MAE 438/538

Hybrid ODF Maglev Ball System

Part 2

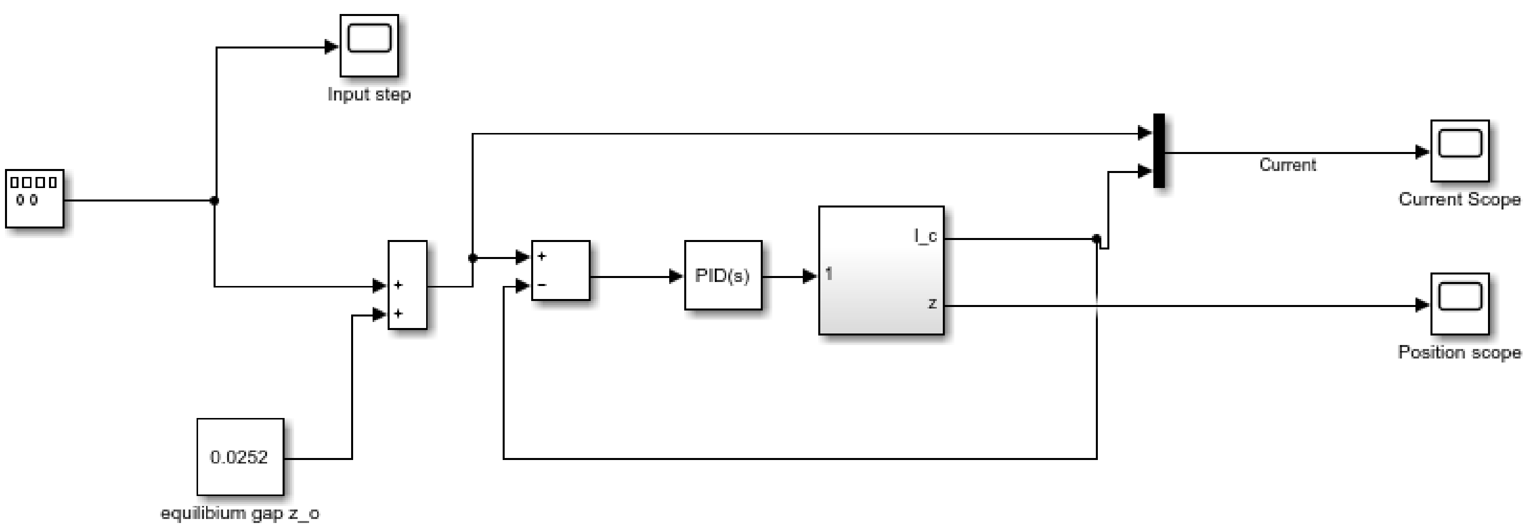
Peter DeTeresa

Email: pdete001@odu.edu

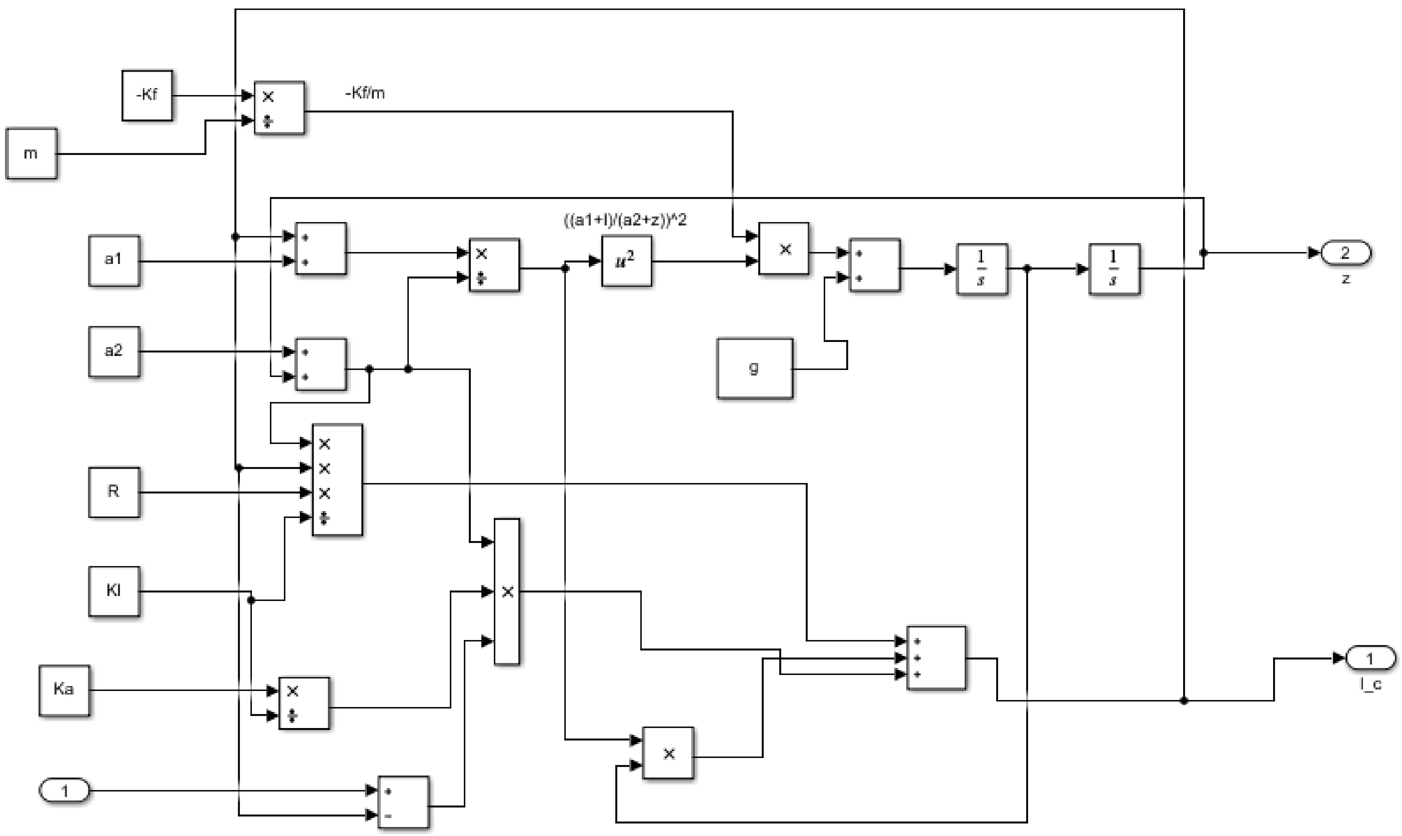
TABLE OF CONTENTS

Part 1 Simulink Model

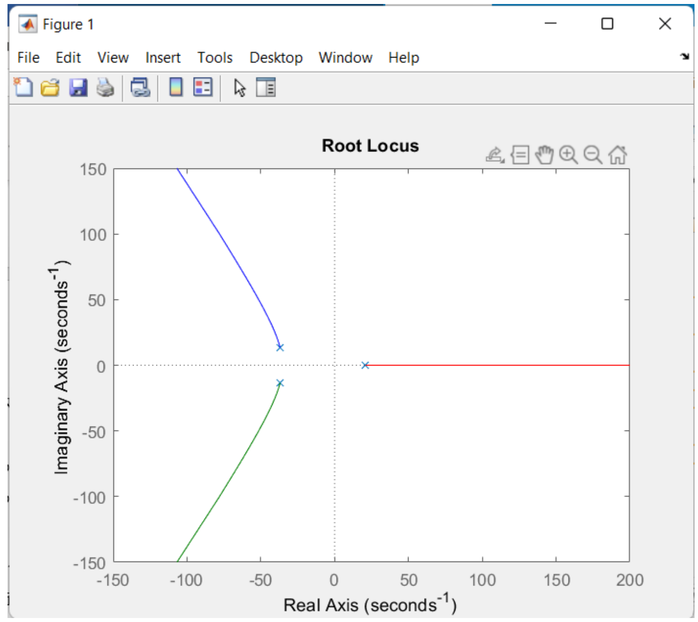
(no controller)



Subsystem



TASK 1 – Nature of open-loop plant



Original Root locus plot indicates system instability since there is a pole to the right of the s-plane

Zeros: None

Poles:

-36.9139 +13.4354i

-36.9139 -13.4354i

20.9019

Gain: -24.379

$\omega_n = \sqrt{1543} = 39.281$

$\zeta = \dfrac{73.83}{\omega_n} = 39.281$

Rise Time: $t_r = \frac{\pi - \phi}{\omega_d} = 0.0781$

Settling time: 0.1310

Settling Min: -0.0158

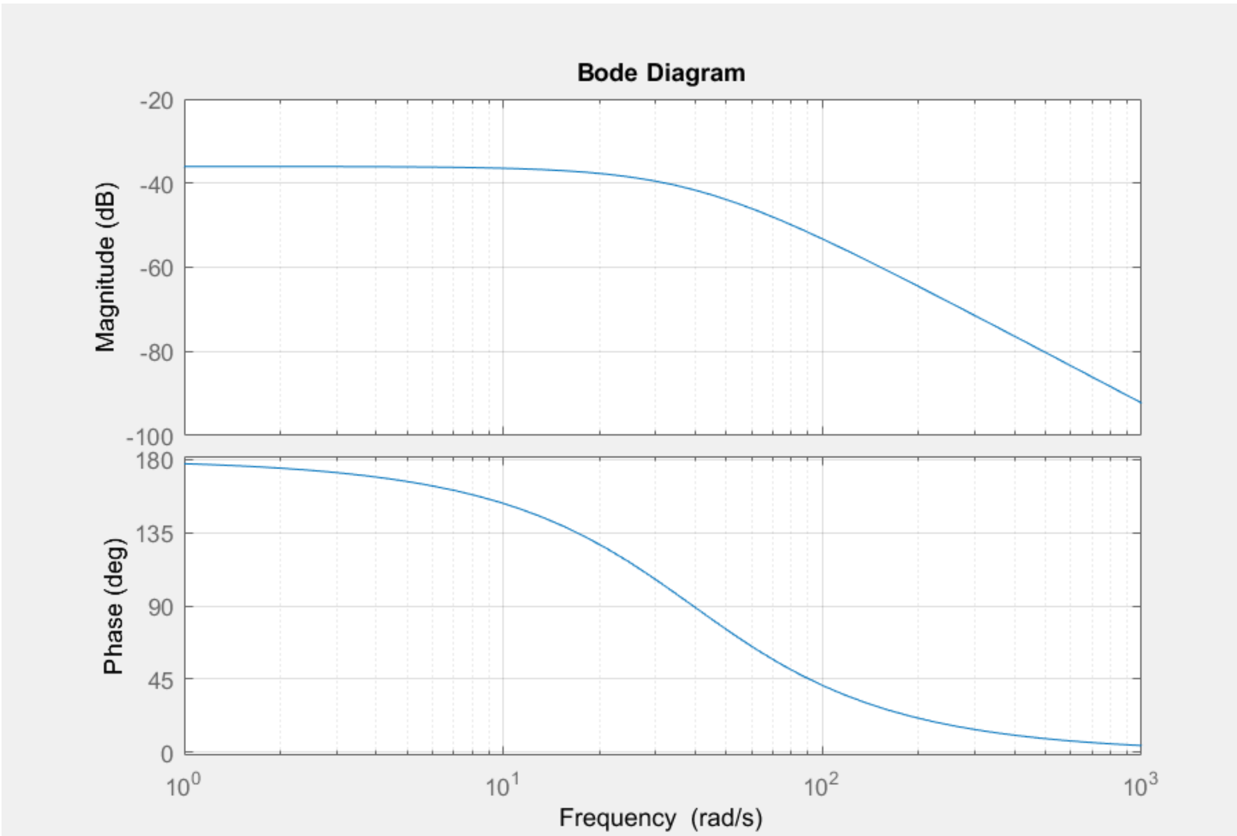
Settling Max: -0.0143

Overshoot: $100 * e^{\left(\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}\right)} = 0.0157$

$\%OS = 1.57$

Peak: 0.0158

Peak time: 0.2233



From the bode plots and the root locus of the first order transfer function for the open-loop model, we can see that the system is unstable.

State-space form:

$$A = \begin{vmatrix} 0 & 1 & 0 \\ 609.3168 & 0 & -0.4176 \\ 0 & 1459.2211 & -52.9358 \end{vmatrix}$$

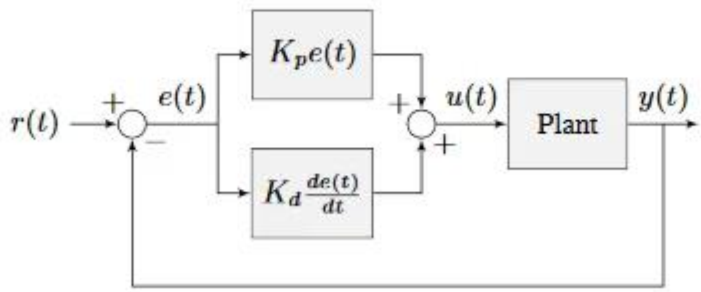
$$B = \begin{vmatrix} 0 \\ 0 \\ 58.3850 \end{vmatrix}$$

$$C = \begin{vmatrix} 1 & 0 & 0 \end{vmatrix}$$

$$D = 0$$

TASK 2 – Classical Compensator

From literature, it would be easiest to design a PD controller for a closed-loop version of the system. It is possible to do this using the linear model and applying Routh stability analysis. After examining several examples of PD controllers, it is clear that they can be used to meet transient requirements. A disadvantage of this controller is that they reduce the damping ratio, leading to an increase in overshoot, which should be accounted for when describing the range of motion of the ball and how it can move reliably without compromising the system's integrity [1].



[2]PD controller block diagram

In this case, the linear transfer function is:

$$G = \frac{24.379}{(s - 20.9)(s^2 + 73.83 + 1543)}$$

The closed loop transfer function for a plant with a PD controller can be denoted by G_{PD}, where

$$tf = \frac{G_{PD}G}{1 + G_{PD}G}$$

And the PD controller has the following form of transfer function:

$$G_{PD} = K_p + K_d s$$

Substituting into the transfer function equation:

$$tf = \frac{(K_p + K_d s)G}{1 + (K_p + K_d s)G}$$

Expanding the denominator of G gives:

$$G = \frac{-24.379}{s^3 + 52.93s^2 - 0.047s - 32249}$$

The numerator and denominator of G can be separated to provide a more comprehensive view of the transfer function

$$tf = \frac{G_{PD} * G_{num}}{\left(1 + G_{PD} * \frac{G_{num}}{G_{den}}\right) G_{den}}$$

$$tf = \frac{G_{PD} * G_{num}}{(G_{den} + G_{PD} * G_{num})}$$

$$tf = \frac{-24.379\left(K_p + K_d s\right)}{s^3 + 52.9358s^2 - K_d(24.379)s - \left(32255 + K_p * 24.379\right)s^0}$$

Routh array criteria:

The first entry for each row must have the same sign. Since the first entry here is 1, then every following entry must be positive

Routh array general setup:

$$\left|\begin{array}{ccccc} s^n & a_0 & a_2 & a_4 & \cdots \\ s^{n-1} & a_1 & a_3 & a_5 & \cdots \\ s^{n-2} & b_1 & b_2 & b_3 & \cdots \\ s^{n-3} & c_1 & c_2 & c_3 & \cdots \\ \vdots & \ddots & \ddots & \ddots & \cdots \\ s^1 & \ddots & \ddots & \ddots & \cdots \\ s^0 & a_n & & & \end{array}\right| \text{ where } b_1 = \frac{a_1 a_2 - a_3 a_0}{a_1}, b_2 = \frac{a_1 a_4 - a_5 a_0}{a_1}, \cdots$$

$$c_1 = \frac{b_1 a_3 - b_2 a_1}{b_1}, c_2 = \frac{b_1 a_5 - b_3 a_1}{b_1}, \cdots$$

Plugging in values for Routh array:

$$\left|\begin{array}{ccccc} 1 & -K_d * 24.379 & a_2 & a_4 & \cdots \\ 52.9358 & -(24.378 + K_p * 24.379) & a_3 & a_5 & \cdots \\ -1290.5K_d & b_1 & b_2 & b_3 & \cdots \\ s^{n-3} & c_1 & c_2 & c_3 & \cdots \\ \vdots & \ddots & \ddots & \ddots & \cdots \\ s^1 & \ddots & \ddots & \ddots & \cdots \\ s^0 & a_n & & & \end{array}\right|$$

Conditions for stability using the Routh array:

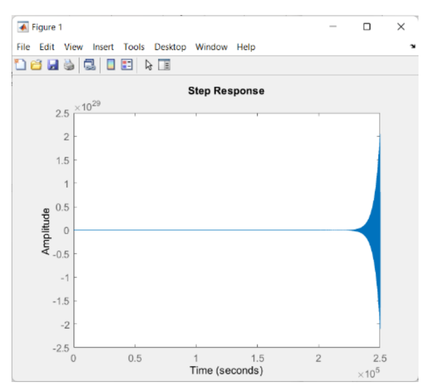
$$K_d < 0$$

$$K_p < -1323.1$$

$$K_p \succ 52.9358 * (K_d - 24.9941)$$

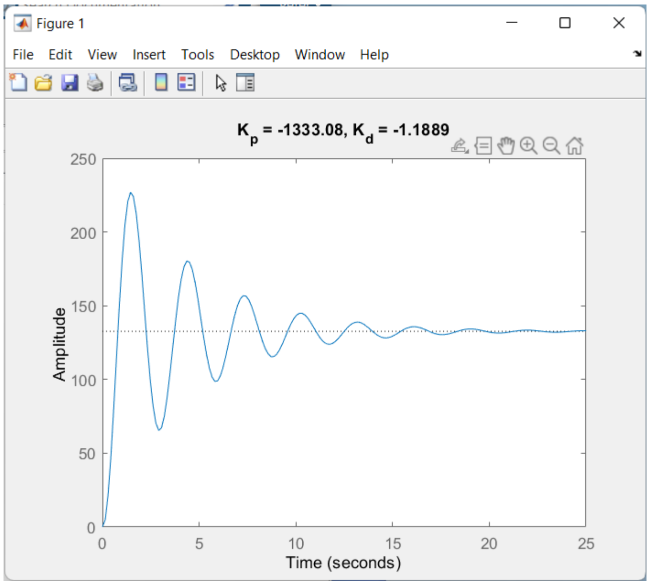
The largest K_p can be is -1323.08 and the least that K_d can be is 5.0604e-05. Plugging these values into Matlab gives the following step function:

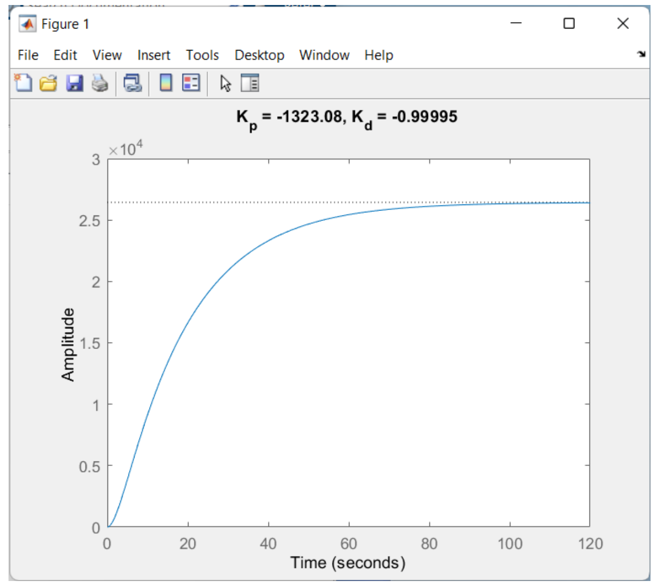
```
G = tf(-24.3794,[1 52.9358 0 -32254.6763])
C_pd1 = tf([Kd Kp],[0 1])
T_pi = feedback(C_pd1*G,1;step(T_pi)
```



The linear model is still unstable, so the values for Kp and Kd need to be reduced further

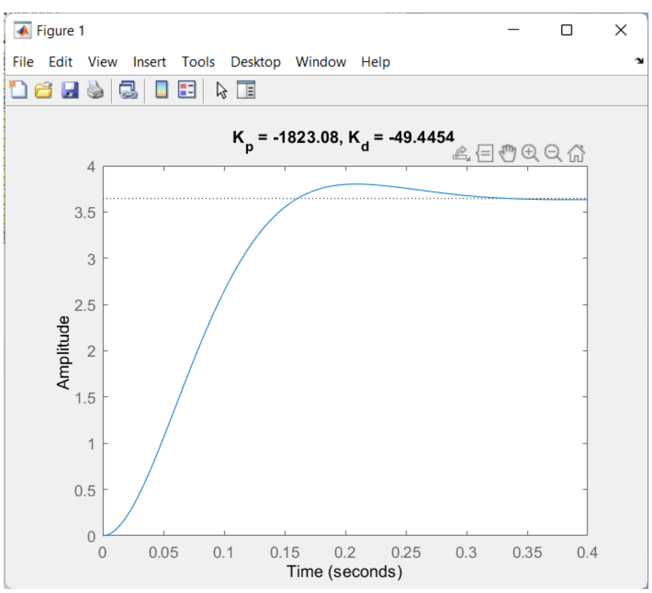
Subtracting 1 from the Kd value produces a stable step amplitude plot. However, the steady state value for the system is over 2.5*10^4, so it is necessary to further reduce Kp.

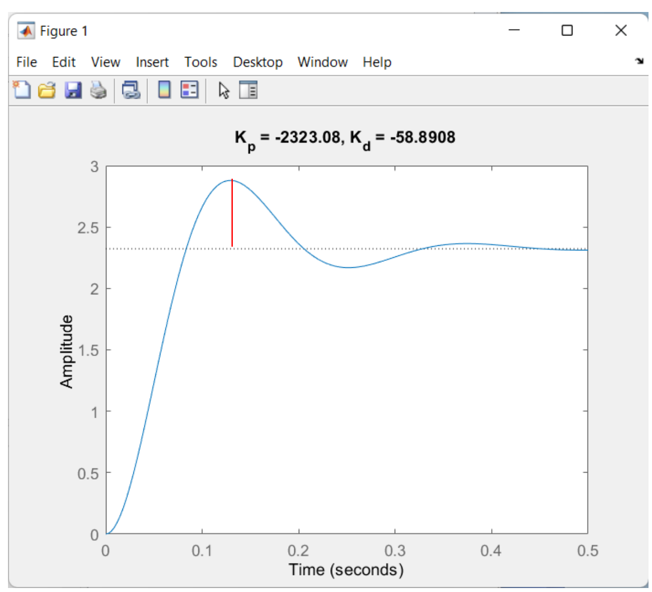




Reducing Kp does lower the steady state value but it also creates more oscillations. Reducing Kp further lowers the steady state value to just over 3.5, at which point the overshoot starts increasing.

$$\boxed{G_{PD} = -1823 - 49.4454s}$$





At this point, to further reduce the steady state value to the desired output value of 1, a full PID controller is necessary since the Integral portion of the controller is responsible for reduction of output error. The equation for a PID controller is:

$$G_{PID} = K_p + K_d s + \frac{K_i}{s}$$

The combined transfer function then has the form:

$$tf = \frac{G_{PID,\mathrm{num}} * G_{num}}{\left(G_{PID,\mathrm{den}} {}^{*} G_{den} + G_{PID,\mathrm{num}} * G_{num}\right)}$$

$$tf = \frac{-24.378\left(K_p s + K_d s^2 + K_i\right)}{s^4 + 52.9358 s^3 - 24.378(K_d)s^2 - \left(32255 + 24.378 K_p\right)s - 24.378 K_i}$$

Applying the Routh Criteria to the PID transfer function gives the following criteria for stability:

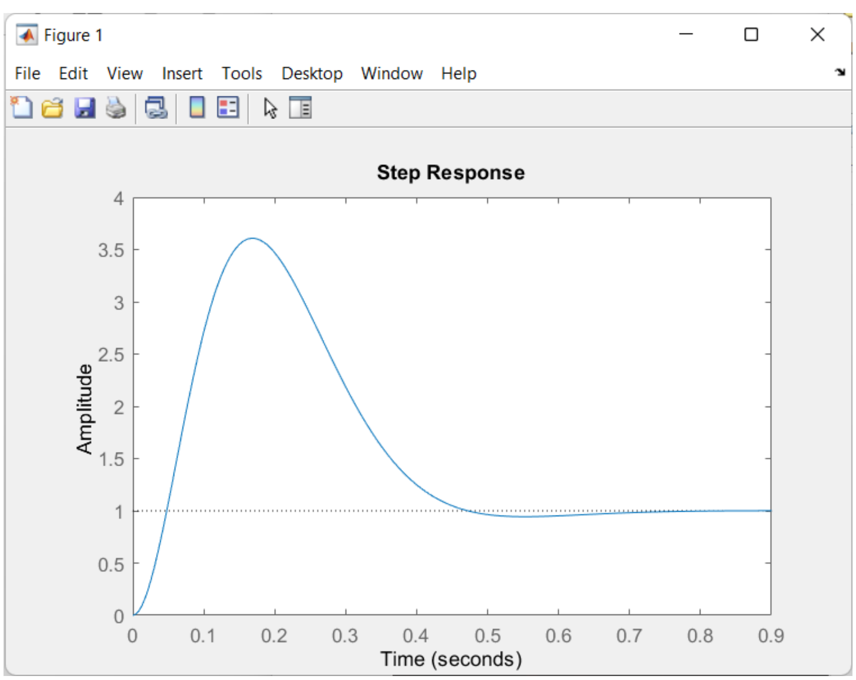
$$K_p < -1323.08$$

$$K_i < \frac{\left(-52.9358 * K_d + K_p + 1323.08\right)\left(1323.08 + K_p\right)}{114.946}$$

$$K_p = -2700$$

$$K_i = -1900$$

$$K_d = -50$$



RiseTime: 0.031471

SettlingTime: 0.58425

SettlingMin: 0.94376

SettlingMax: 3.6077

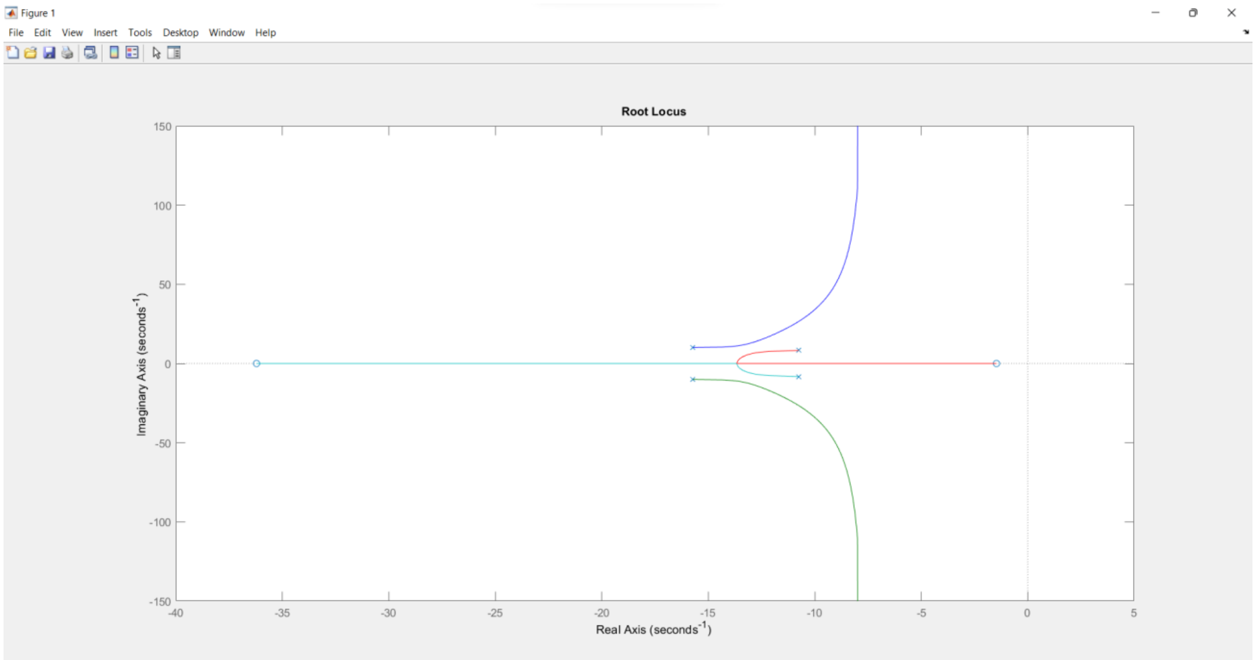
Overshoot: 260.77

Undershoot: 0

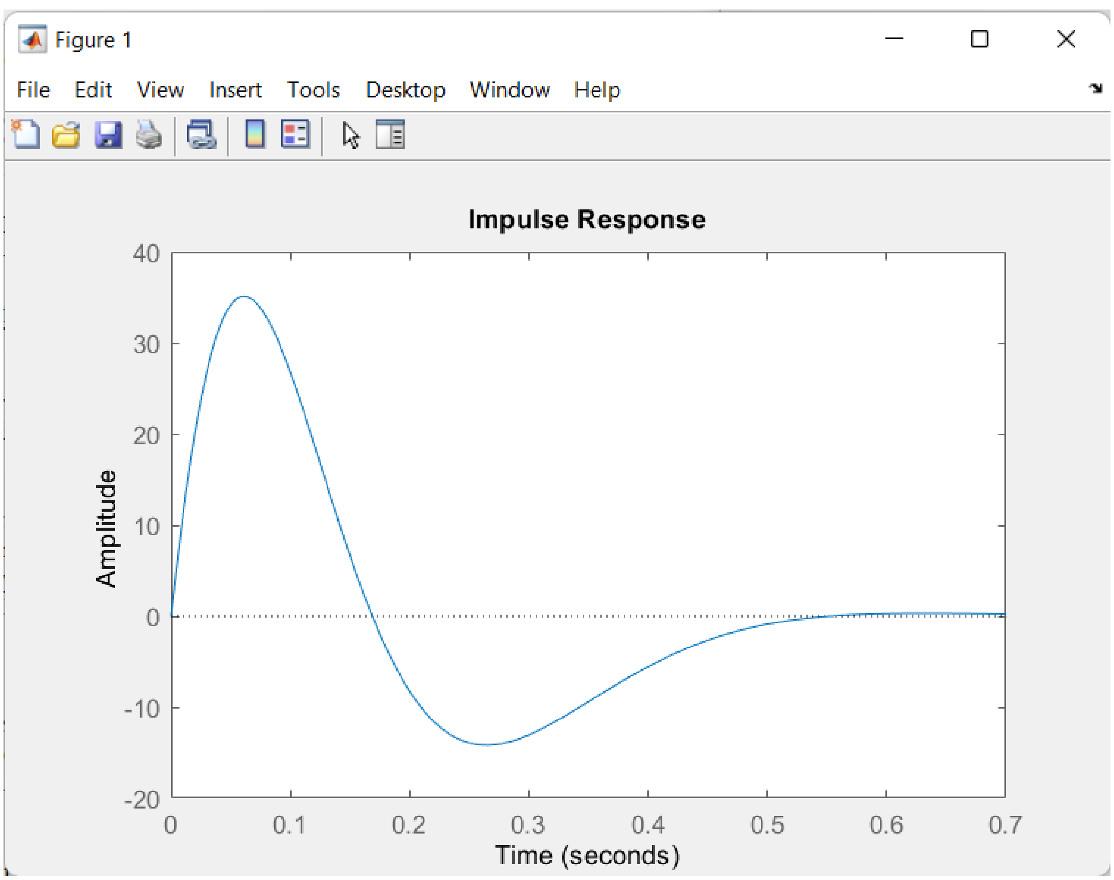
Peak: 3.6077

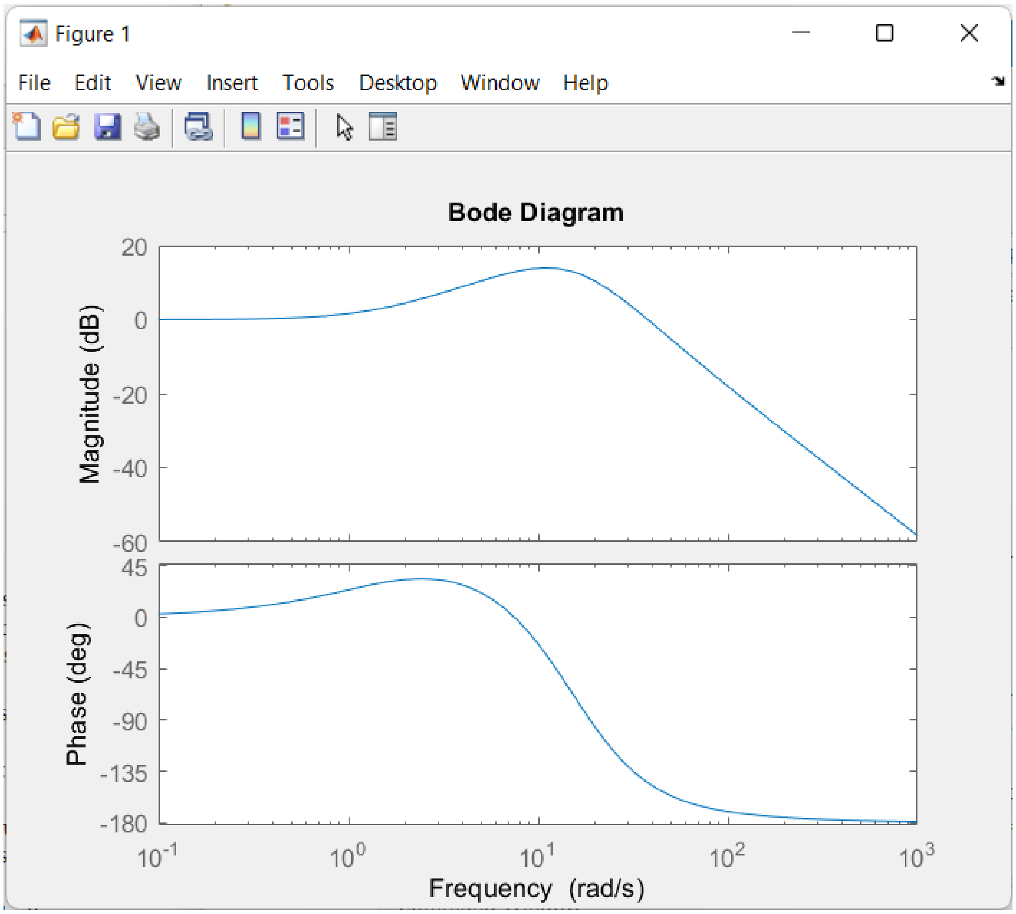
PeakTime: 0.16709

Root locus of the Transfer function of the PID controller:



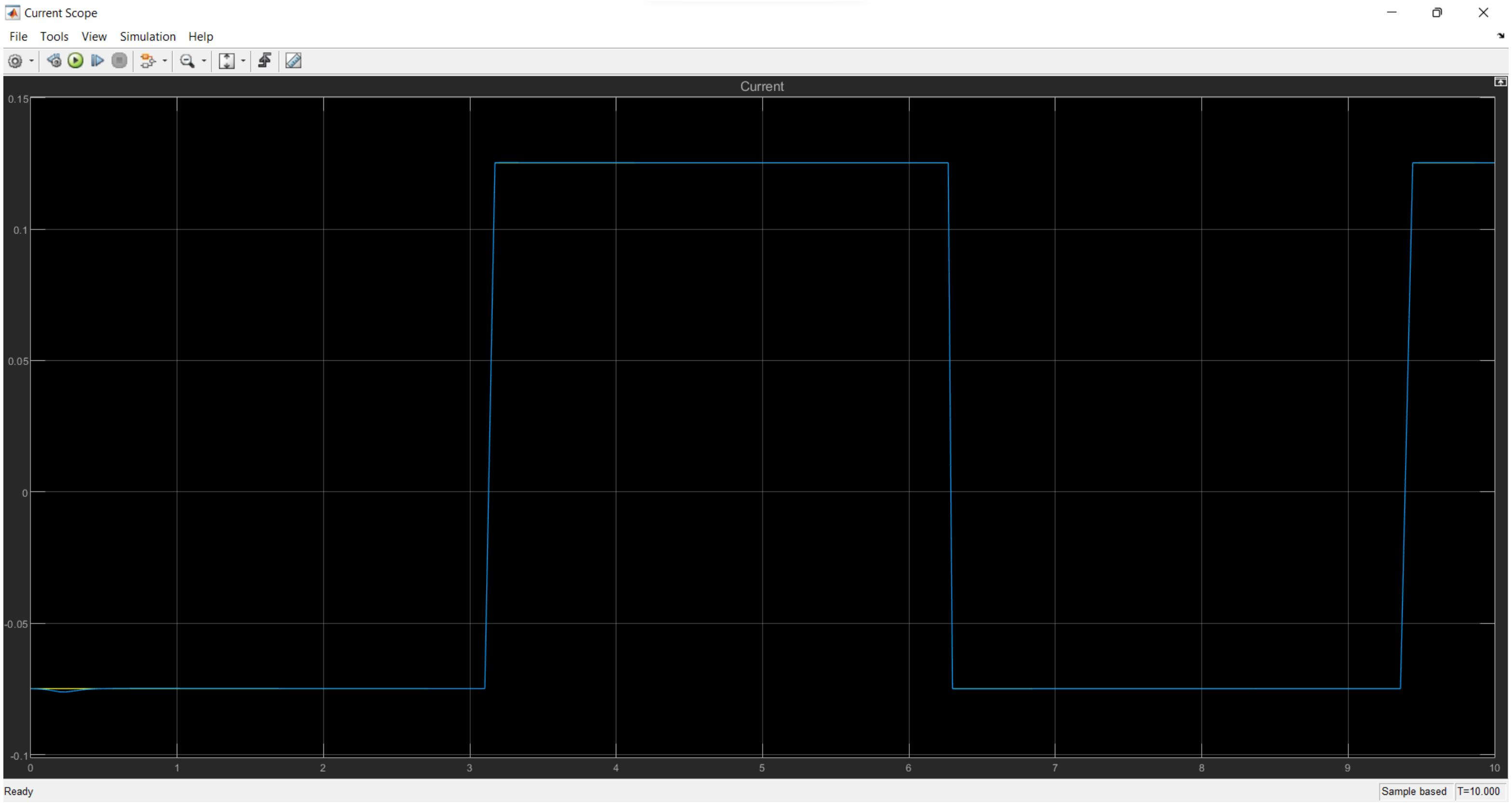
Impulse Response of transfer function:

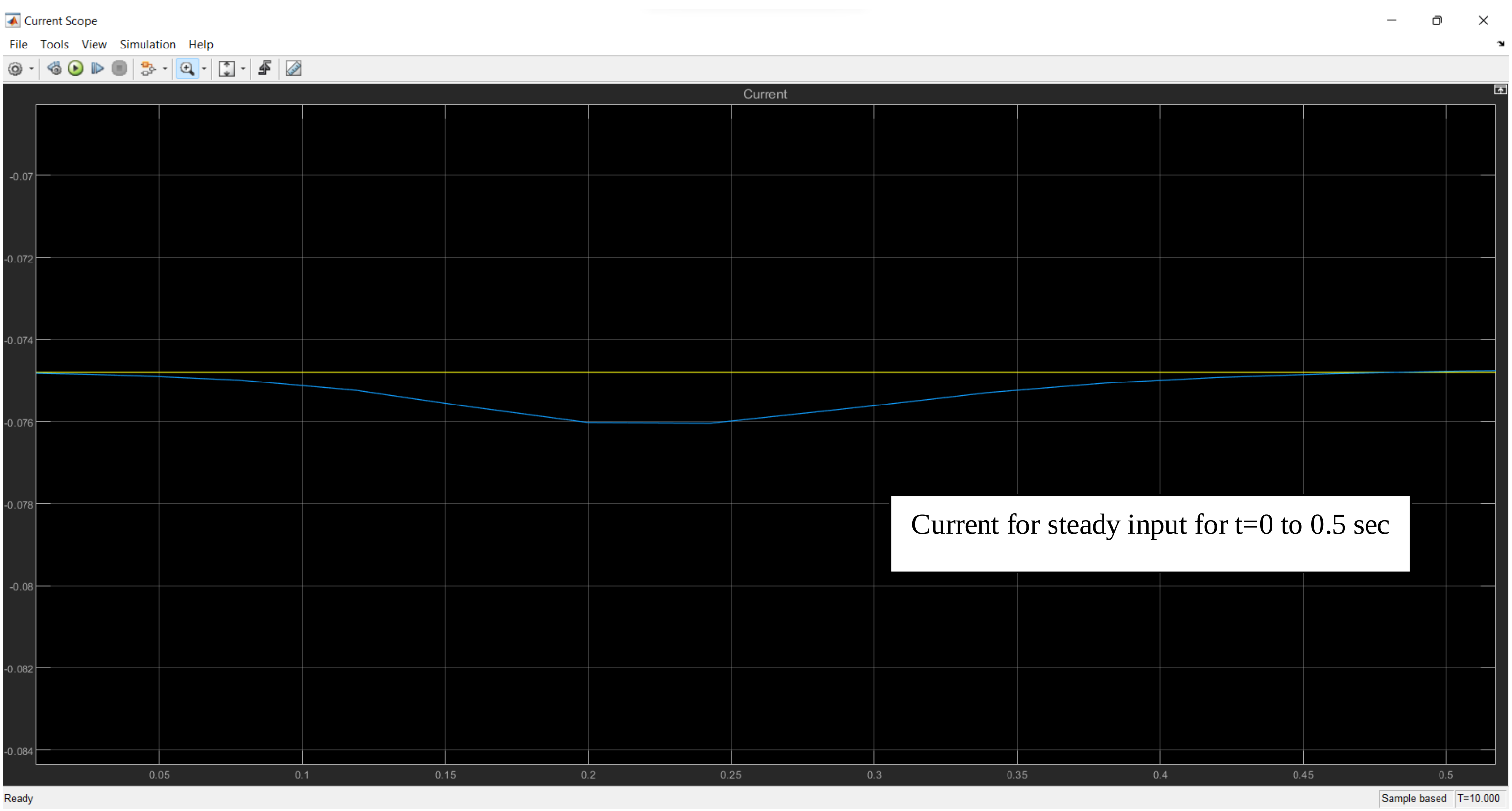




After looking at the results from the step input and impulse input plots along with the root locus and bode plots, it is clear that the system is now stable but could use some optimization. It is possible that the response could be improved with successive loop closure or a lag-lead compensator, but I think that the PID controller is the right decision and to reduce the overshoot and minimum settling time requires more analysis. I intend to look into this before starting part 3 of the project, but for now I will focus on implementing the designs into my Simulink model.

Simulink Model with PID Controller





Current for steady input for t=0 to 0.5 sec

References

[1] Libretexts. "3.3: Pi, PD, and Pid Controllers." *Engineering LibreTexts*, Libretexts, 5 Mar. 2021, https://eng.libretexts.org/Bookshelves/Industrial_and_Systems_Engineering/Book%3A_Introduction_to_Control_Systems_(Iqbal)/03%3A_Feedback_Control_System_Models/3.3%3A_PI%2C_PD%2C_and_PID_Controllers.

[2] "Introduction to PID." *FIRST Robotics Competition Documentation*, https://docs.wpilib.org/en/stable/docs/software/advanced-controls/introduction/introduction-to-pid.html.

[3] Mekky, Ahmed E.. "Modeling, Identification, Validation and Control of a Hybrid Maglev Ball System" (2012). Master of Science (MS), Thesis, Mechanical & Aerospace Engineering, Old Dominion University, DOI: 10.25777/jz35-fk25 https://digitalcommons.odu.edu/mae_etds/139