EE-420 - Feedback Control Systems

Department of Electrical & Computer Engineering
San Diego State University

Magnetic Levitator

EE-420 Course Project

Date: May 17th, 2021

Name: Andrew Chung, RedID: 821542300

ABSTRACT

The purpose of this project is to design a magnetic levitation system controlled by a unity feedback network to suspend and stabilize a magnetic ball in the air. This magnetic system includes an electromagnet and sensors that detect the position of the magnetic ball. Since the initial state of the system is unstable, a compensator must be designed to stabilize the system and suspend the ball at a steady-state position of 0.05m from the electromagnet. In designing the system, the system's equation was linearized about a point of equilibrium, and a linear approximation of the nonlinear system was then obtained. Then, a lead compensator was designed to stabilize the system, and was tested on the nonlinear system. The steady-state error is eliminated by redesigning the compensator as a PID controller.

PROCEDURE

1. Find the steady-state value I_{SS} of the current required to balance the ball at an arbitrary desired equilibrium point $y = r > 0$.

Equation of motion of the ball: $my'' = -ky' + mg - \frac{L_0 i^2}{2a(1+\frac{y}{a})^2}$

$$y'' = -\frac{k}{m}y' + g - \frac{L_0 i^2}{2am(1+\frac{y}{a})^2} = f(y', y, i)$$

Derivatives are zero at equilibrium point:

$$I_{SS} = \bar{i} = \sqrt{\frac{2amg}{L_0}}(1 + \frac{\bar{y}}{a})$$

If equilibrium point $\bar{y} = r = 0.05m$, $m = 0.1kg$, $k = 0.001N/m/s$, $g = 9.81m/s^2$, $a = 0.5$, and $L_0 = 0.01H$

$$I_{SS} = 6.2642A$$

2. Linearizing the nonlinear equation

Linearize about the points $y = \bar{y},\ y' = 0,\ i = \bar{i}$

$$i_\delta = i - \bar{i}, y_\delta = y - \bar{y}, y'_\delta = 0$$

$$y''_\delta = \frac{df(0,\bar{y},\bar{i})}{dy}y_\delta + \frac{df(0,\bar{y},\bar{i})}{dy'}y'_\delta + \frac{df(0,\bar{y},\bar{i})}{di}$$

$$y''_\delta = \frac{L_0\bar{i}^2}{ma^2(1+\frac{\bar{y}}{a})^3}y_\delta - \frac{k}{m}y'_\delta - \frac{L_0\bar{i}}{ma(1+\frac{\bar{y}}{a})^2}i_\delta$$

$$(s^2)y_\delta - \frac{L_0\bar{i}^2}{ma^2(1+\frac{\bar{y}}{a})^3}y_\delta + (\frac{k}{m}s)y'_\delta = (-\frac{L_0\bar{i}}{ma\ (1+\frac{\bar{y}}{a})^2})i_\delta$$

Linearized model:

$$G_p(s) = \frac{y_\delta(s)}{i_\delta(s)} = \frac{-\frac{L_0 \bar{i}}{ma\,(1+\frac{\bar{y}}{a})^2}}{s^2+\frac{k}{m}s-\frac{L_0 \bar{i}^2}{ma^2(1+\frac{\bar{y}}{a})^3}} = \frac{-3.132}{s^2+0.01s-196.2}$$

$$G_p(s) = \frac{-3.132}{(s+14)(s-14)}$$

Thus, the linearized system has poles at approximately $s_{1,2} = \pm\ 14$

Using Matlab:

```
Gp = tf(-3.132, [1 .01 -196.2])
pole(Gp)
rlocus(Gp)

Gp =

          -3.132
  --------------------
  s^2 + 0.01 s - 196.2

Continuous-time transfer function.


ans =

  -14.0121
   14.0021
```

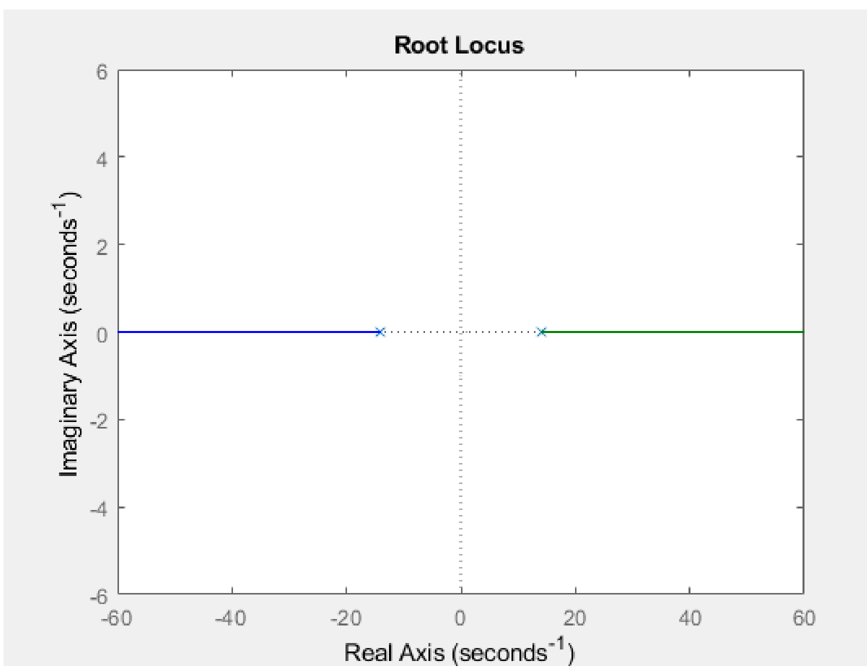
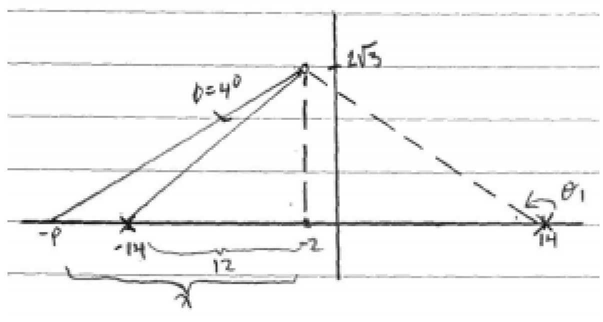


Figure 1: Root locus of unstable system

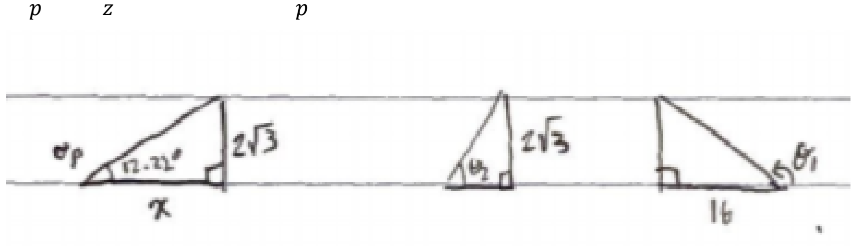
The system is unstable because there is right-hand plane (RHP) pole at $s = 14.0021$

3. Next, a linear compensator $G_c(s)$ is designed for the linearized plant $G_p(s)$ to regulate the value of the output to $y = r = 0.05m$.The compensator that was chosen to be utilized is a lead compensator with the form $G_c(s) = \frac{K_c(s+z)}{(s+p)}$. Steady-state specifications $\delta = 0.5$, 2% settling time = $2s \rightarrow \omega_n = 4\frac{rad}{s}$ lead to $s_0 =- \delta\omega_n \pm j\omega_n\sqrt{1-\delta^2} = -2 \pm j2\sqrt{3}$
$= -2 \pm j3.4641$. Then,
$\varphi = -\angle G(s_0) - 180 = -\angle\frac{3.132}{(s_0+14)(s_0-14)}\Big|_{s_0=-2+j2\sqrt{3}} - 180$
$= -176 - 180 = -356 = 4°$. The compensator $G_c(s)$ must provide $4°$ at
$s_0 = -2 + j2\sqrt{3}$

Using pole-zero cancellation, set $z = 14$, which will cancel the pole at -14. Pole $p = 18$ is found using geometry



$\theta_2 = tan^{-1}(\frac{2\sqrt{3}}{12}) = 16.1°$, $\theta_1 = 180 - tan^{-1}(\frac{2\sqrt{3}}{16}) = 167.78°$,
$\theta_1 + \theta_2 = 183.88°$
$183.88° + \theta_p - \theta_z = 180° \rightarrow \theta_p = 12.22°$



$tan(12.22°) = \frac{2\sqrt{3}}{x} \rightarrow x = \frac{2\sqrt{3}}{tan(12.22°)} \rightarrow x = 16 \Rightarrow -p = -18$
$G_c(s) = K_s\frac{(s+14)}{(s+18)}$

With Magnitude criterion: $K_c \left| \frac{(s_0+14)(-3.132)}{(s_0+18)(s_0-14)(s_0+14)} \right|_{s_0=-2+j2\sqrt{3}} = 1 \rightarrow K_c = 85.57$

Thus, lead compensator: $G_c(s) = \frac{85.57(s+14)}{(s+18)}$

$$G_{OL}(s) = G_c(s)G_p(s) = \frac{85.57(s+14)}{(s+18)} \frac{-3.132}{(s+14)(s-14)}$$

Evaluating the lead compensator on the nonlinear system through simulink:

```
clear, clc;
m=0.1;       % m=0.1kg
k=0.001;     % N/m/s
g=9.81;      % m/s^2
a=0.05;
Lo=0.01;     % H

y_ss = 0.05;         % steady-state position r=0.05
i_ss = (m*g*2*a*(1+y_ss/a)^(2))^(1/2)/(Lo)^(1/2);    % steady-state current, 6.2642A
```

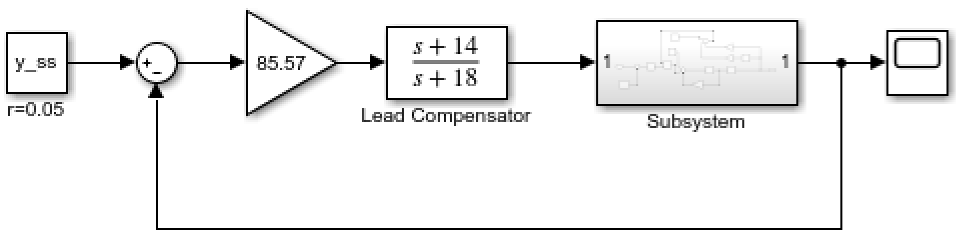


Figure 2: Lead compensator Simulink model

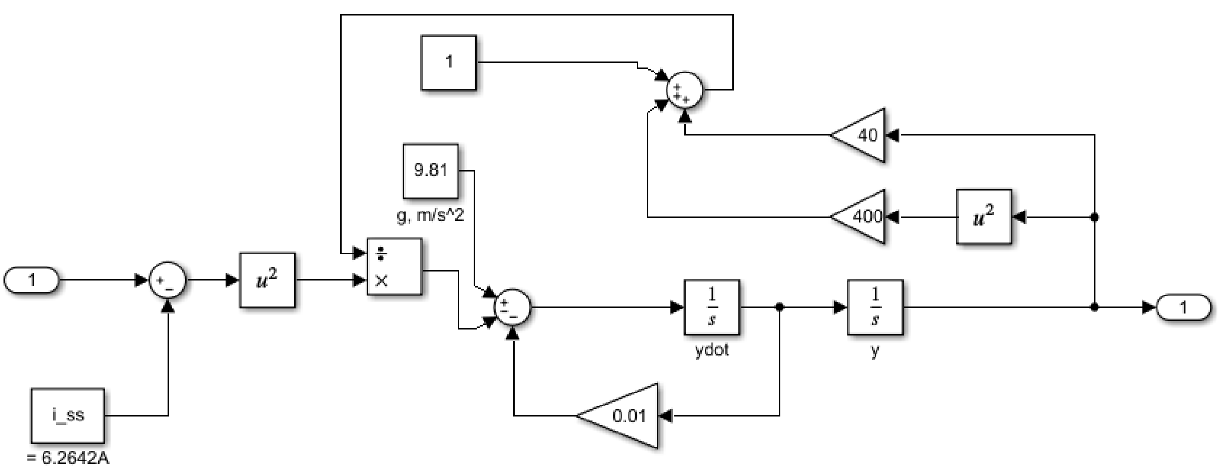


Figure 3: Linearized plant (Subsystem)

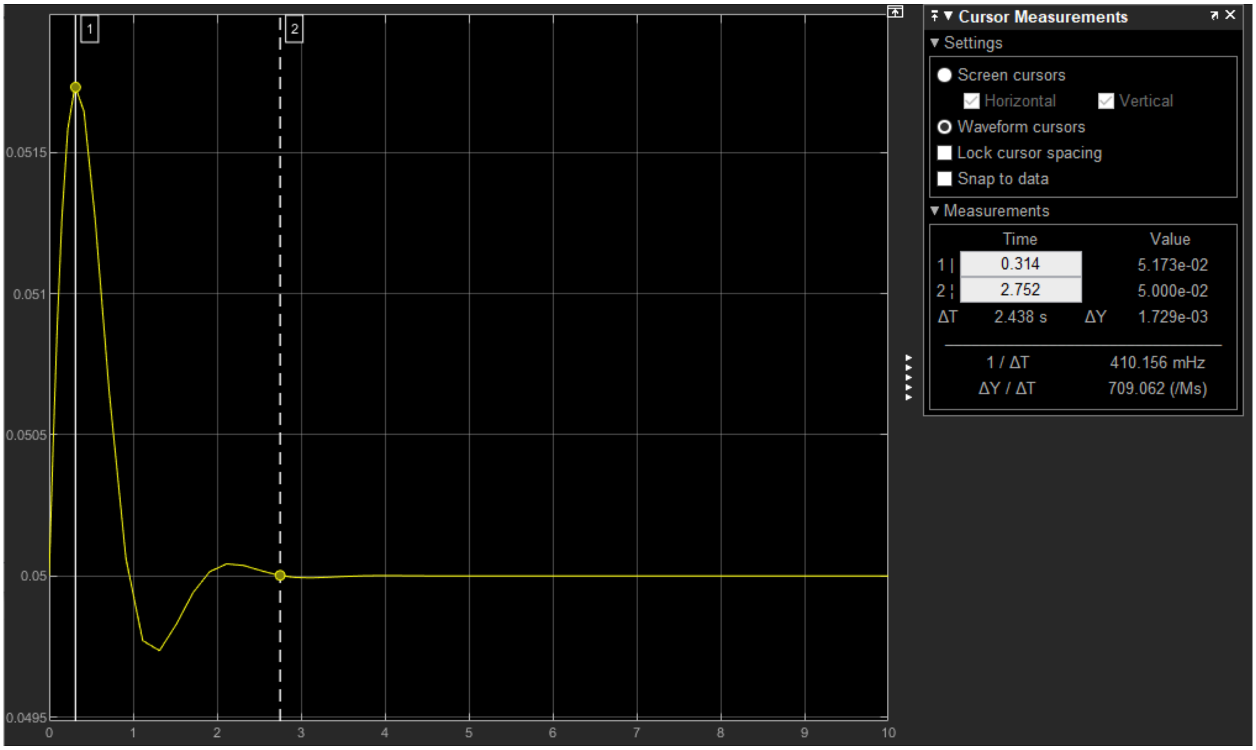


Figure 4: Step response of the lead compensated system

We have the desired steady-state value of y=0.05m, with a 2% settling time at close to 2.3s, and the percent overshoot is around 3.3%.

An initial velocity must be chosen and the initial position changed for the ball to find how far away from equilibrium it can move while still being able to go back to equilibrium position. I set the initial velocity to 0.1m/s, and the initial position to 0m, and the ball returned to the equilibrium point.

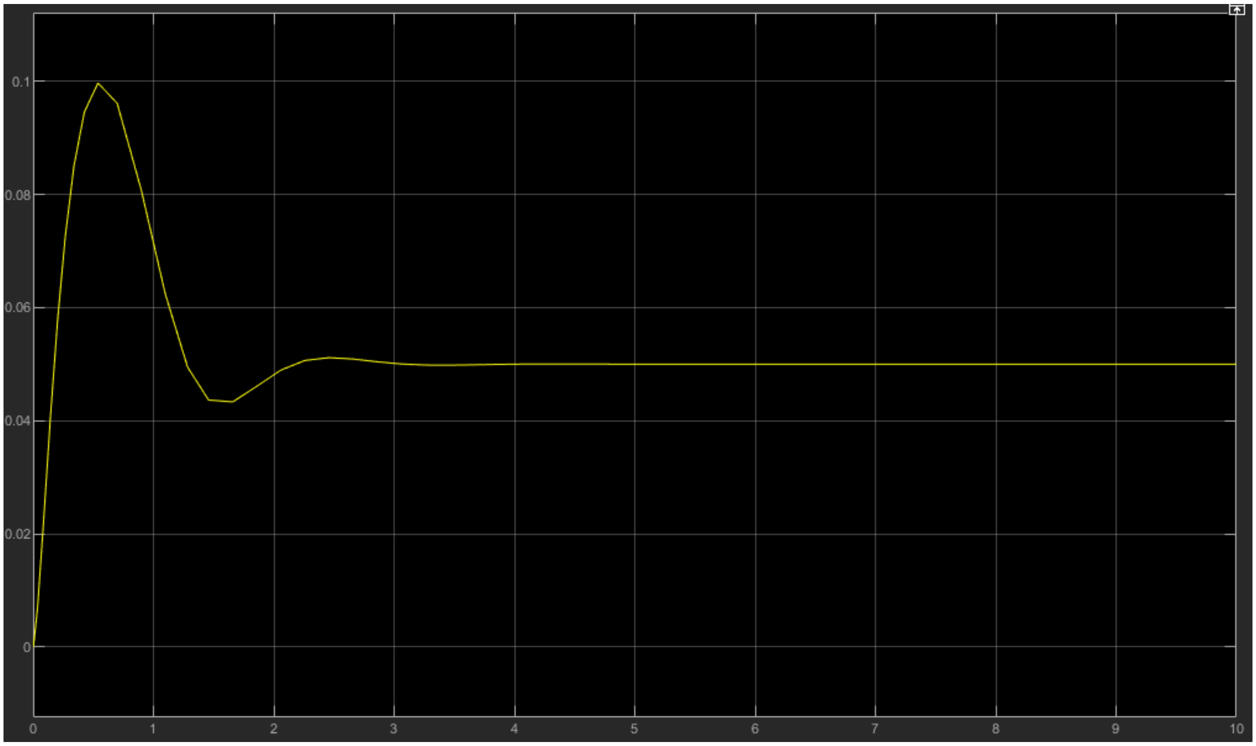


Figure 5: Step response with initial position = 0m

4. In order to determine how far the ball can be moved at 0.1m/s, I increment the position to see if the ball would still return to the equilibrium point. The ball does return to the equilibrium position at 1.7m, but any higher and the system no longer returns to equilibrium. Therefore, the furthest the ball can go below equilibrium at 0.1m/s is about 1.7m.

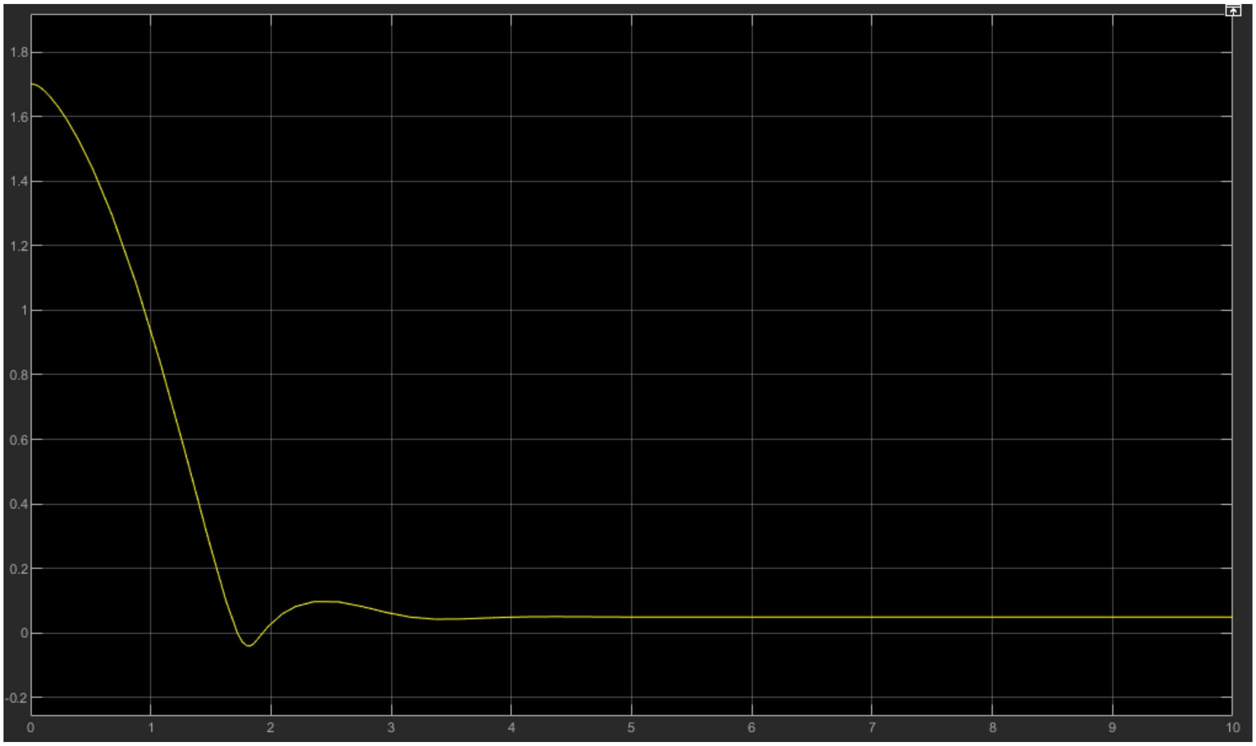


Figure 6: Step response with initial position = 1.7m

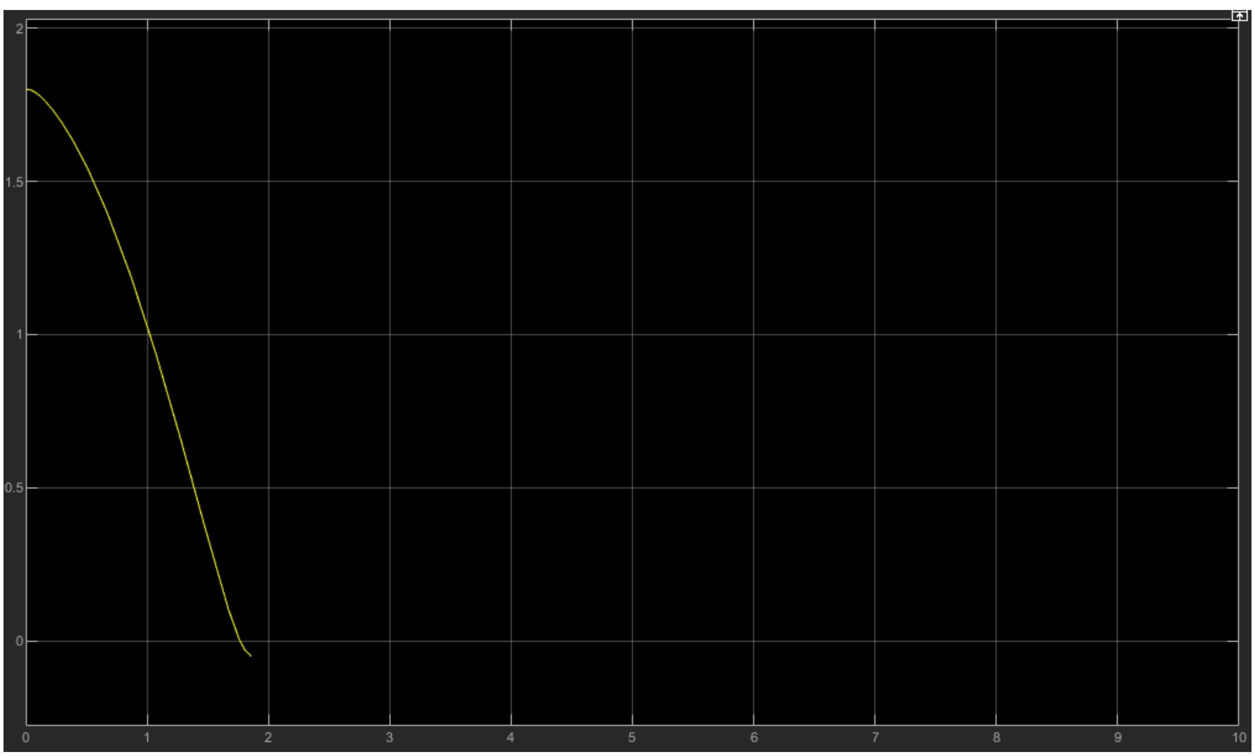


Figure 7: Step response with initial position = 1.8m

If the system were truly linear, these graphs would be straight lines. The values would vary linearly with the change in inputs.

5. We can change the mass in the Simulink model to find the effect of perturbations of mass m on the system. By trial and error, I found that even a small increase in mass will destabilize the system. Increasing to anything greater than 0.11kg will cause the system to become unstable.

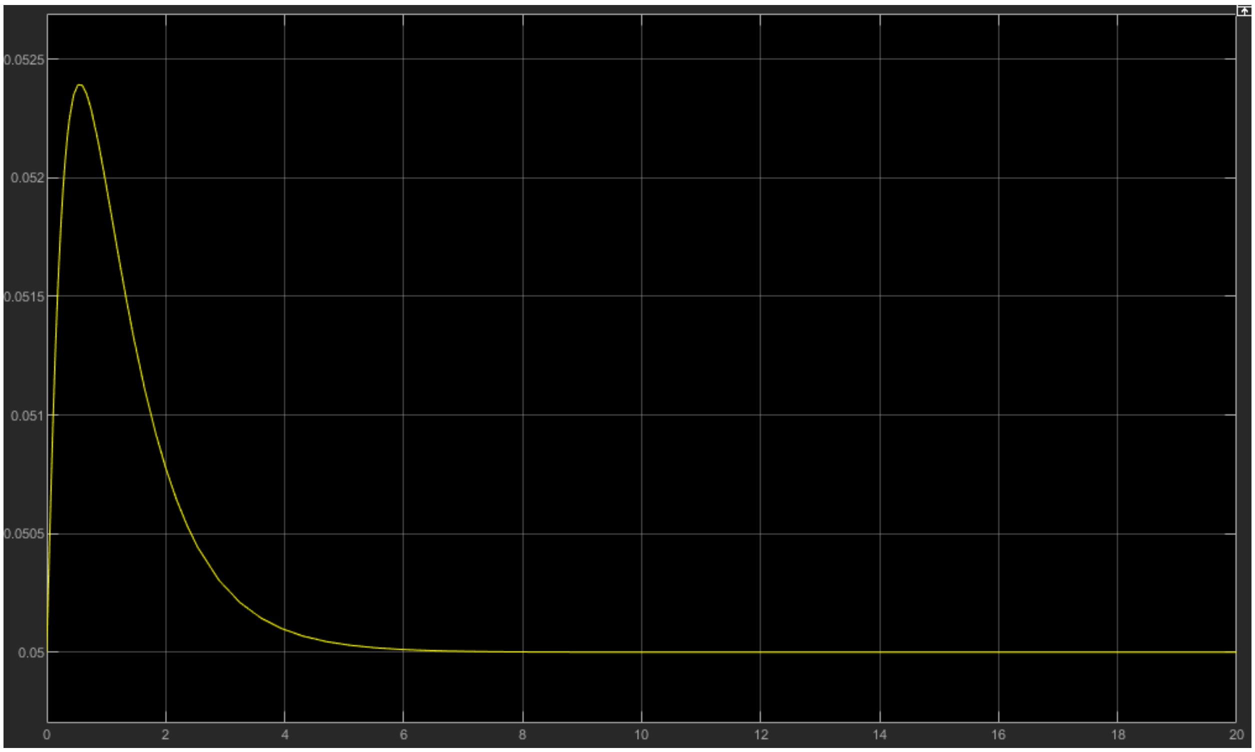


Figure 8: Response of system with mass = 0.11kg

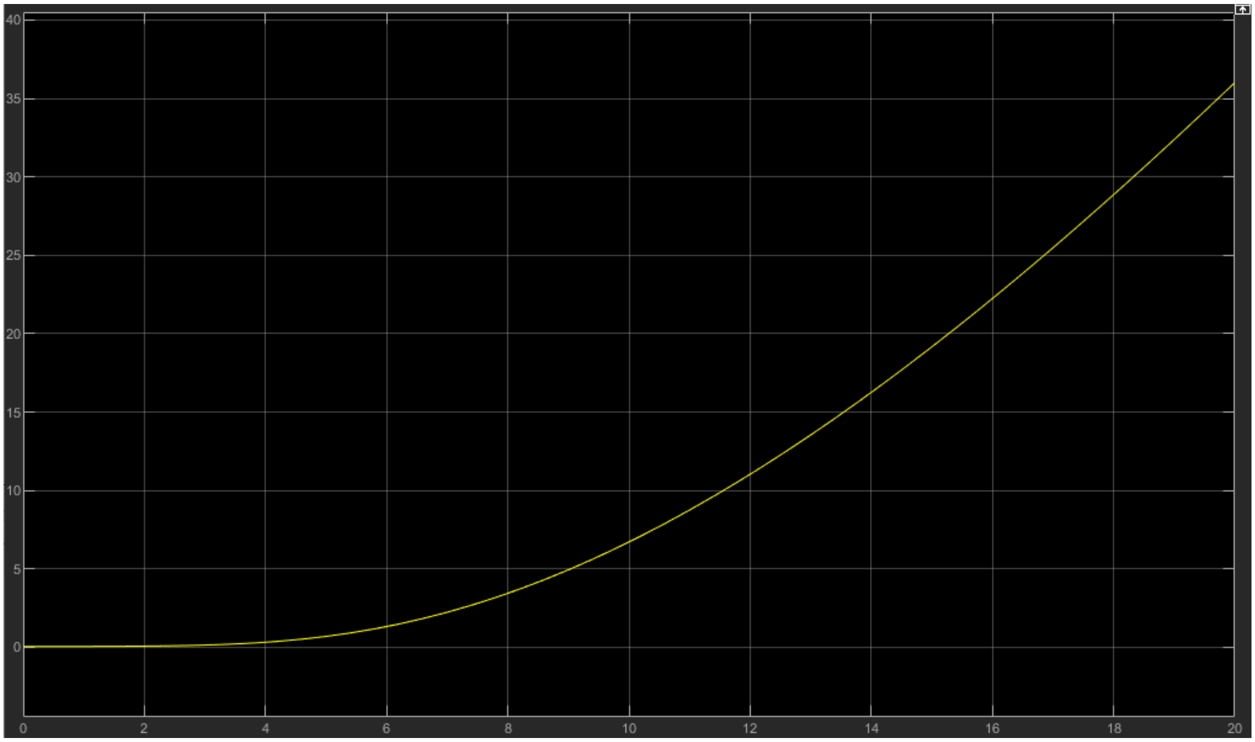


Figure 9: Response of system with mass = 0.12kg

On the other hand, by decreasing the mass, the system would oscillate more while in the transient state, until settling at a steady state value of y = 0.05m. As mass is decreased, the frequency of oscillations increases.

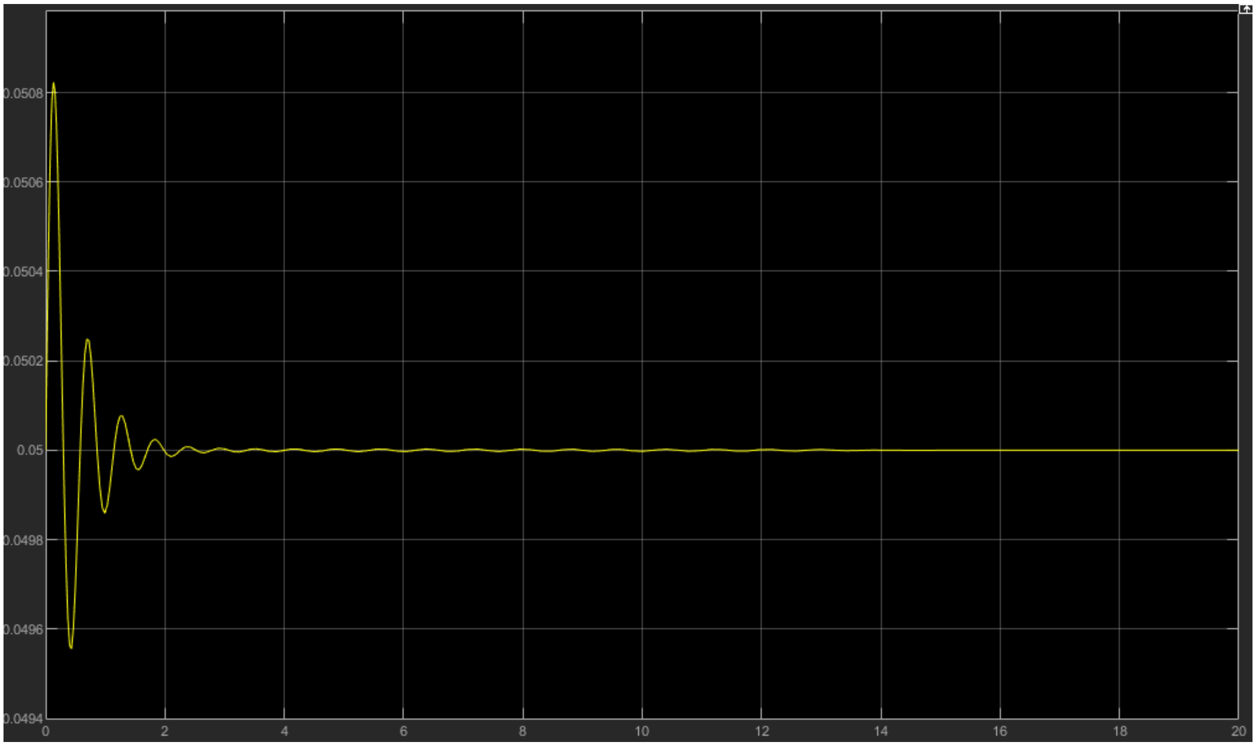


Figure 10: Response of system with mass = 0.05kg

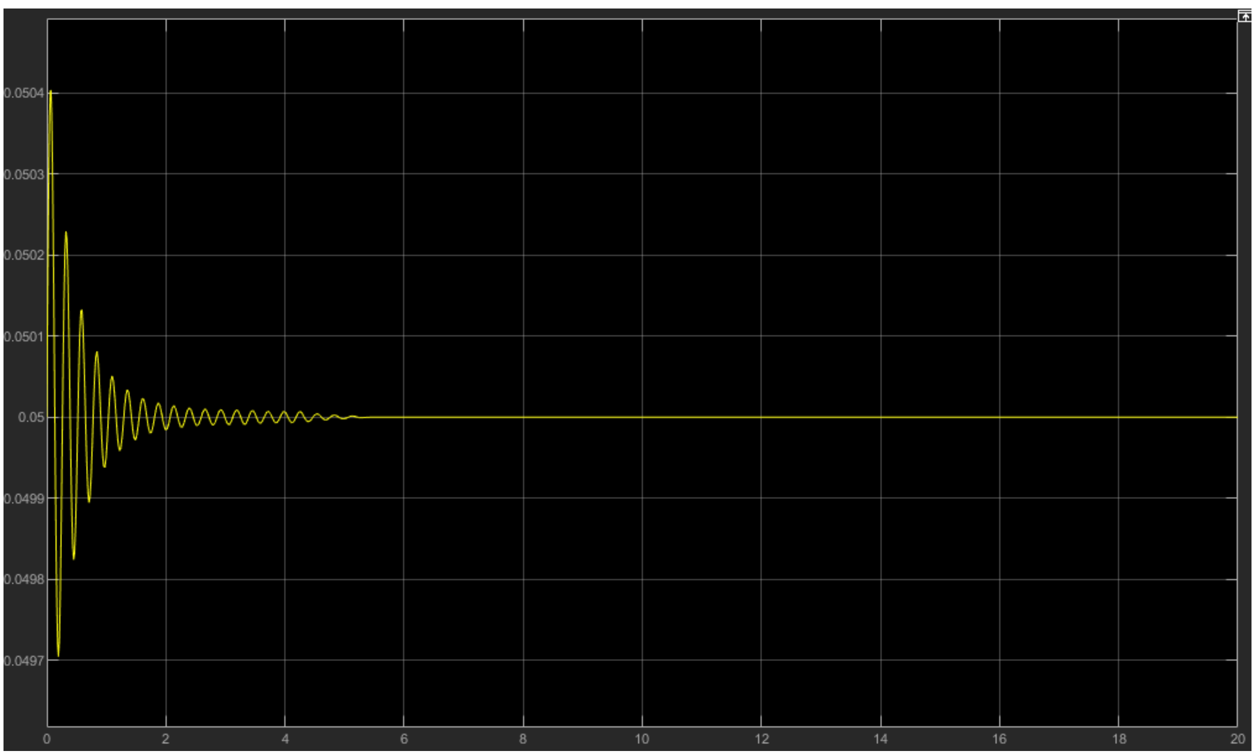


Figure 11: Response of system with mass = 0.01kg

6. When the parameters are perturbed, there is a steady-state error. Thus, the controller must be redesigned using integral control (PID) to get rid of the steady-state error.

$$G_c(s) = \frac{K_D s^2 + K_p s + K_I}{s(1+\varepsilon s)},\ \varepsilon << 1$$

Factoring out K_D,

$$G_c(s) = \frac{K_D(s^2 + \frac{K_p}{K_D}s + \frac{K_I}{K_D})}{s(1+\varepsilon s)} \rightarrow G_{OL}(s) = G_c(s)G_p(s) = \frac{K_D(s^2 + \frac{K_p}{K_D}s + \frac{K_I}{K_D})(-3.132)}{s(1+\varepsilon s)(s^2+0.01s-196.2)}$$

Next, a MATLAB script was used to inspect the root locus of the open loop gain of the system, where the open loop gain is the nonlinear plant and the PID is the controller in series.

```
K = 1;
Gp = tf(-3.1321,[1 0.01 -196.2]);                  % Original plant
Gc = tf(-k*conv([1 14.012], [1 1]), [1 0]);        % Proposed controller
G_open = series(Gp, Gc);
rlocus(G_open);
xlim([-50 25])
```

The first zero 14.012 was chosen by using pole-cancellation of the stable pole in the system plant. The second pole was found by messing around, and $(s + 1)$ was arbitrarily chosen for simplicity. Now we have

$$G_c(s) = \frac{K_D(s^2+15.0125s+14.012)}{s},\ \text{where } \frac{K_p}{K_D} = 15.0125 \text{ and } \frac{K_I}{K_D} = 14.012$$

Plotting the root locus:

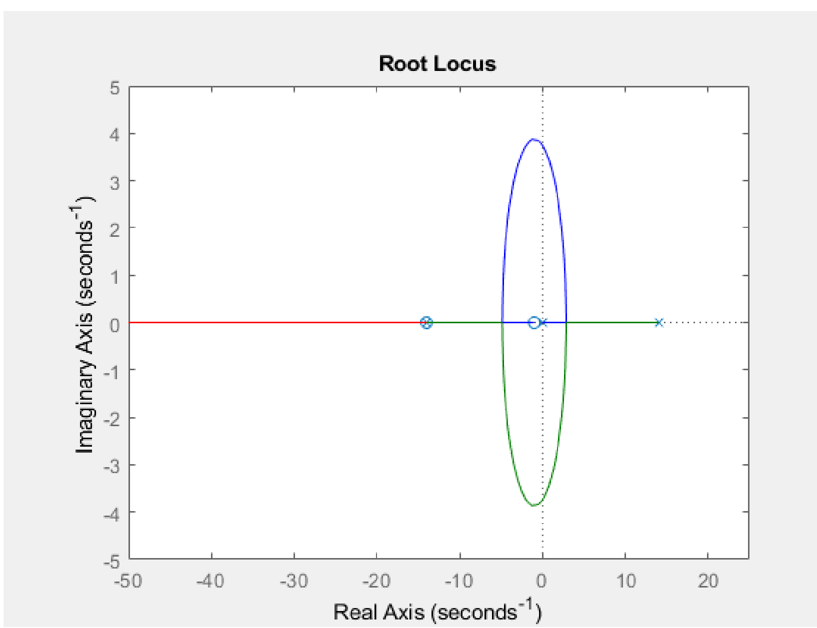


Figure 12: Root locus of open loop gain of system plant and PID controller

I found K_D by inspecting the root locus of the open loop gain. I set $K_D = 8$ as it represented the gain of the system on both the real axis and on the root locus. Now,

$$K_D = 120.096\ ,\ K_I = 112.096$$

Now, these values could be used to implement the PID controller to create the input to the nonlinear system plant.

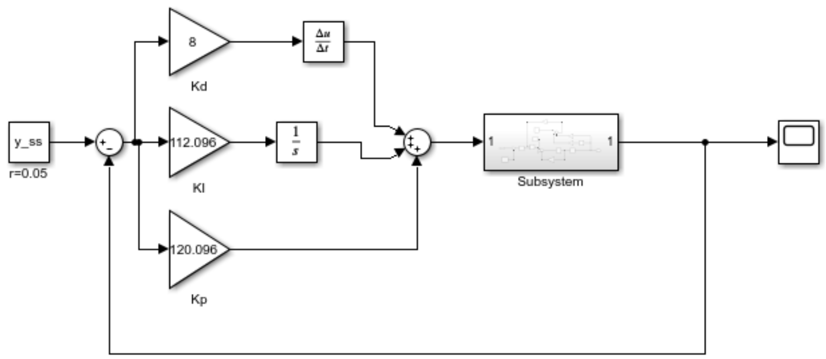


Figure 13: PID controller in Simulink

The step response was much better than the lead compensator, as the percent overshoot is about 0.54% versus 3.3%. There is still no steady state error.

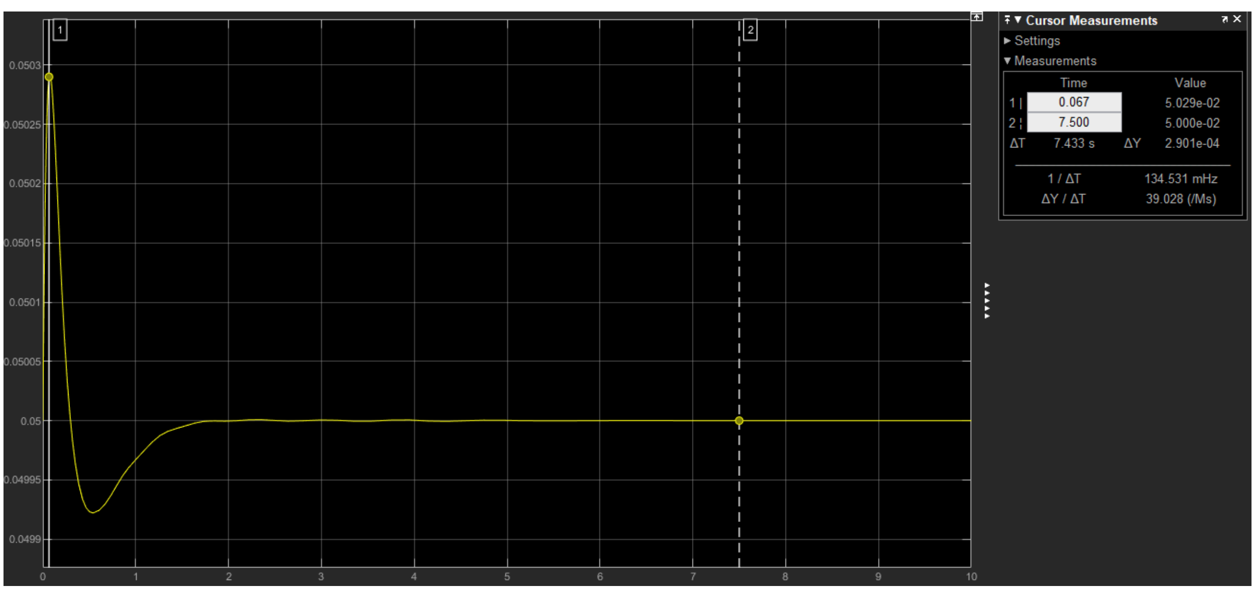


Figure 14: Step response of PID controller with m = 0.1175kg

Figure 15: Step response of PID controller with m = 0.001kg

7. By including a 15A limiter in the system, we can see the realistic nature of the controller design. A saturation limiter was placed in the subsystem and set to limits of $\pm\ 15A$. Measurements were taken to see if the limiter would affect the range of values for mass and distance from the equilibrium point.

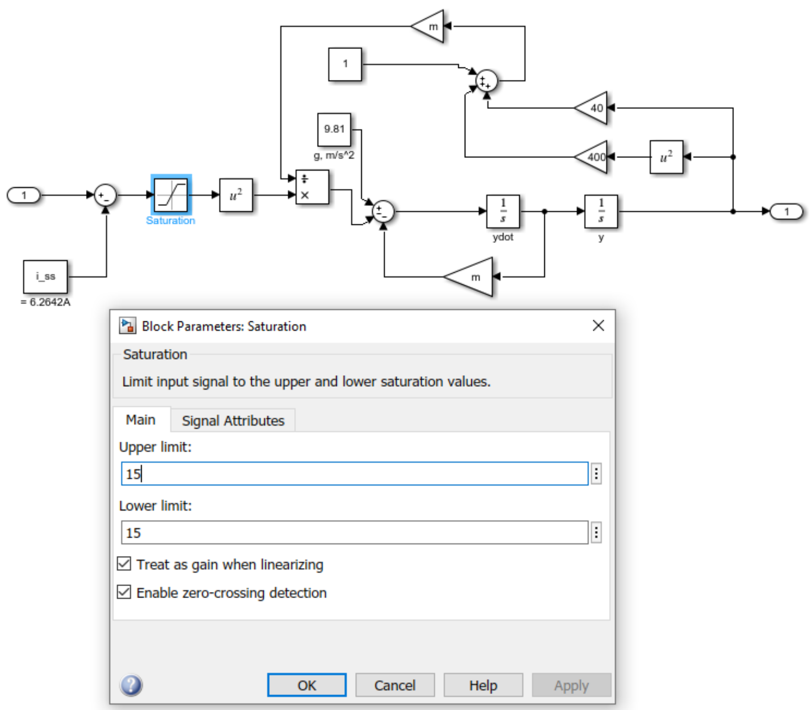


Figure 16: Nonlinear subsystem with limiter

With the limiter, the step responses of the lead compensator and the PID controller are the same. This means that the original controller specifications were realistic and even without the addition of the limiter, remained within practical limits of current. Limitations from the mass of the ball and distance away from equilibrium did not change.

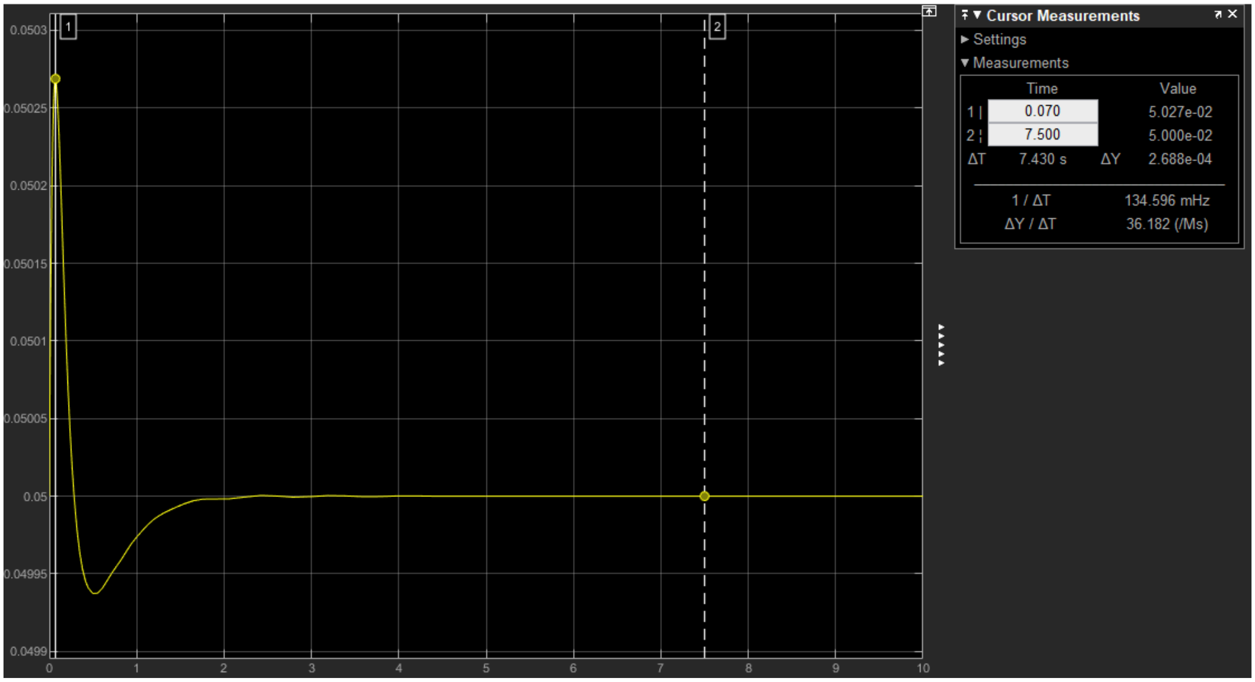


Figure 17: Step response of lead compensator with limiter

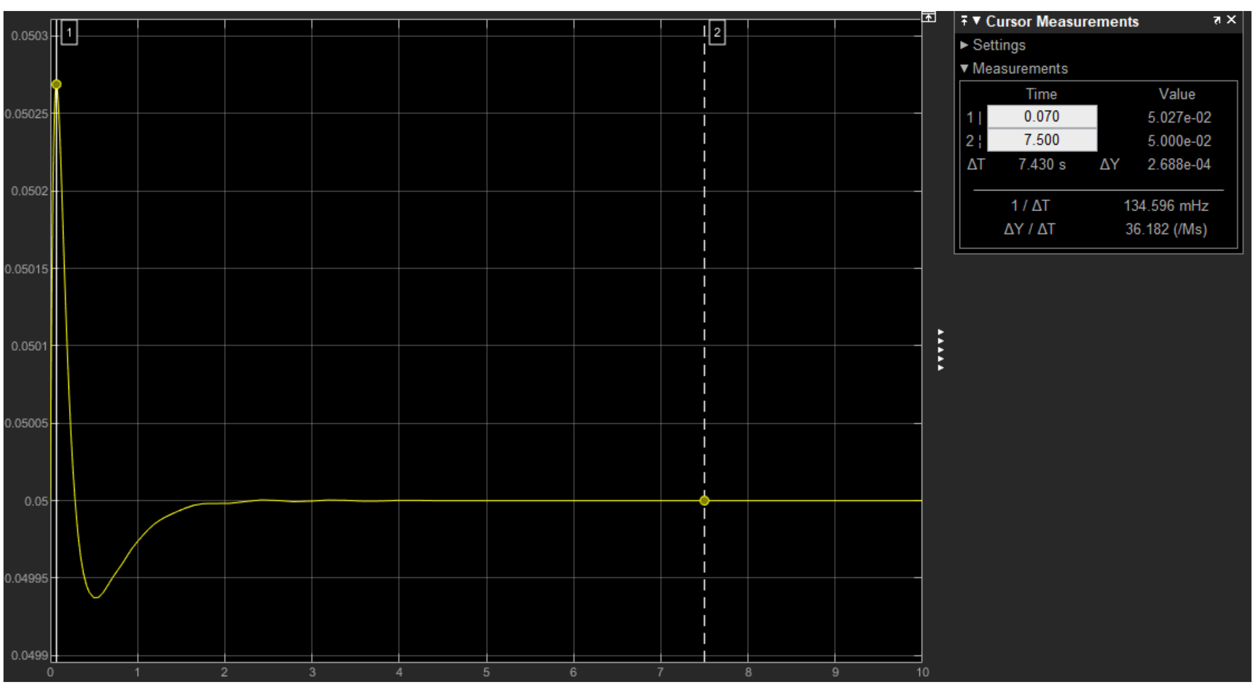


Figure 18: Step response of PID controller with addition of limiter

8. The final part was discretizing the lead compensator used with the limiter. This could be done using the “c2dm” command and ‘Tustin’ method of discretization, from the MATLAB Control Systems Toolbox. Using MATLAB to find the discrete transfer function representation of the lead compensator:

```
num = [85.57 1197.98];
den = [1 18];
Ts = 0.005;

[numd, dend] = c2dm(num,den,Ts,'tustin');
Lead = tf(numd,dend)
```

```
Lead =

  84 s - 73.01
  ------------
   s - 0.8349
```

This discrete transfer function was then placed in series with a zero order hold block and the nonlinear plant. The time constant was found to be 0.5s. However, a smaller sampling period of 0.005s achieves an accurate response because there is minimal space between each sample, which will more accurately represent the continuous controller.

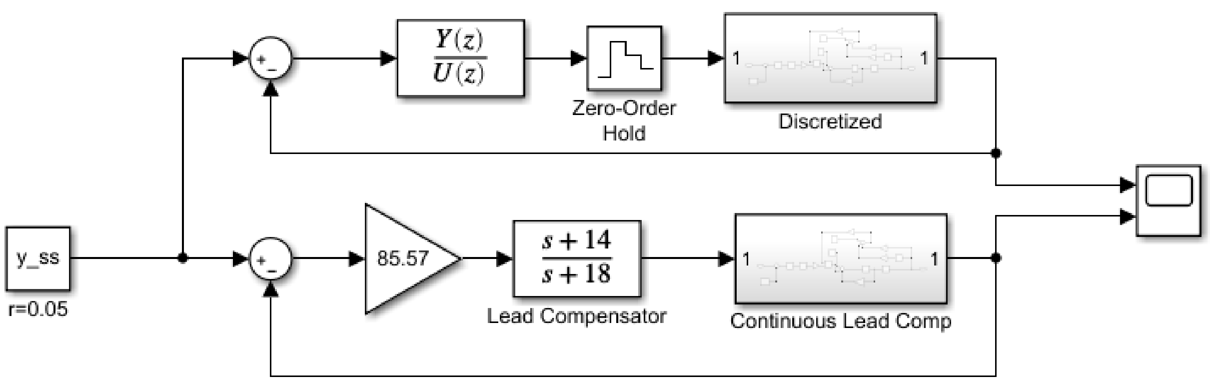


Figure 19: Discrete and Continuous lead compensator systems

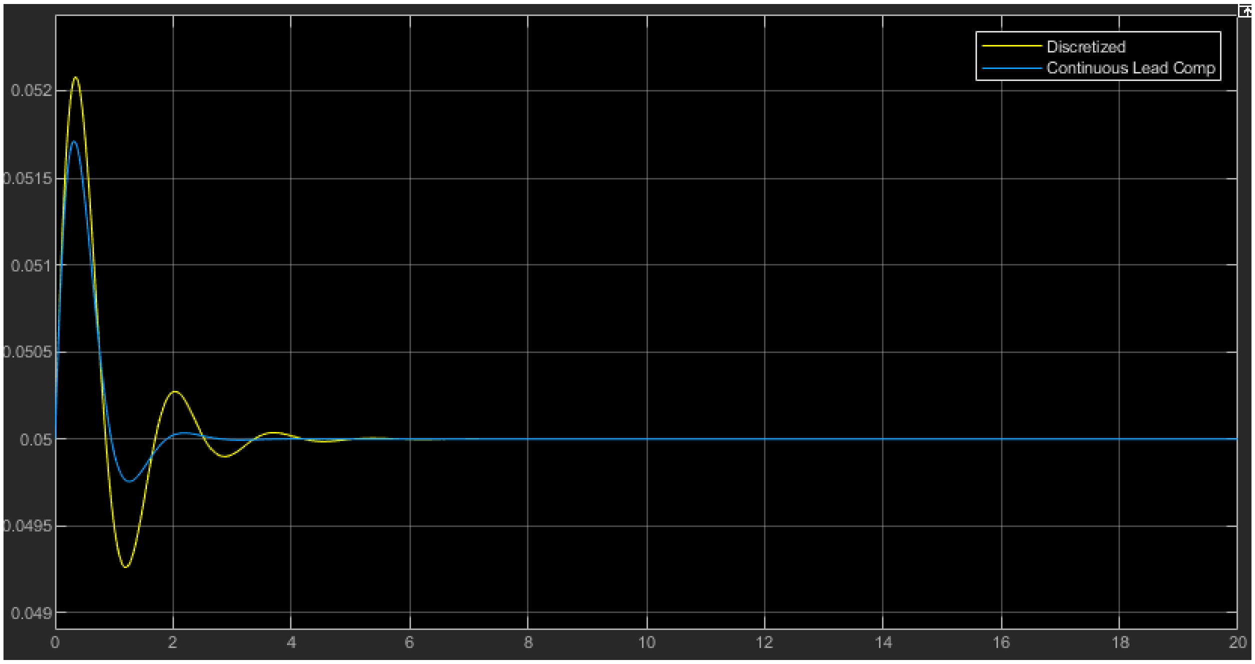


Figure 20: Step response of CT and DT systems, $T_s = 0.01s$

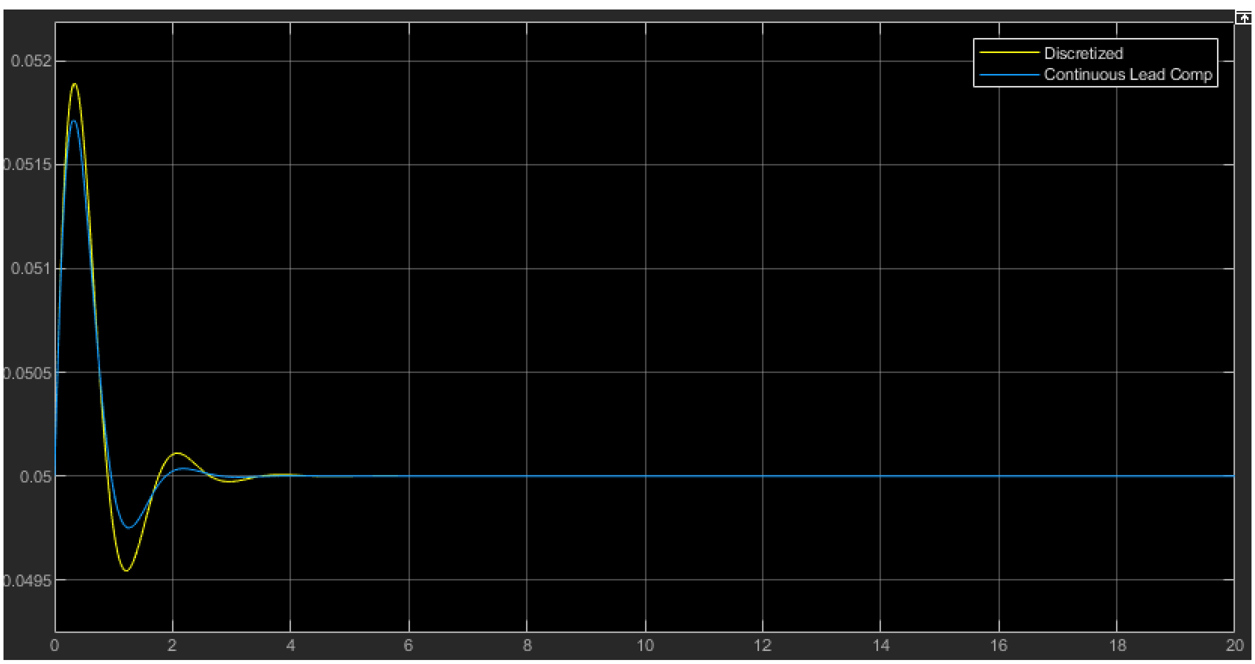


Figure 21: Step response of CT and DT systems, $T_s = 0.005s$

The DT step response gets closer to the CT step response with a smaller sampling rate. The overshoot and the first dip before steady-state is reached are both more in line with the continuous response with the smaller sampling rate.

CONCLUSION

I am very pleased with the results of this project. Although I'm not sure how realistic it is moving the ball about 1.7 meters before it becomes unstabilized, this was a great experience that stressed how important accurate data is and the difficulty of practical applications of control systems. It was great using theory from the course along with our knowledge of Matlab and Simulink to perform this project, and I think it sets us up well for further real-world applications. However, we must note that values are never ideal (as assumed in simulation), so the actual project may yield varying results.

One of the aspects of this project that I am most proud of is the fact that the limiter had very little effect on the lead compensator and on the PID controller, meaning the original specifications were pretty accurate.

REFERENCES

H.K. Khalil. *Nonlinear Systems*, Third Edition. Prentice Hall, Upper Saddle River, New Jersey, 2002