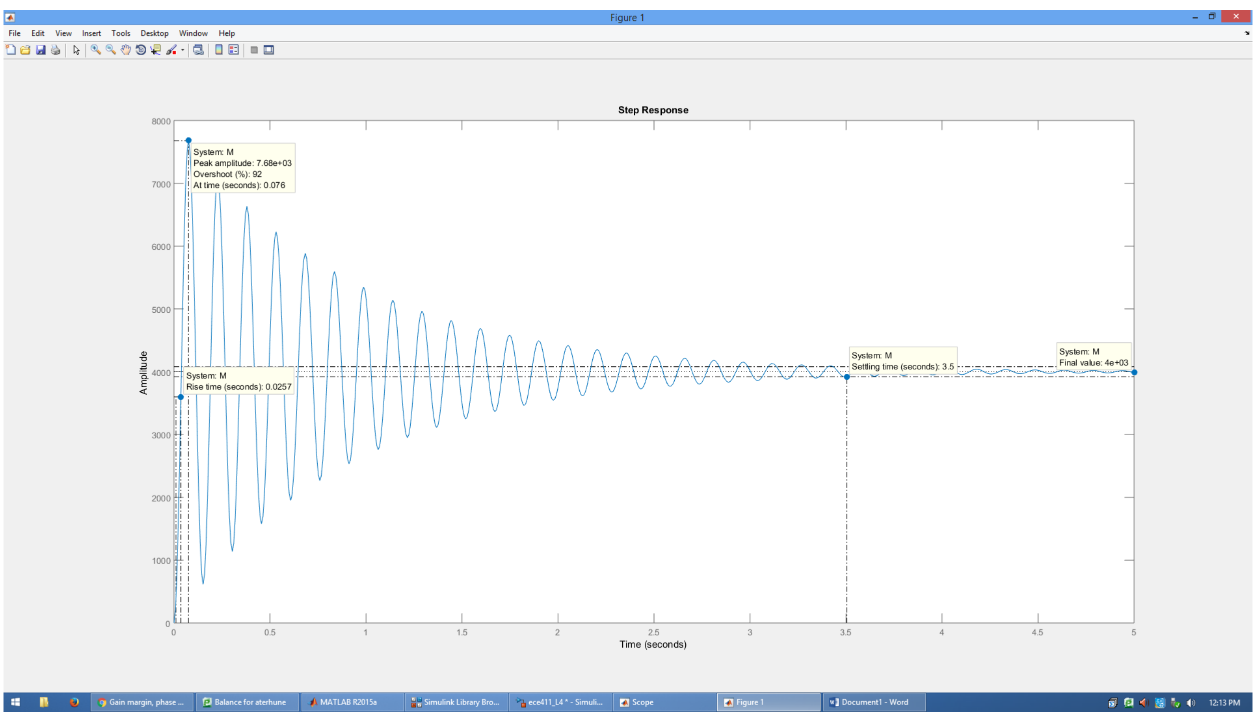
Part 4

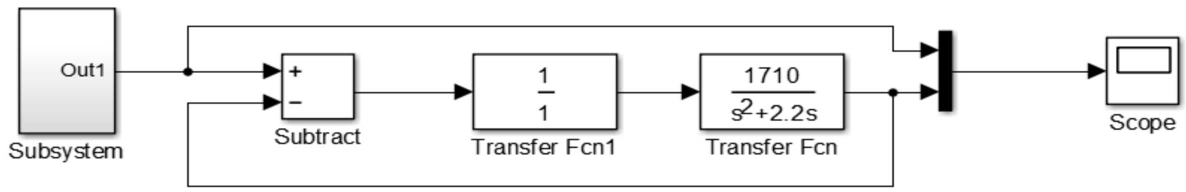
For the fourth and final part of this project I tested both controllers designed on Simulink on a Model 220 Servo Trainer. This step involved using a piece of equiptment and graphing the 'real world' results and then simply comparing them to the CAD design, to show theory versus practicality.

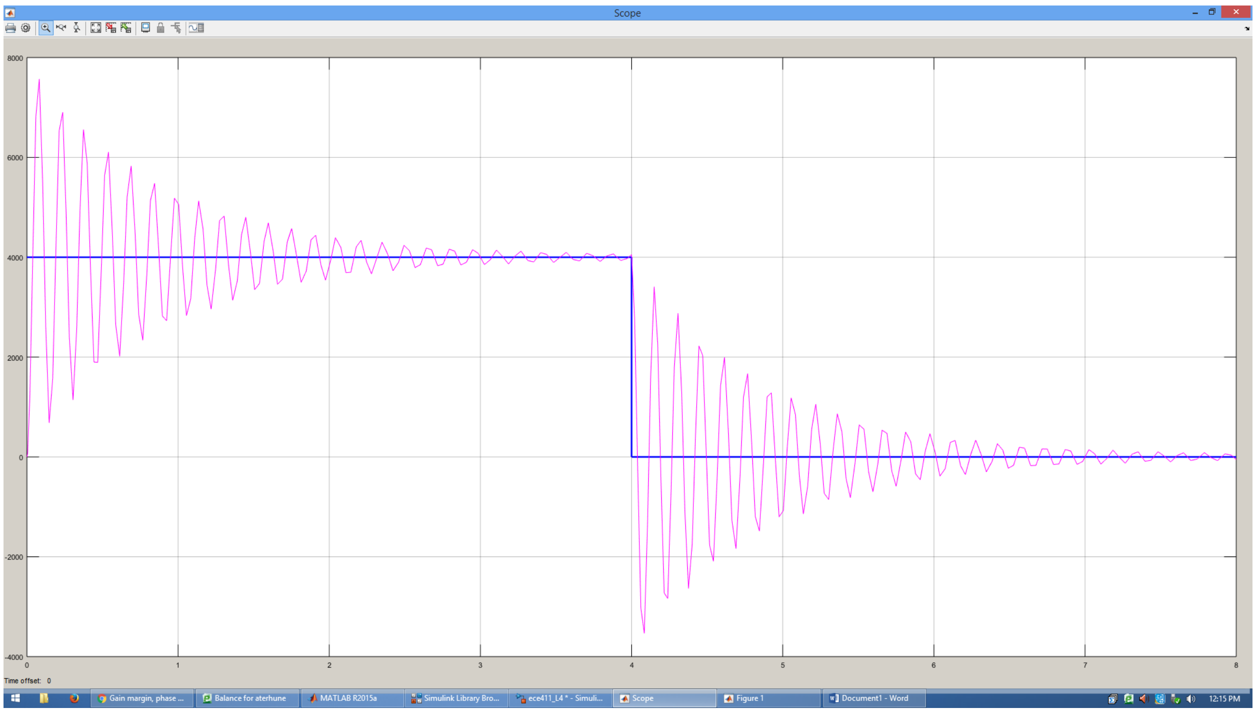
Results and Discussion:

Part 1









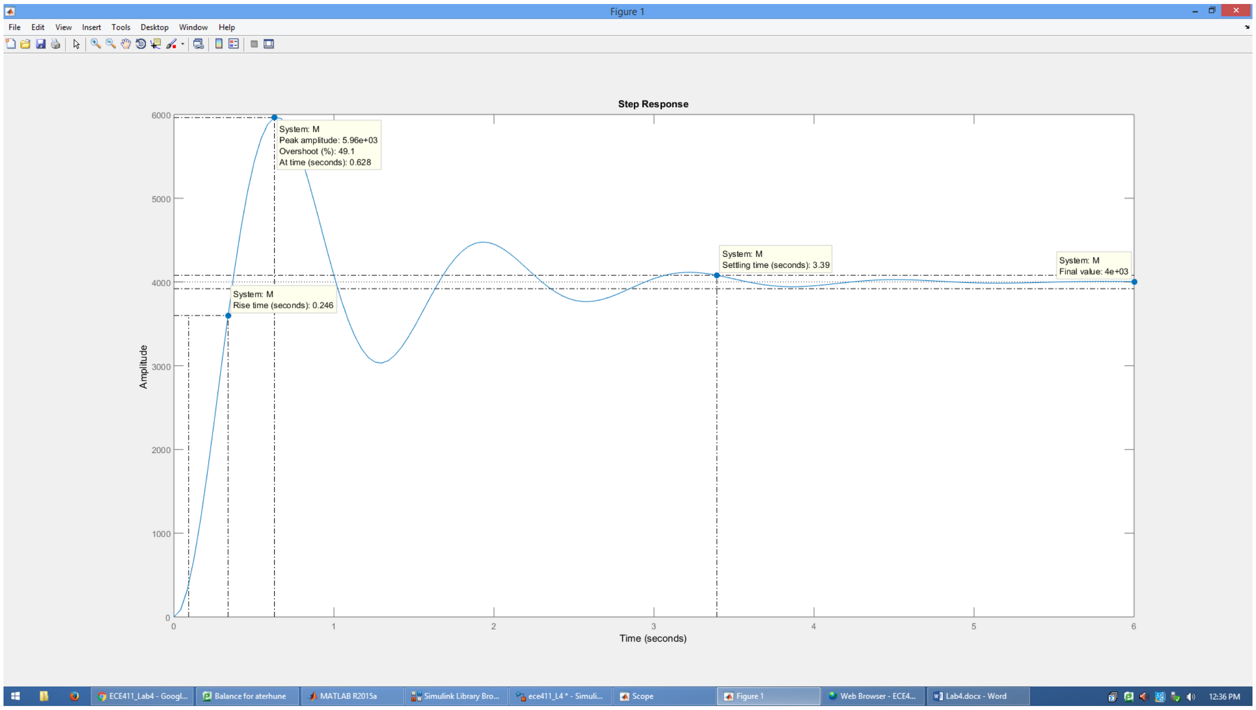

```
H = tf(1710,[1 2.2 1710]);
M=4000*H;
[Gm,Pm,Wgm,Wpm] = margin(H)
```

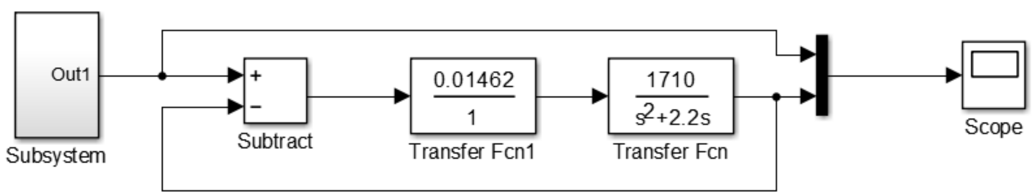
```
Gm =
   Inf
Pm =
    4.3118
Wgm =
   Inf
Wpm =
   58.4397
```

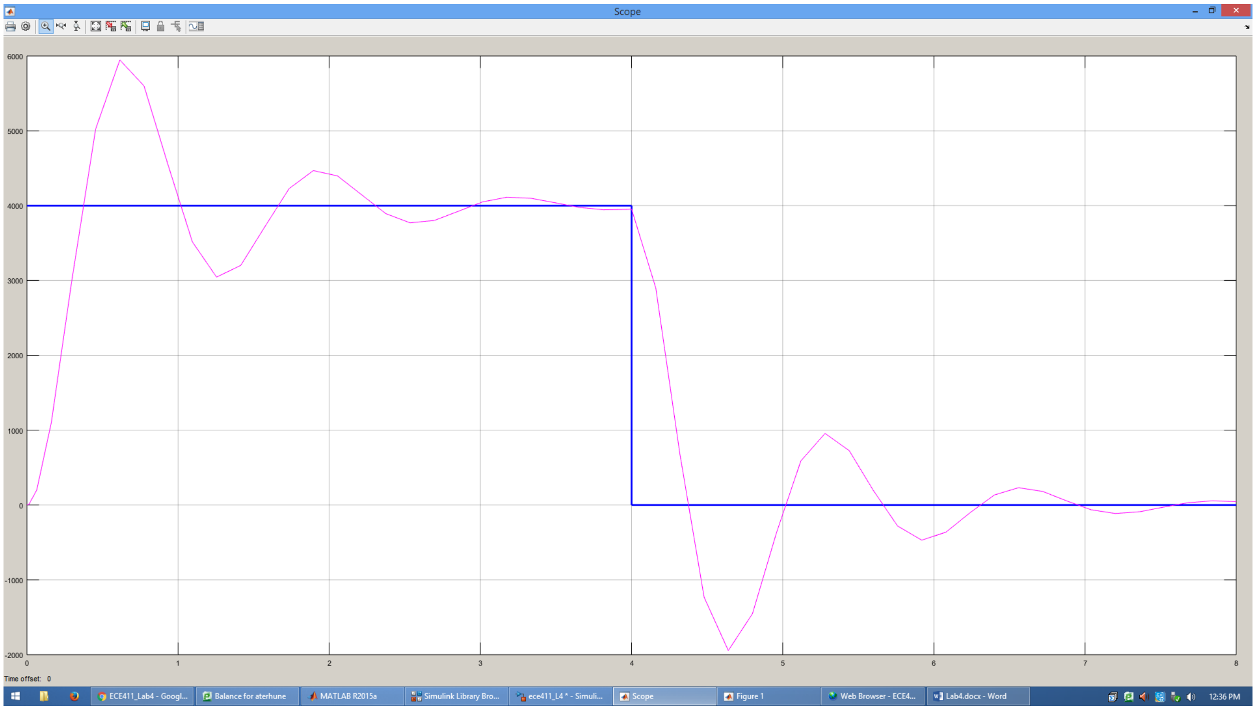
In the graphs above you can see the time domain behavior of the uncompensated system through my bode plot, step response and transient. All of which are in an acceptable range. Above are also all the needed calculations via Matlab (Gm, Pm). Also included is a schematic of the system I used.

Part 2

Number 1: Adding gain compensator








```
H = tf(25,[1 2.2 25]);

M=4000*H

[Gm,Pm,Wgm,Wpm] = margin(H)

step(M)
```

```
M =

        100000
  ----------------
  s^2 + 2.2 s + 25

Continuous-time transfer function.

Gm =
   Inf

Pm =
   36.2480

Wgm =
   Inf

Wpm =
    6.7206
```

```
### (see explanation below)
```

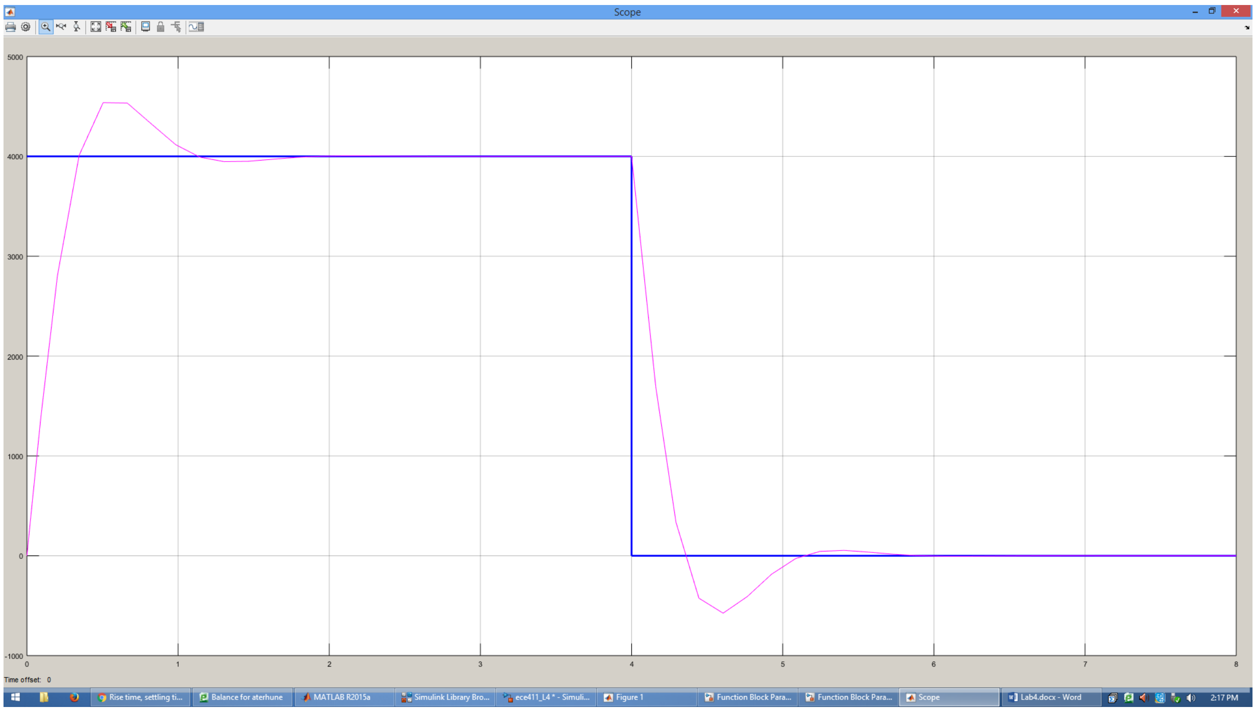
$$G_c(s) = K_c \frac{(1+\tau_1)}{(1+\ \tau_2)}$$

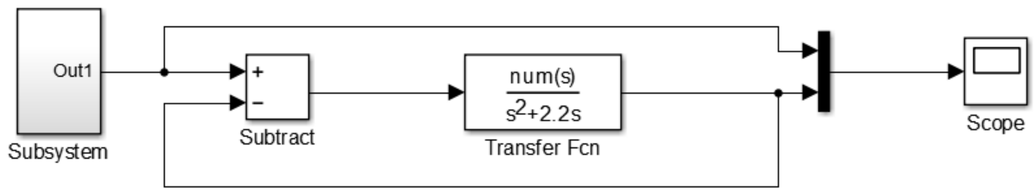
From above determined Kc t1 and t2 to be:

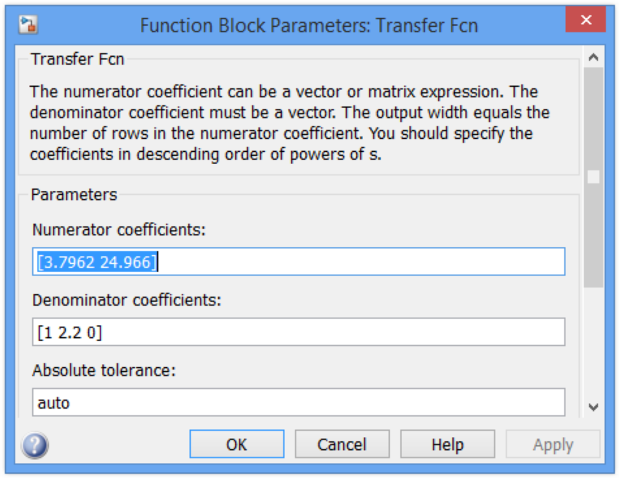
K_c = (1/68.4)

τ_1 = 0.152

τ_2 = 0




Out1
Subsystem
+
–
Subtract
num(s)
s2+2.2s
Transfer Fcn
Scope


Function Block Parameters: Transfer Fcn
Transfer Fcn
The numerator coefficient can be a vector or matrix expression. The denominator coefficient must be a vector. The output width equals the number of rows in the numerator coefficient. You should specify the coefficients in descending order of powers of s.
Parameters
Numerator coefficients:
[3.7962 24.966]
Denominator coefficients:
[1 2.2 0]
Absolute tolerance:
auto
OK
Cancel
Help
Apply

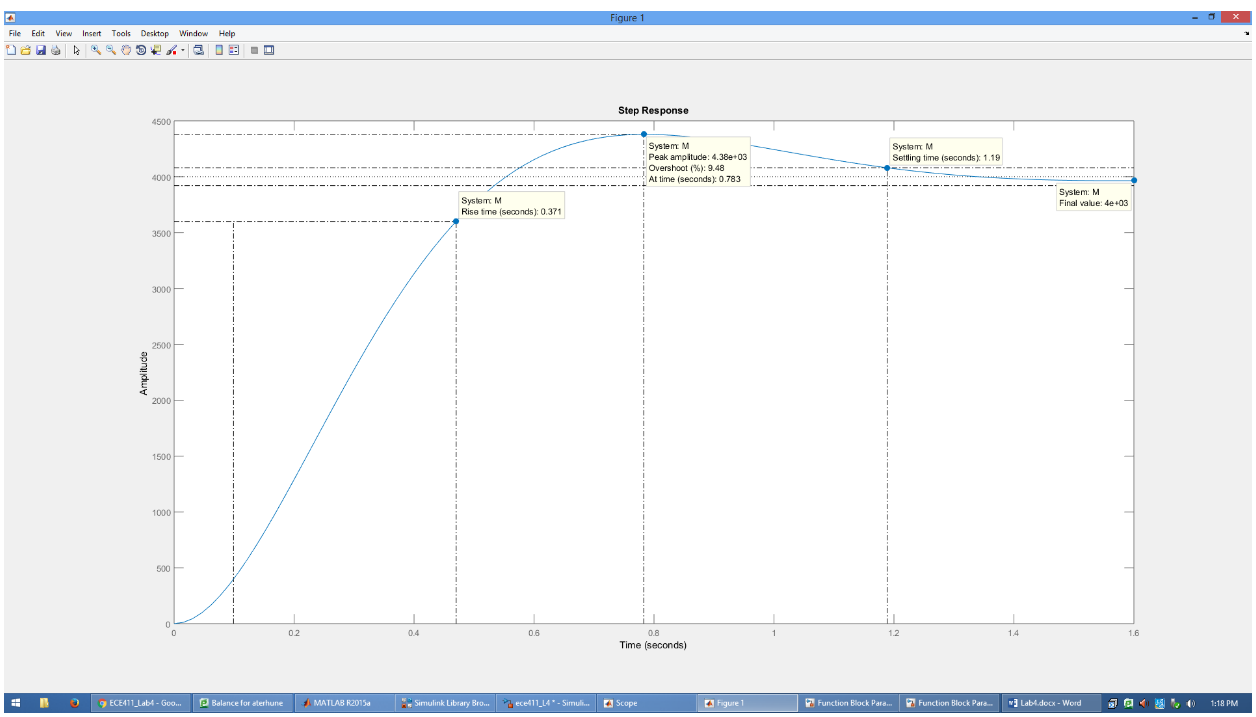
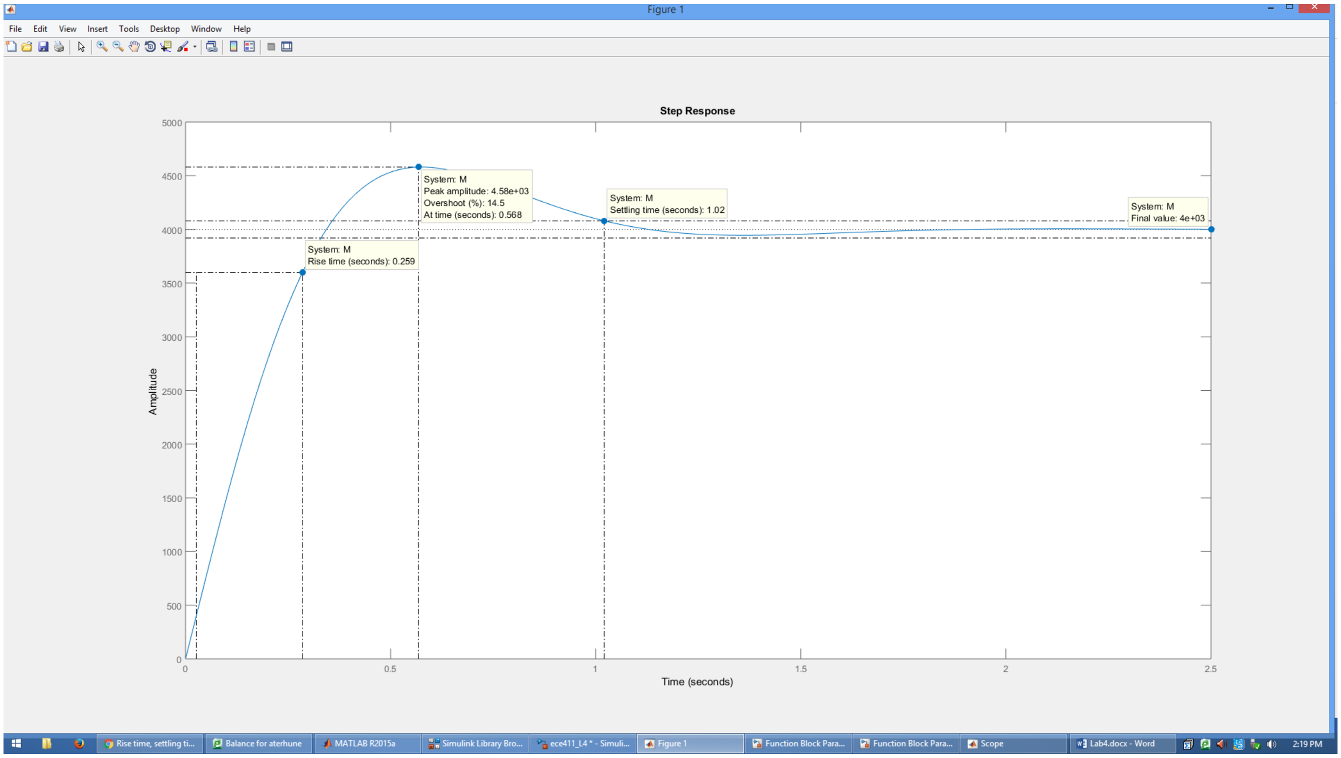


Figure 1
File Edit View Insert Tools Desktop Window Help
Step Response
System: M
Peak amplitude: 4.38e+03
Overshoot (%): 9.48
At time (seconds): 0.783
System: M
Rise time (seconds): 0.371
System: M
Settling time (seconds): 1.19
System: M
Final value: 4e+03
Amplitude
Time (seconds)




```
B = 1710*(1/68.4);
A = 1710*0.152*(1/68.4);
H = tf([A B],[1 6 25]);
M=4000*H;
[Gm,Pm,Wgm,Wpm] = margin(H)
step(M)
stepinfo(M)
```

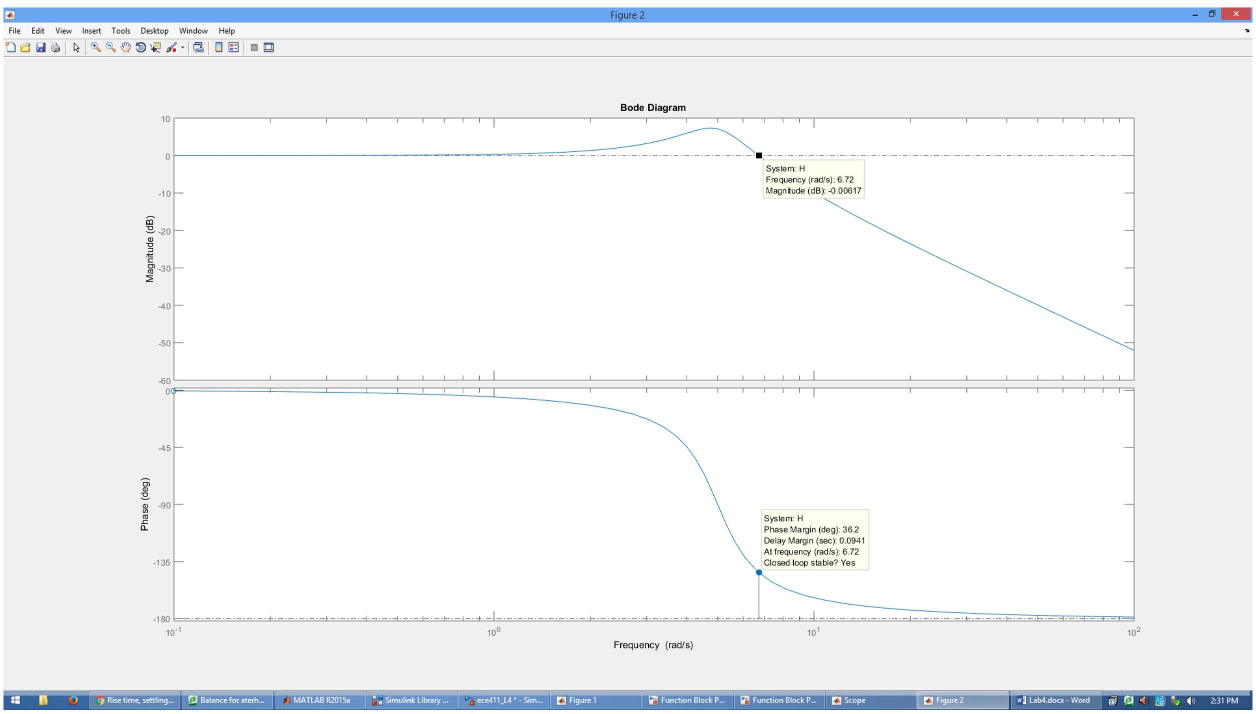
```
Gm =
   Inf

Pm =
  122.8924

Wgm =
   NaN

Wpm =
    5.3329

ans =

        RiseTime: 0.2591
    SettlingTime: 1.0208
     SettlingMin: 3.6509e+03
     SettlingMax: 4.5808e+03
       Overshoot: 14.5190
      Undershoot: 0
            Peak: 4.5808e+03
        PeakTime: 0.5680
```

In part two I added the simple gain compensator and observed better results (such as shorter rise time and better signal attenuation which can be seen in the step response plot), compared to part 1. In the section labeled "###" of part two I used a different Gc and analyzed what this did to the system. As you can see from the step response and bode plots the signal attenuated better and there was also less overshoot (and again shorter rise time), all of these values are printed out above from Matlab.

Part 3

Number 1:

K = 1/68.4



PM = 36.2

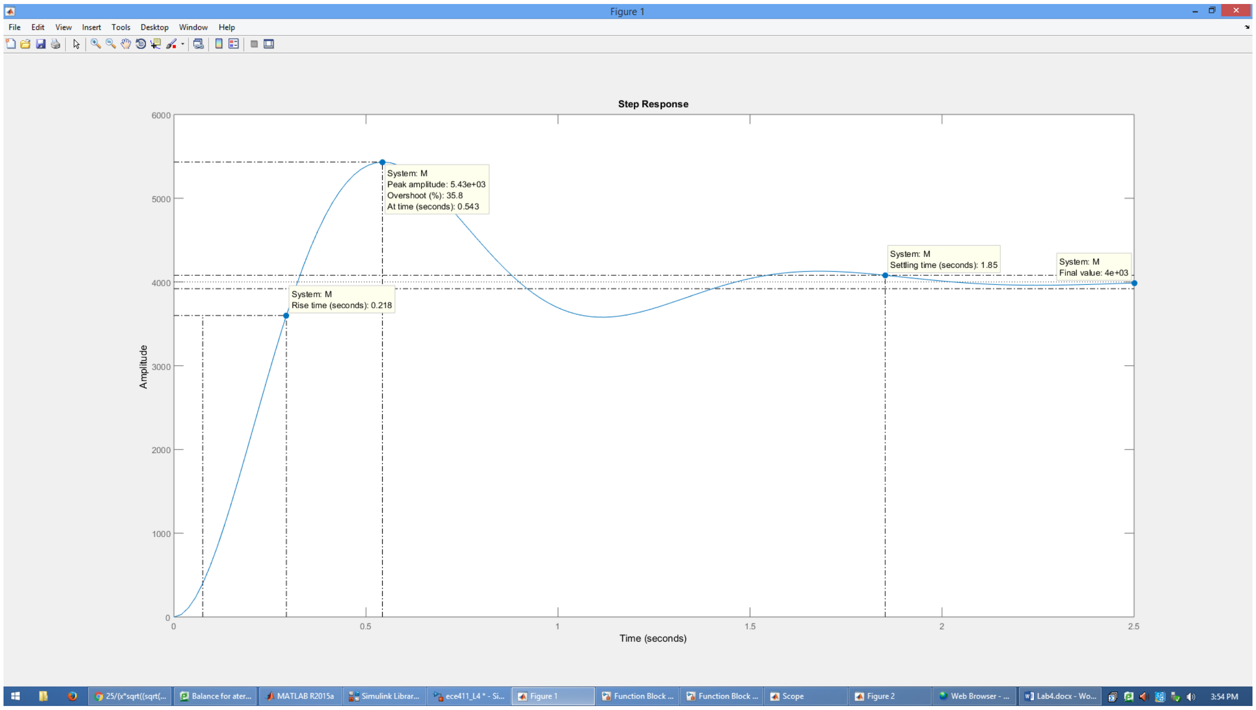
GM = Inf

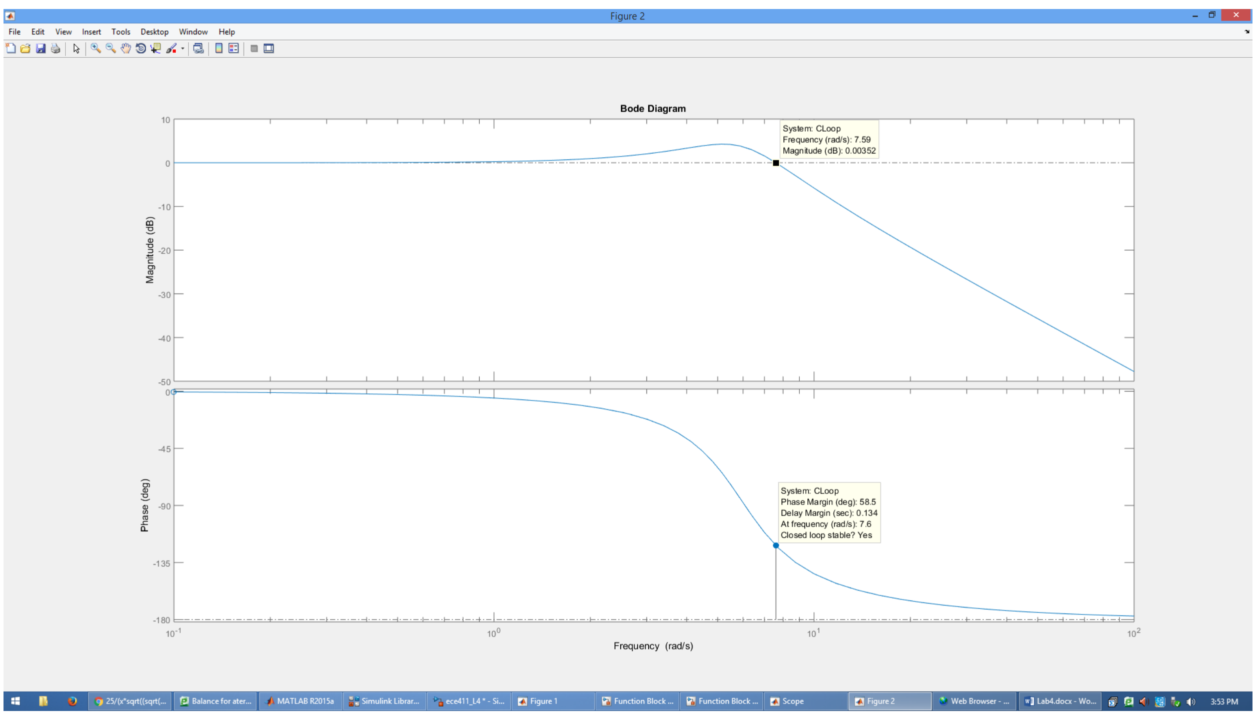
Number 2:

For a ϕ_{max} of 8.8 + 5 (rule of thumb) the following Phase-Lead transfer function was determined

$\alpha = 1.627$

$\omega_1 = 4.26$

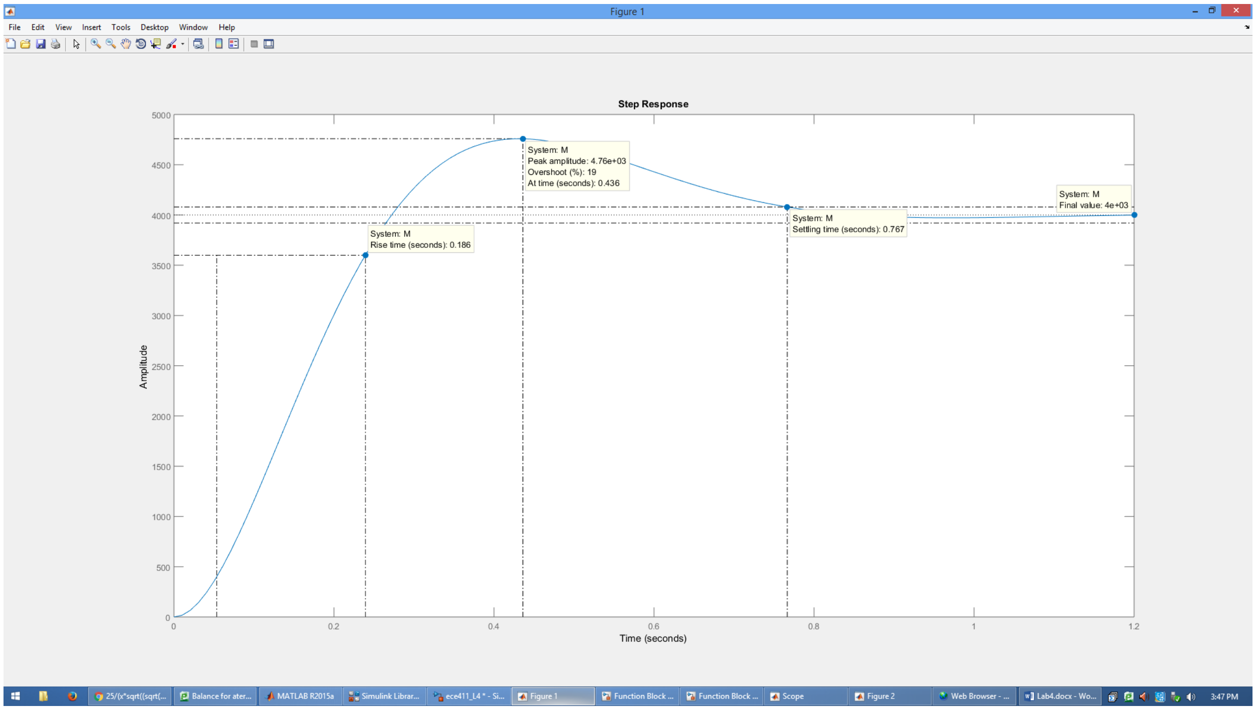


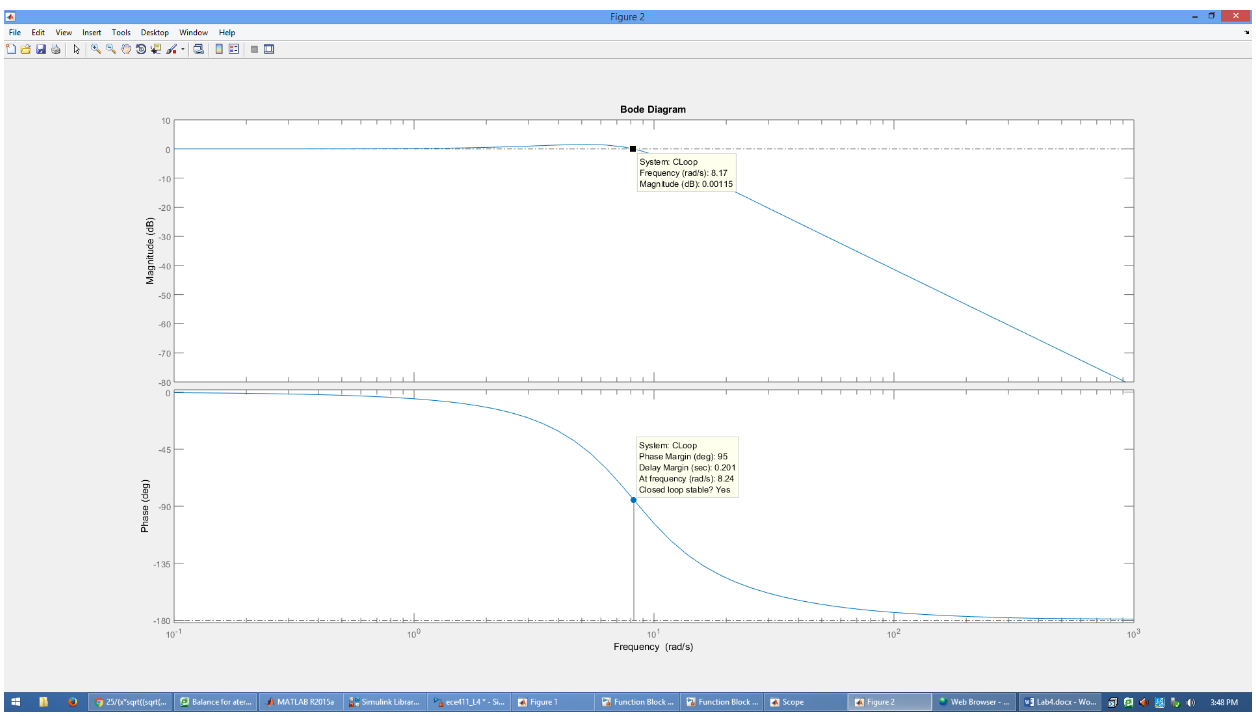


When I increase ϕ_{max} gradually to 45 degrees this improved the system response…

$\alpha = 3.414$

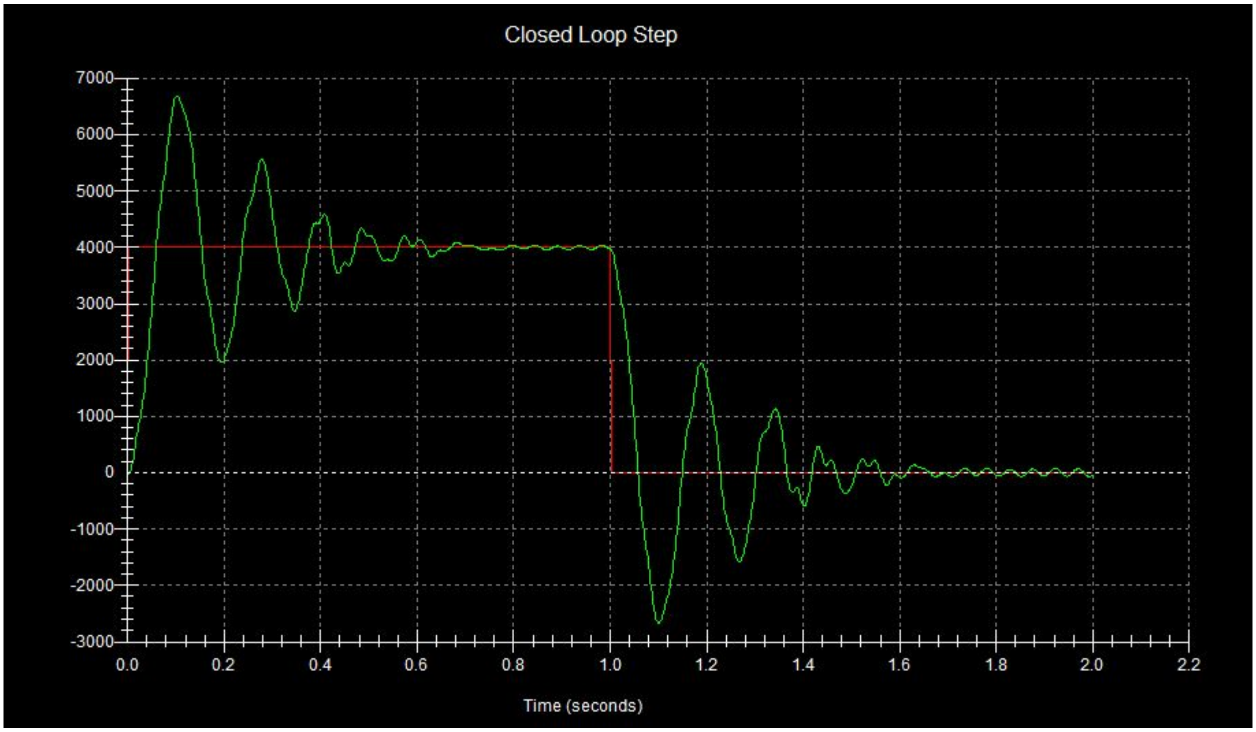
$\omega_1 = 3.583$

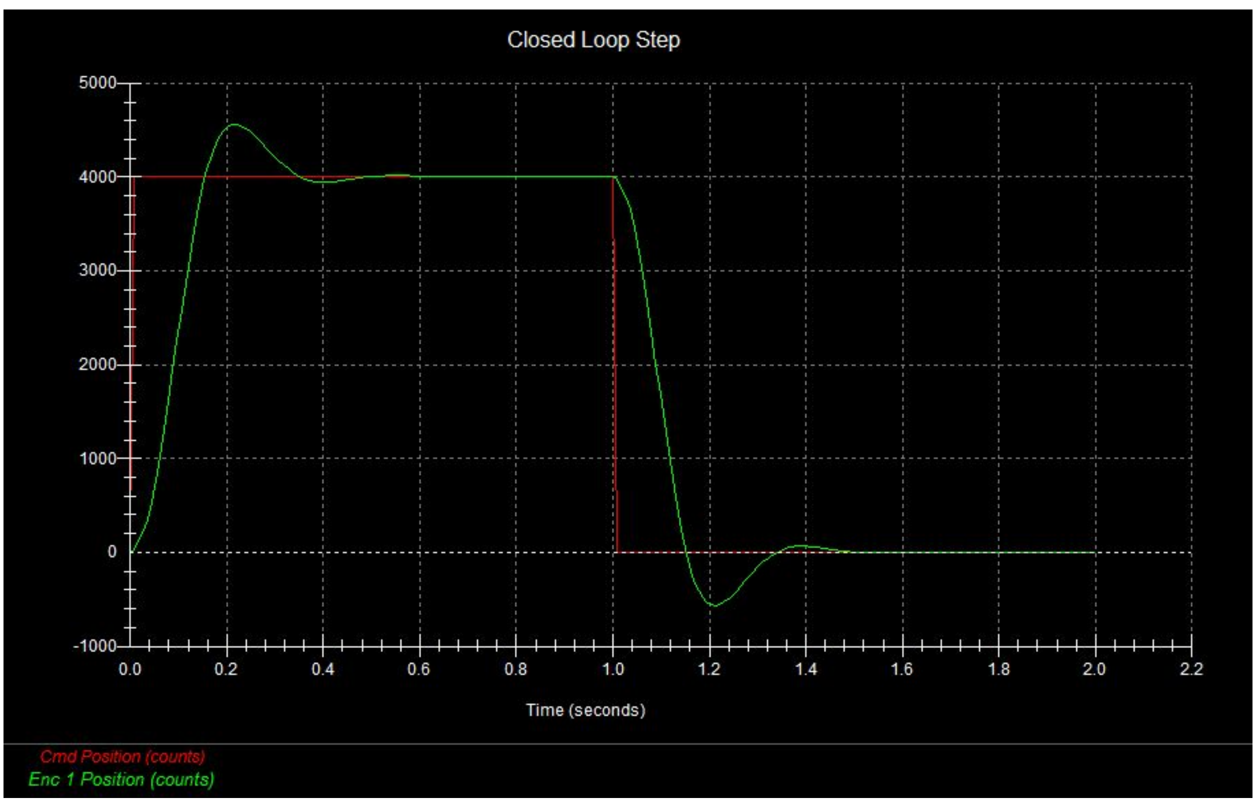




In part three using a K=1/68.4 I generated the former bode plots. In "number 2" of part 3 we specified our values of GM and PM and plotted the results of our design and derived values of our free variables.

Part 4


Closed Loop Step
7000
6000
5000
4000
3000
2000
1000
0
-1000
-2000
-3000
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
2.2
Time (seconds)


Closed Loop Step
5000
4000
3000
2000
1000
0
-1000
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
2.2
Time (seconds)
Cmd Position (counts)
Enc 1 Position (counts)

In part four the two plots above are generated results from the Model 220 Servo Trainer. They represent the closed loop step function and as you can see there is more overshoot than anticipated but the signal does attenuate more or less pretty quickly. The rise time is longer than anticipated as well. The first graph shows closed loop plot for the uncompensated system and the second shows the compensated system.

Conclusion:

I demonstrated my ability to model a control system using Simulink as well as the theory behind controls engineering. I demonstrated how real-world data and modeled/simulated data do not always match but the closer the two sets the better your simulation is and the better chance you have to explain the responses and results of your system, furthermore the better chance you have to vary the parameters of your system in order to obtain desired results and then observe those results in real world application.