Lab 4 – Ball and beam

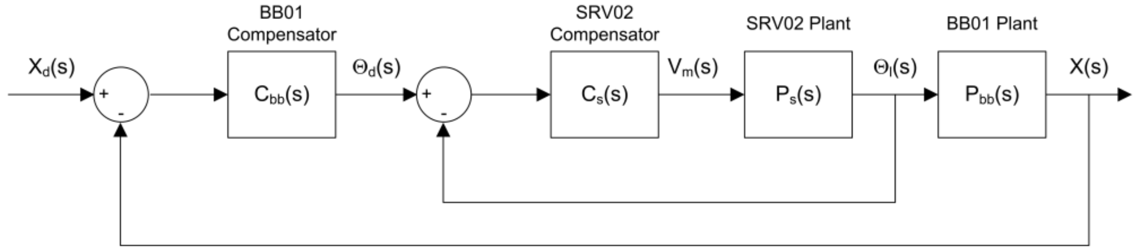
Submitters:

Or Shaul

Saray Sokolovsky

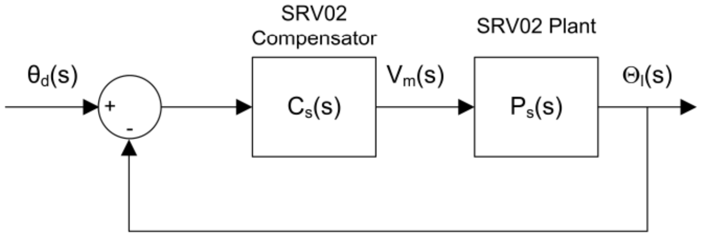
Introduction

The goal of this laboratory is to stabilize a ball at a desired position along a beam using the proportional derivative (PD) control method. A cascade control system is designed to fulfill a specific set of requirements. The following diagram illustrates the closed-loop system used to control the ball's position:

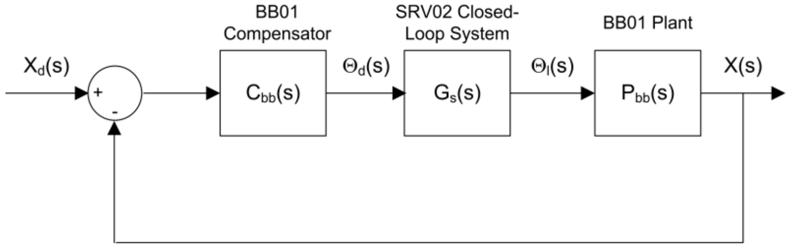


We will break the system into two parts:

- The closed loop:



- the closed loop as a black box along with the rest of the system:



3 Pre Lab

1. Find K_{bb} by simplifying the expression given in Equation 2.18. Then, evaluate it using the system parameters given in [7].

   From equation 2.18 - $\frac{d^2}{dt^2}x(t) = \frac{m_b g \sin\theta_l(t) \mathrm{r_{arm}}\, \mathrm{r}_b^2}{L_{beam}(m_b r_b^2 + J_b)}$

   For little angles (close to zero) the sine function can be approximates by - $sin\theta_l(t) = \theta_l(t)$.

   Using the hint - $J_b = \frac{2m_b r_b^2}{5}$, and lump the coefficient parameters of $\theta_l(t)$ into a single parameter K_{bb} we are getting:

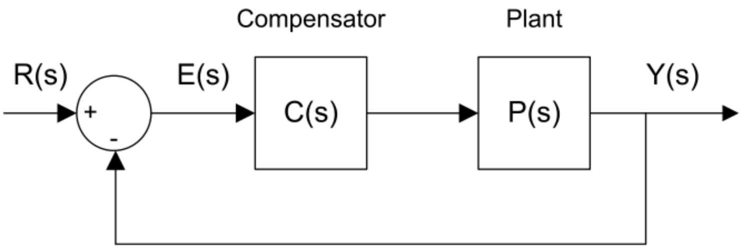
$$K_{bb} = \frac{m_b g \mathrm{r_{arm}}\, \mathrm{r}_b^2}{L_{beam}(m_b r_b^2 + \frac{2m_b r_b^2}{5})} = \frac{g \mathrm{r_{arm}}}{L_{beam}(1 + \frac{2}{5})} = \frac{5g\mathrm{r_{arm}}}{7L_{beam}}$$

   From Quanser Inc. Ball and beam user manual:

Symbol	Description	Matlab Variable	Value
	Mass of ball beam module		0.65 kg
	Calibration base length		50 cm
	Calibration base depth		22.5 cm
L_{beam}	Beam length	L_beam	42.55 cm
	Lever arm length		12.0 cm
r_{arm}	Distance between SRV02 output gear shaft and coupled joint	r_arm	2.54 cm
	Support arm length		16.0 cm
r_b	Radius of ball	r_ball	1.27 cm
m_b	Mass of ball	m_ball	0.064 kg
K_{bs}	Ball position sensor sensitivity	K_BS	-4.25 cm/V
V_{bias}	Ball position sensor bias power		± 12 V
V_{range}	Ball position sensor measurement range		± 5 V

$$K_{bb} = \frac{5g\mathrm{r_{arm}}}{7L_{beam}} = \frac{5 \cdot 9.8 \cdot 2.54}{7 \cdot 42.55} = 0.4178 \left[\frac{1}{ms^2c}\right]$$

2. Find the steady-state error of the the Ball and Beam system given by the $P_{bb}(s)$ transfer function. The system is shown in Figure 3.1. The compensator is unity:



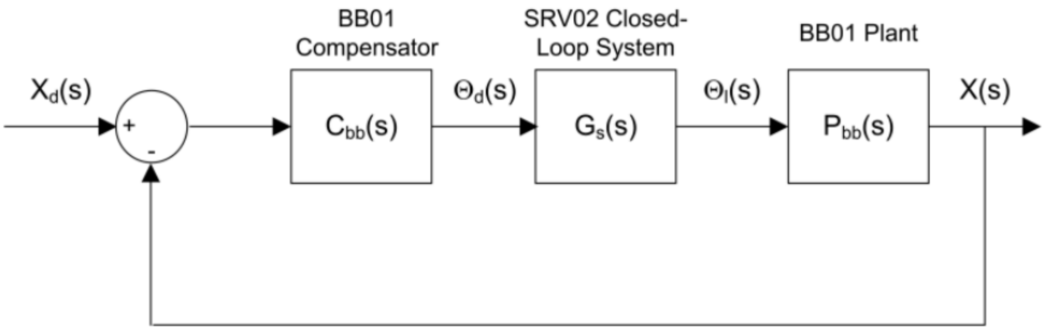
$$c(s) = 1, R(s) = \frac{R_0}{s}, G(s) = P_{bb}(s) = \frac{K_{bb}}{s^2}$$

   where R0 is the step amplitude. Note that in this calculation the SRV02 dynamics is to be ignored and only the BB01 plant is to be considered.

   The formula for steady state error:

$$e_{ss} = \lim_{s\to 0} \frac{sR(s)}{1 + G(s)} = \lim_{s\to 0} \frac{s\frac{R_0}{s}}{1 + \frac{K_{bb}}{s^2}} = "\frac{R_0}{\infty}" = 0$$

3. Using Figure 2.6, find the closed-loop transfer function of the BB01 system with proportional control $C_{bb}(s) = K_c$

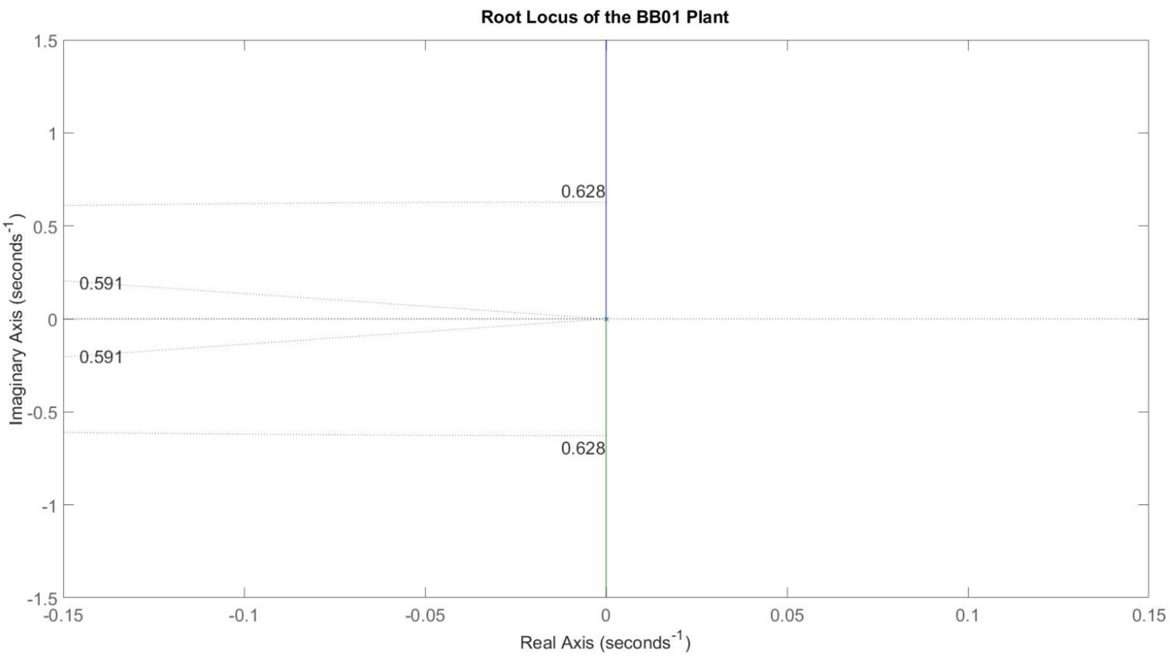


SRV02 Closed Loop System is: $G_s(s) = \frac{P(s)}{1+P(s)}$

The open loop transfer function of BB01: $L(s) = C_{bb}(s) \cdot G(s) \cdot P_{bb}(s) = K_c \cdot \frac{K_{bb}}{s^2}$

Therefore, the closed loop transfer function of BB01: $T(s) = \frac{L(s)}{1+L(s)} = \frac{K_c K_{bb}}{K_c K_{bb}+s^2}$

4. Plot the root locus of the BB01 plant $P_{bb}(s)$ Describe how the poles behave as K_c goes to infinity.
   The root locus plot:



We see that the poles move on the imaginary axis and as K_c goes to inf the complex roots becomes bigger.

5. Find the natural frequency and damping ratio required to achieve the time-domain specifications of the Ball and Beam plant given in Section 2.2.1

   We can use the OS formula given in the pdf: $O\% = 100 \cdot e^{-\frac{\pi\xi}{\sqrt{1-\xi^2}}}$ for O%=10% and $t_s =$ 3.5 sec the value of $\xi = \pm 0.59$ and for the settling requirements we are getting : $\omega_n = 1.45$

6. After you plot the root locus of the BB01 plant $P_{bb}(s)$ describe where the poles should be to satisfy the desired response specifications:

We have two options to satisfy the requirements for the placement of the poles:

1. Arc intersection
2. Diagonal lines

This is clear from the RL, we got $0 < |\xi| < 1$ therefore we would have 2 complex poles. So we want the poles to be on the left side of the real axis assuming stability.

7. Discuss the response if the poles lie beyond the radius circle along the diagonal lines, i.e. away from the imaginary axis. Also, comment on what happens if the poles of the system lie inside the diagonal lines along the radius circle, i.e. moving towards the real axis. Make references to its effects on the settling time and overshoot of the response.

   Poles that are far from the center along the diagonal cause the natural frequency to blow up which means settling time is faster but increases overshoot. If the poles are placed on the arc moving to the real axis, moving them around changes the regime of damping that we are in so the system will have lower overshoot and slower response.

8. Based on the root locus obtained in question 6 previously, can the specifications of the Ball and Beam system be satisfied using a proportional controller? Discuss.

   It can be seen from the root locus plots that such a positioning cannot be obtained. A proportional controller alone may not be sufficient to meet all the desires specifications. On our case all the poles lies on the img axis therefore to satisfy it $\omega_n$needs to be 0 and it's not the case.

9. Assume that a traditional PD controller ($C_{bb}(s) = K_c(s+z)$) is used in the system given in Figure 2.6. Find the BB01 error transfer function
   The error transfer function is given by:
$$E(s) = \frac{R(s)}{1 + C(s)P(s)} = \frac{R_0 s}{s^2 + K_c K_{bb}(s+z)} = \frac{R_0 s}{s^2 + K_c K_{bb} s + K_c K_{bb} z}$$
10. Find the steady-state error of the BB01 closed-loop system with the traditional PD controller. Can the steady state error requirement in 2.26 be satisfied?
    From the final value theorem, we get that:
$$e_{ss} = \lim_{s\to 0} sE(S) = \lim_{s\to 0} \frac{R_0 s}{s^2 + K_c K_{bb} s + K_c K_{bb} z} = 0$$
    Yes. We want $|e_{ss}| \leq 0.005\ m$ and we got 0.
11. Based on the expressions found in Equations 2.40 and 2.41, evaluate numerically the zero location and gain needed to satisfy the specifications.
$$K_c = \frac{2\xi\ \omega_n}{K_{bb}} = \frac{2 \cdot 0.59 \cdot 1.45}{0.4178} = 4.1$$
$$z = \frac{\omega_n^2}{K_{bb} K_c} = \frac{1.45^2}{0.4178 \cdot 4.1} = 1.227$$

12. Find expressions for the zero location, z, and the compensator gain, Kc for the practical PD controller to satisfy ωn and ζ in Section 2.2.1 and the desired filter cutoff frequency in 2.43.

Then, evaluate numerically the pole time constant, zero location and gain needed to satisfy the specifications.
We want $\omega_f = 6.28\frac{rad}{sec}, K_{bb} = 0.4178$
This can be solved with coefficient comparison and and as given in the prelab:

$$\omega_f = \frac{\omega_n^2}{zT_pK_{bb}K_c}$$

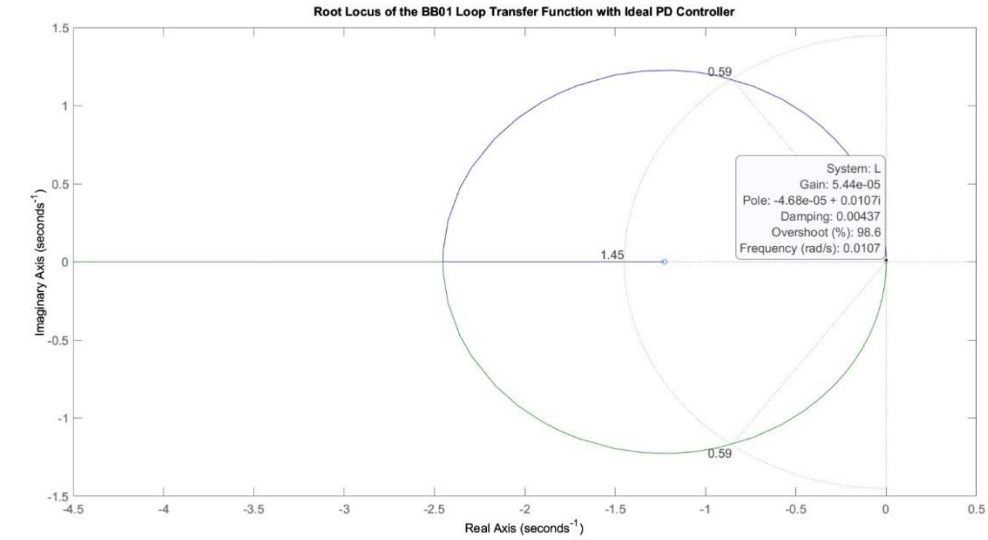
$$K_{bb}K_c(z+\omega_f) = \omega_n^2 + \frac{2\xi\ \omega_n}{T_p}$$

So:

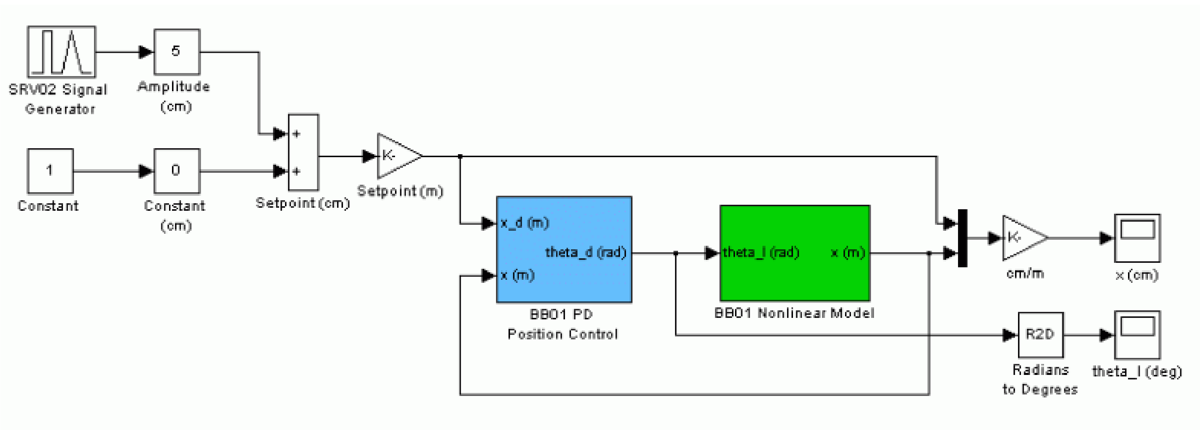
$$K_c = 3.5[cm^{-1}]$$
$$T_p = 0.23[sec]$$
$$z = 1.28[cs^{-1}]$$

4. Post Lab Ball on Beam
4.1 Cascade Control with an Ideal PD Controller
4.1.1 Simulation with No Servo Dynamics

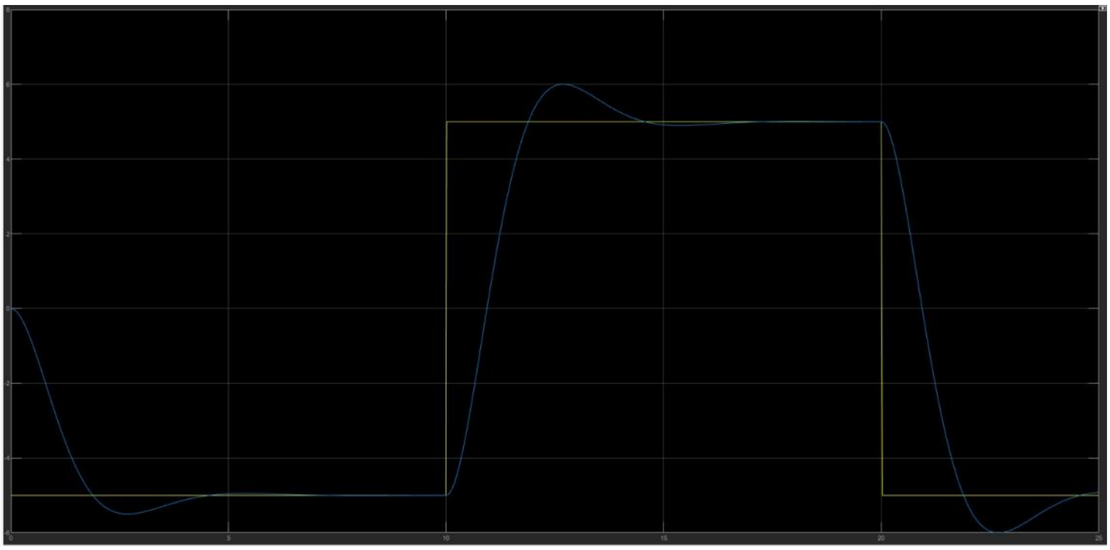
The root locus plot as obtained in MATLAB:

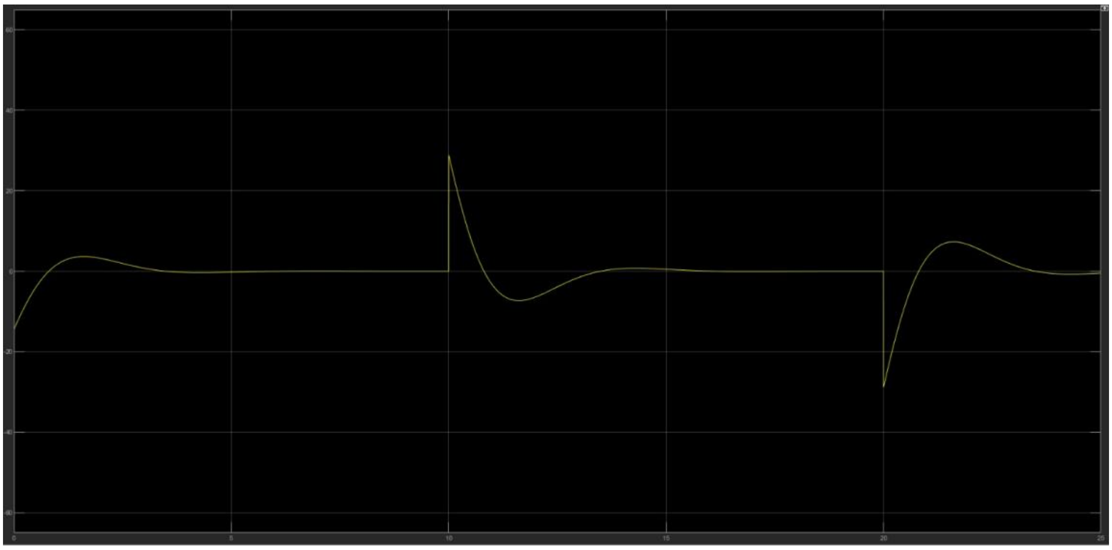


For the following:

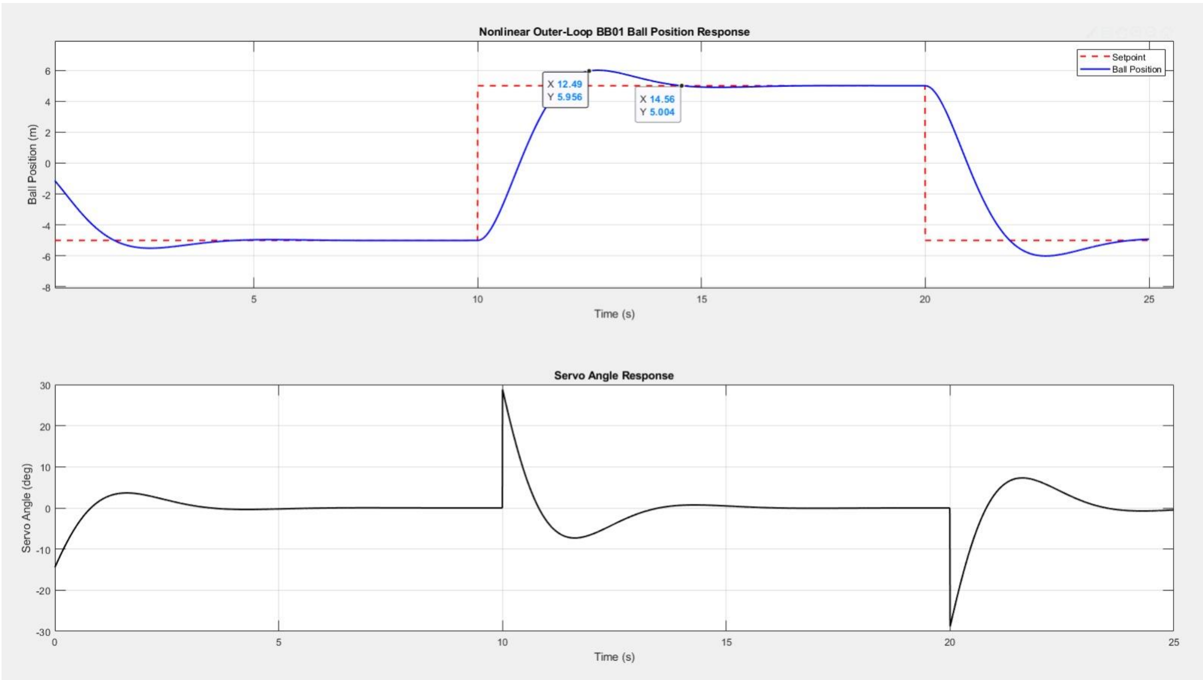


8. The simulation:





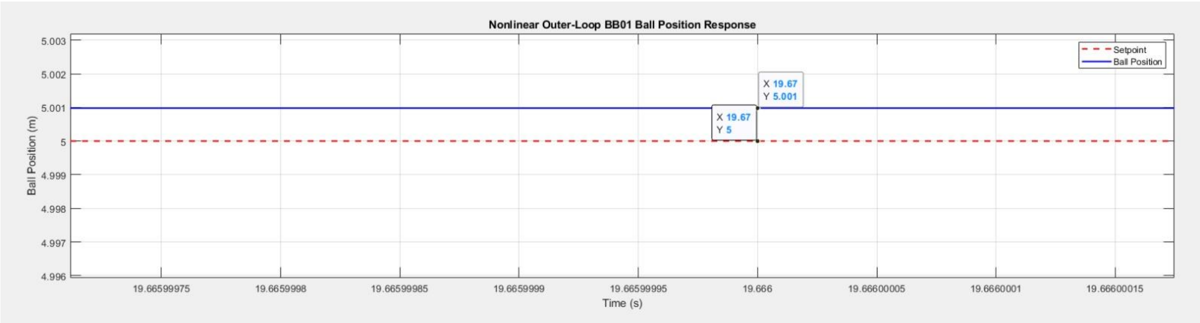
9+10. the steady-state error, the settling time, and the percent overshoot of the simulated response:



The OS% : $OS = \frac{5.956-5}{5} * 100 = 19.3\%$

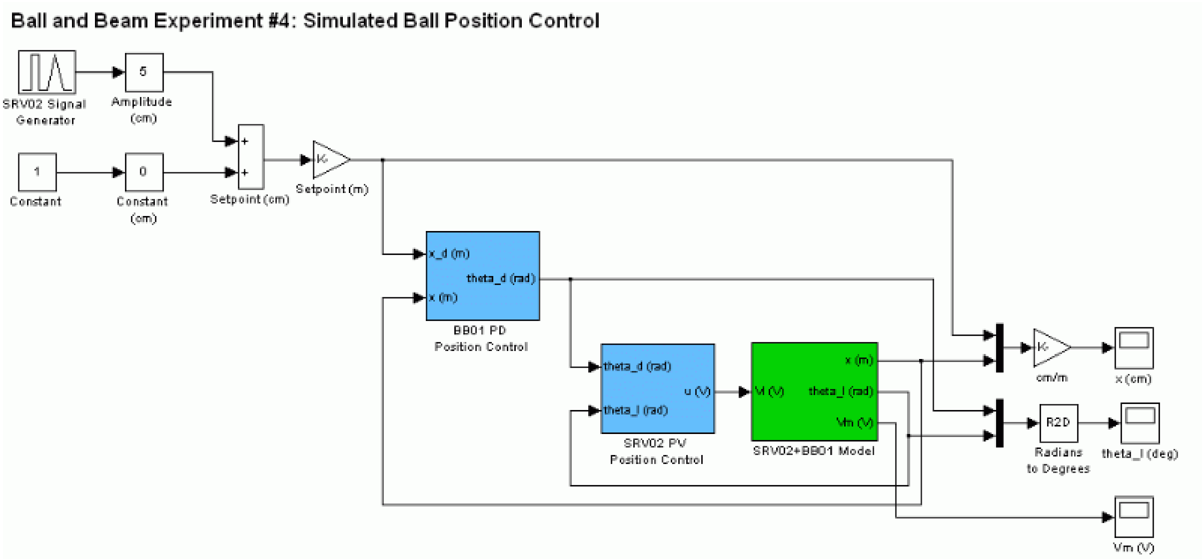
The settling time: $14.56 - 10 = 4.56\ sec$

11. For the error, we need to look at the end of the rectangular pulse. Below is the plot at the end of the pulse:

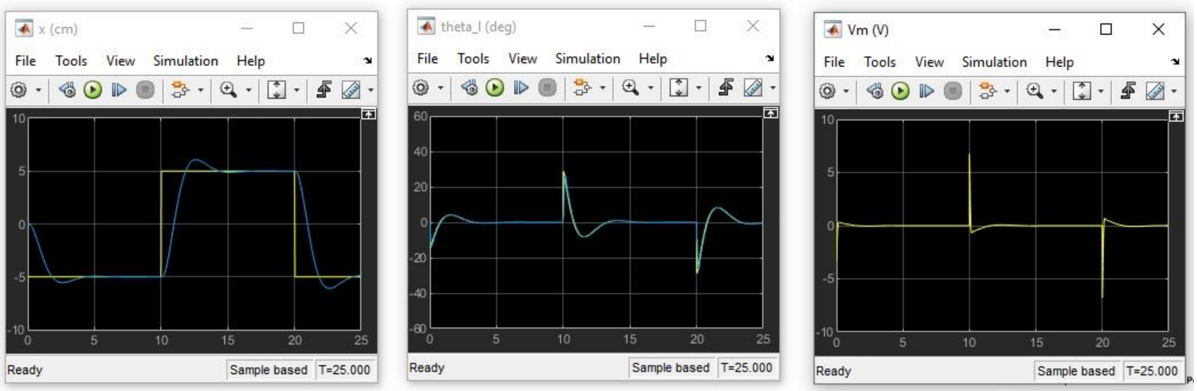


The error is 0.001

4.1.2 Simulation with Servo Dynamics

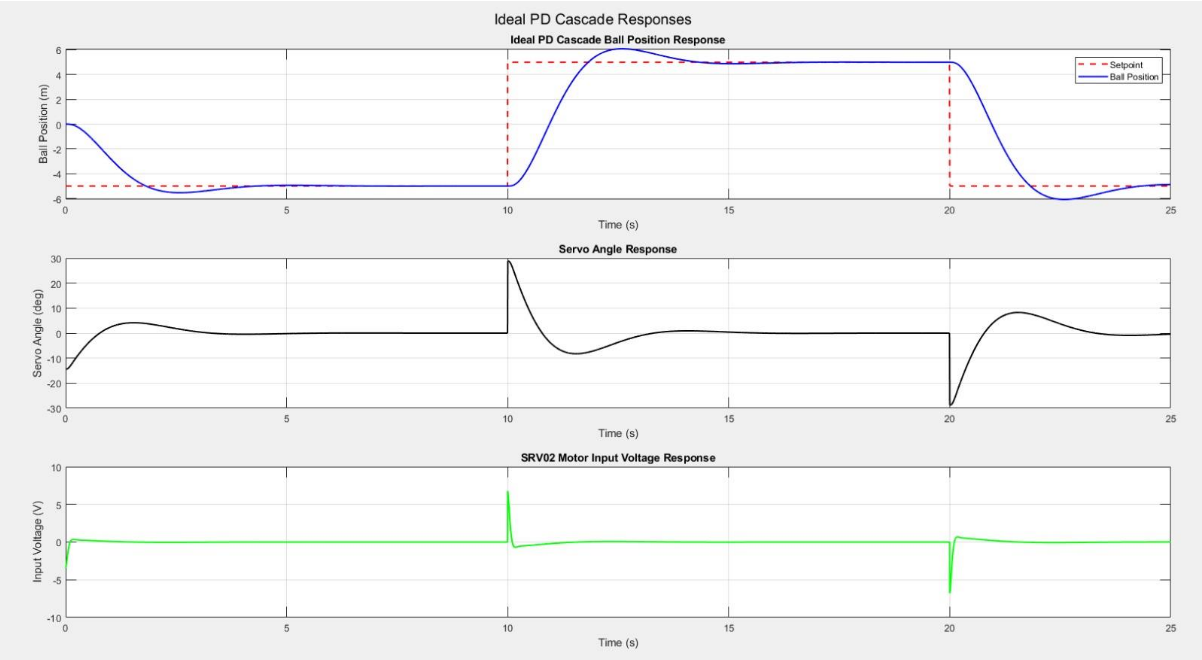


9. The system is built as per the specifications mentioned Following are the plots along with the system in Simulink-

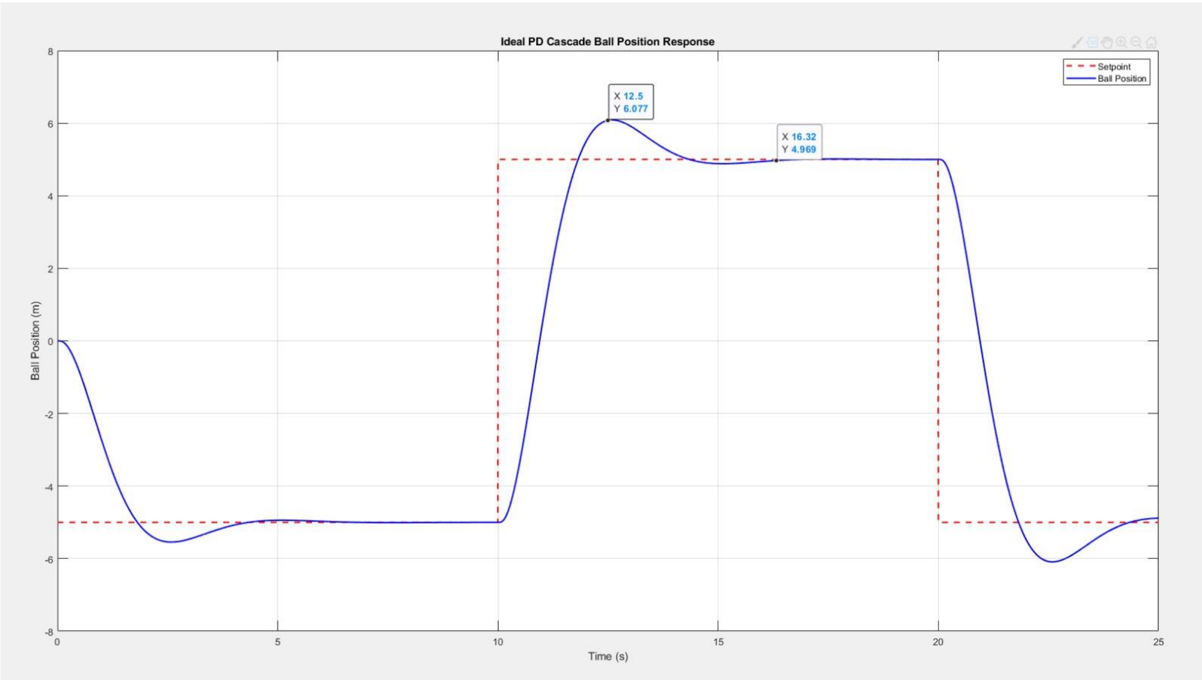


As above, you have the ball position, the servo angle and the input voltage response

10. Below are the plots obtained in matlab after exporting the data:

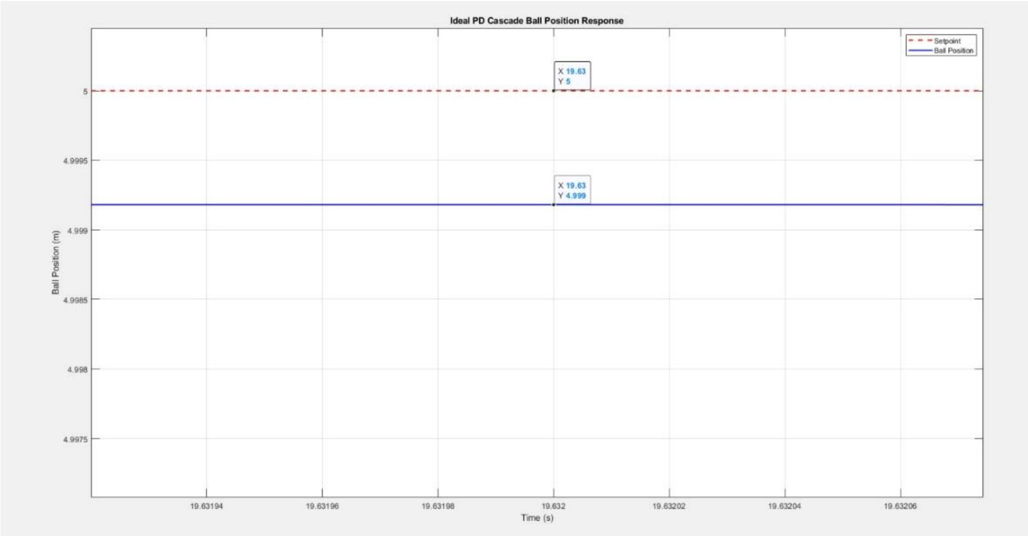


11. Measure the steady-state error, the settling time, and the percent overshoot of the ideal PD cascade control response –



The OS% - $OS = \frac{6.007-5}{5} * 100 = 20.14\%$

The settling time : $16.32 - 10 = 6.32\ sec$

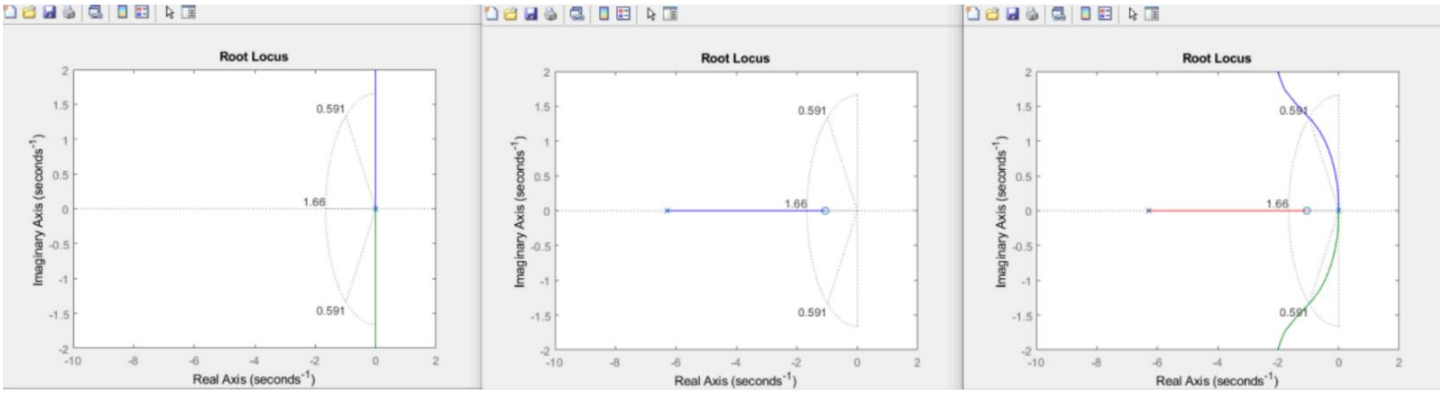


The error is 0.001

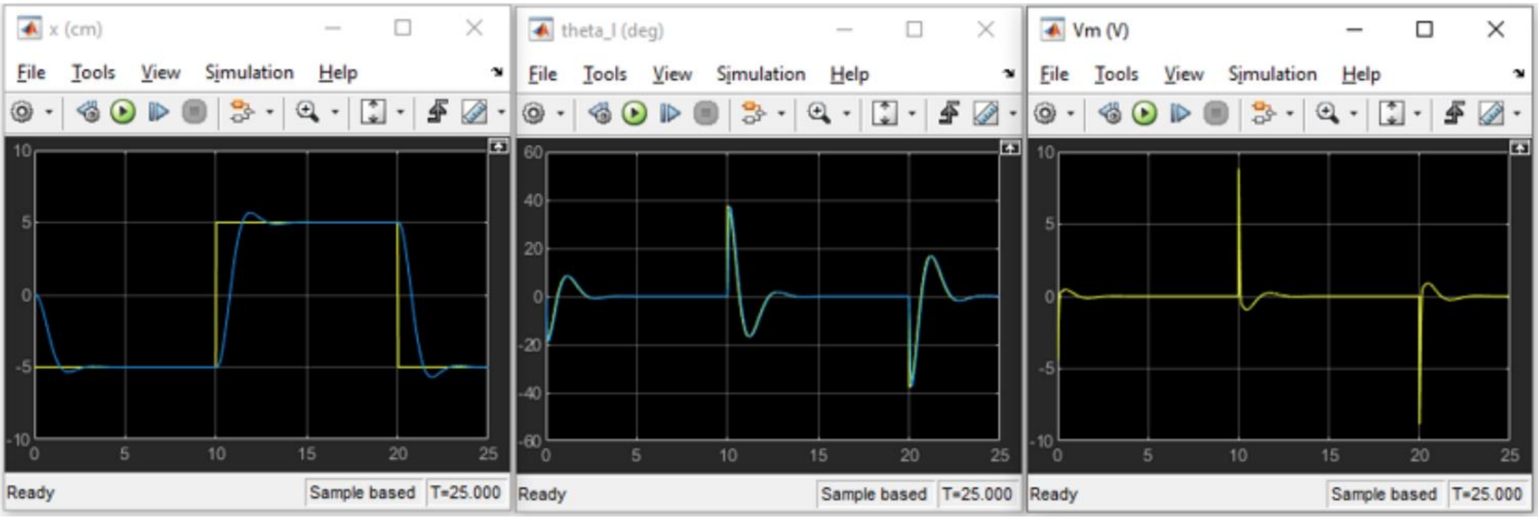
4.2 Cascade Control with Practical PD Controller and Servo Dynamics

4.2.1 Simulation with Practical PD Controller

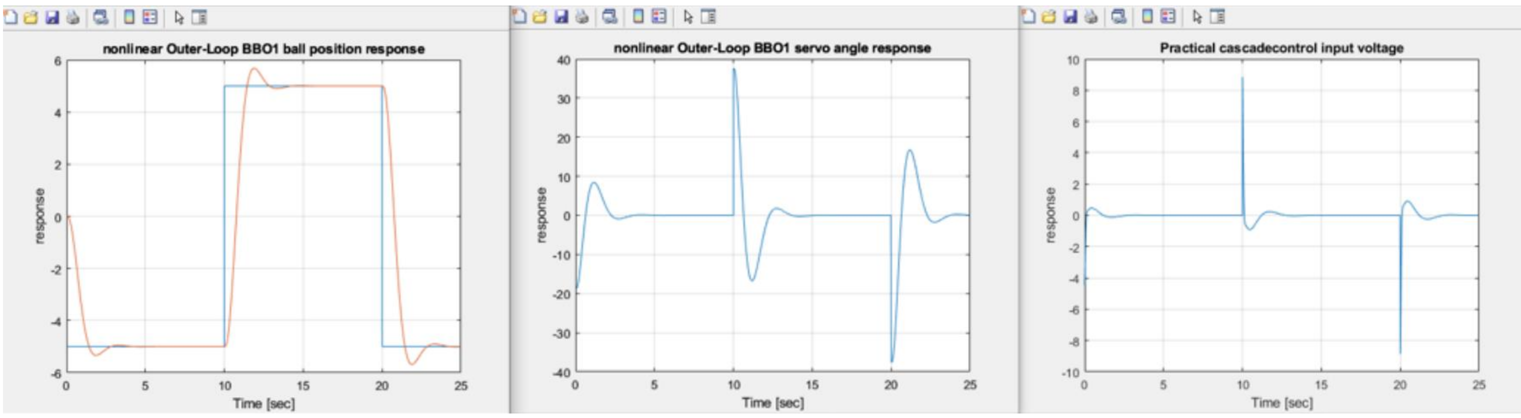
6. The root locus:



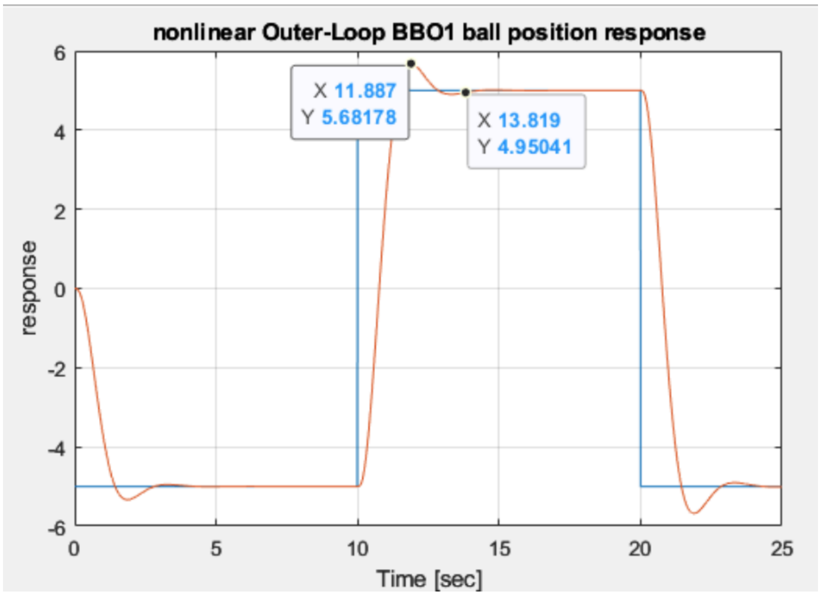
8. The simulation:



9.In MATLAB:



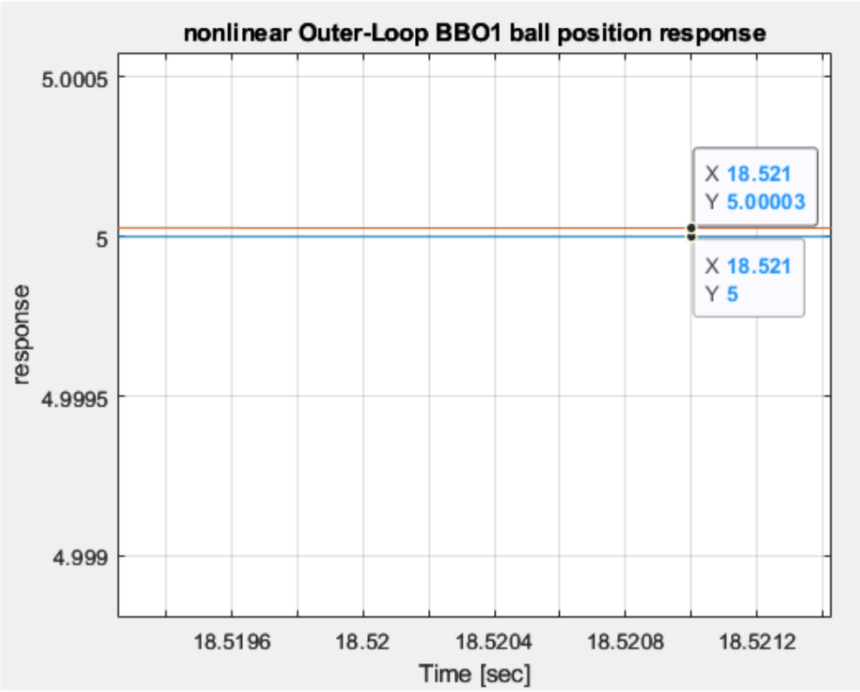
10.



The OS% - $OS = \frac{5.6817-5}{5} * 100 = 13.634\%$

The settling time: $3.819\ sec$

We see that the specifications of an actual PD Controller are better than that of an ideal PD controller. For calculating the error, we look at the end of the rectangular pulse:



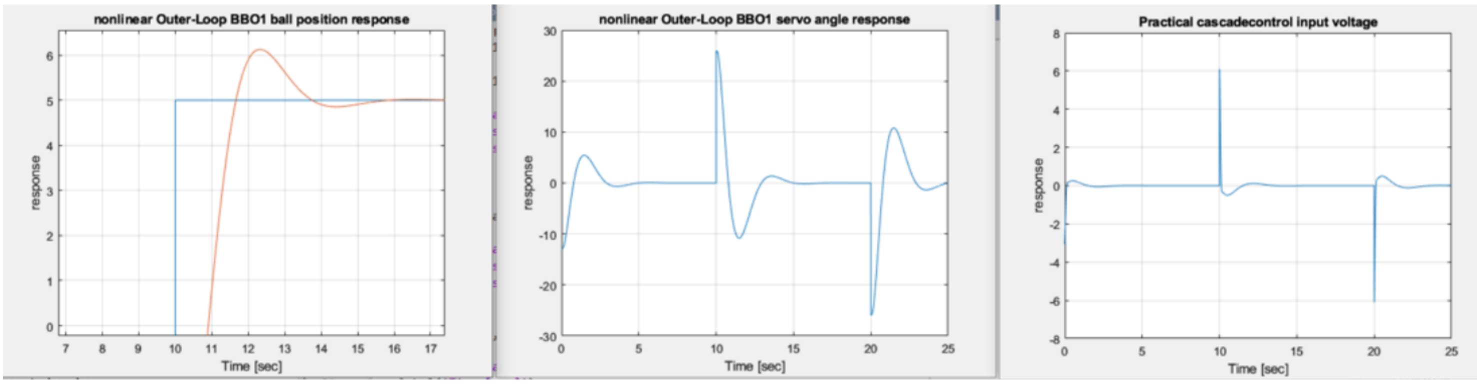
The error is 0.00003

11. The %OS is still not the required one.

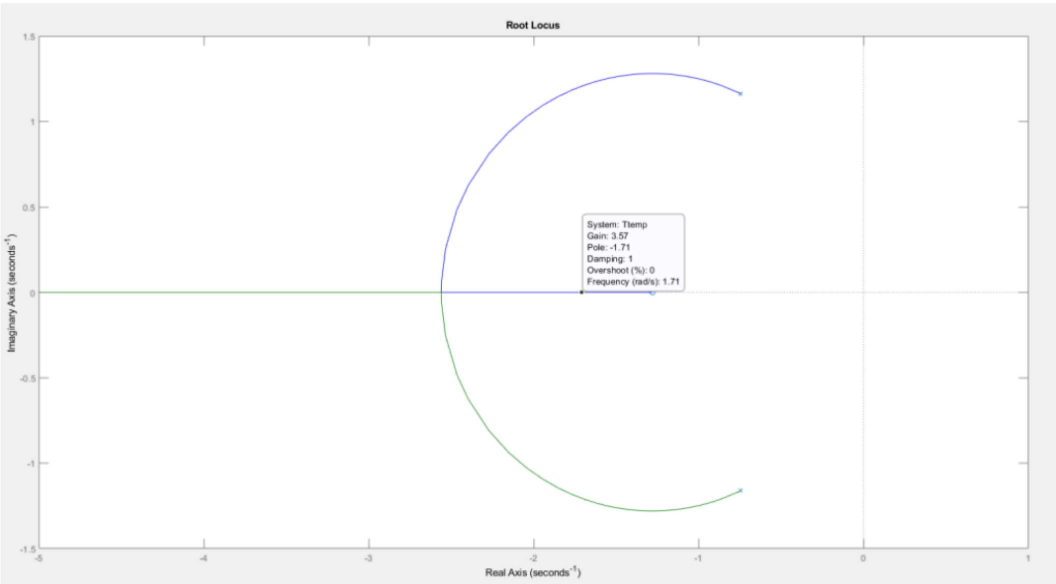
12. Following is the MATLAB code we implemented:

```
zeta = -log(PO_bb/100)*sqrt(1/((log(PO_bb/100))^2+pi^2));
wn = -log(c_ts*(1-zeta^2)^(1/2))/(zeta*ts_bb);
Tp = 1/(wf - 2*zeta*wn);
z = wn*wf/(-wn+wf*wn*Tp+2*wf*zeta);
Kc = wn^2/(Tp*K_bb*z*wf);
C_bb = tf([z+wf z*wf],[1 wf]);
```

And the plots:

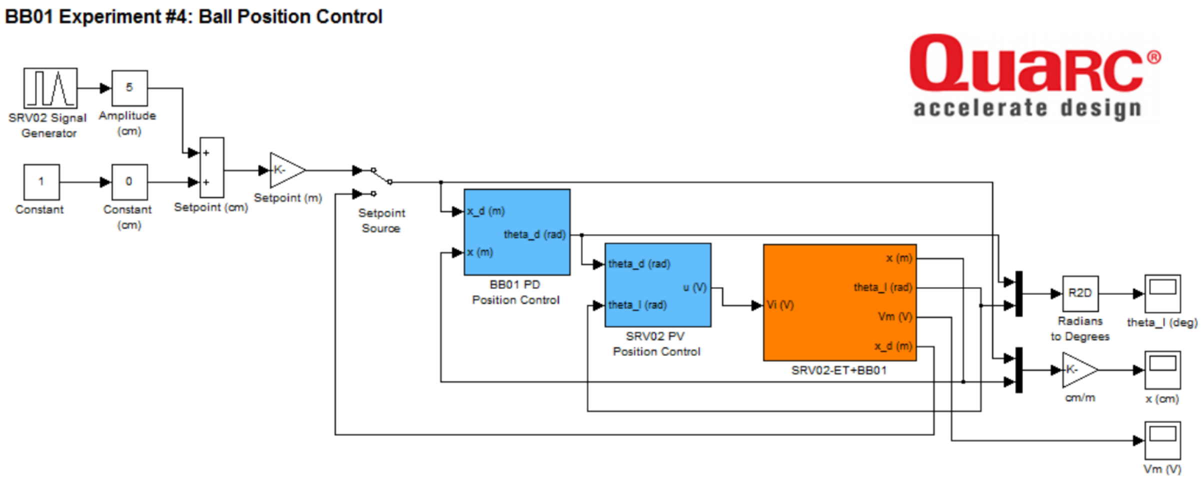


The root locus to show stability:

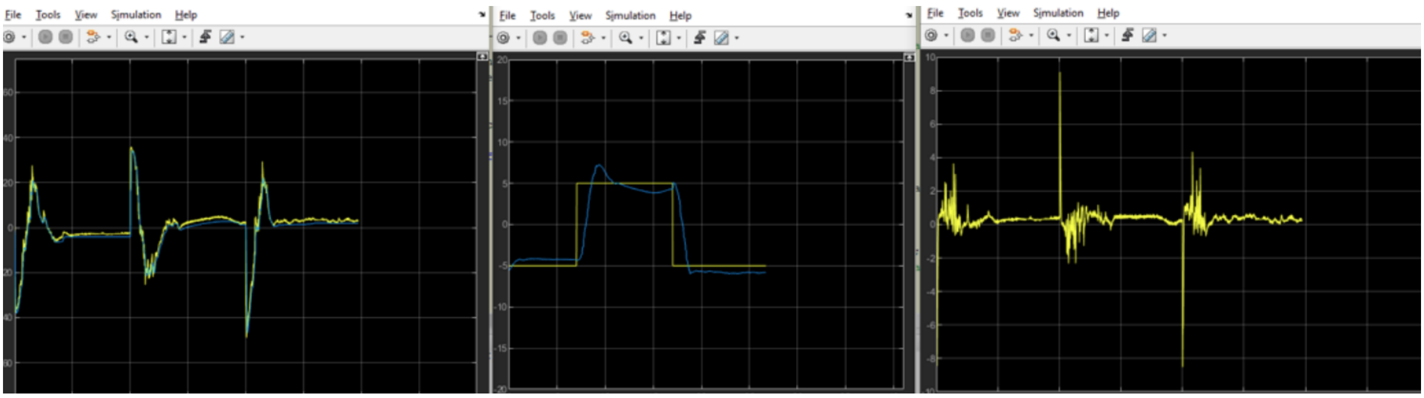


We see very similar results for the gain and an approximate result for the pole if we compare to our Root Locus results with the results of section 12 in the prelab report.

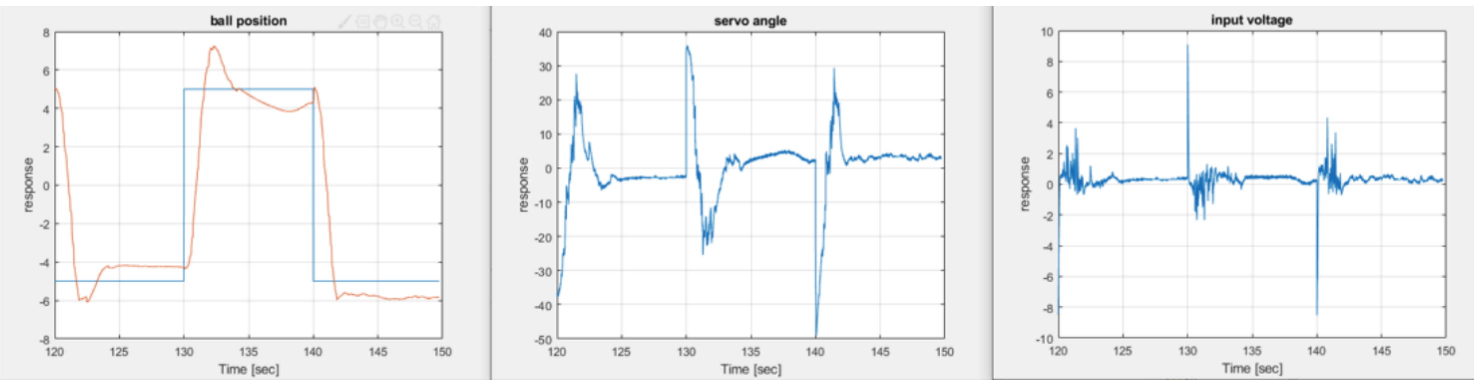
4.2.2 Implementation with Practical PD Controller



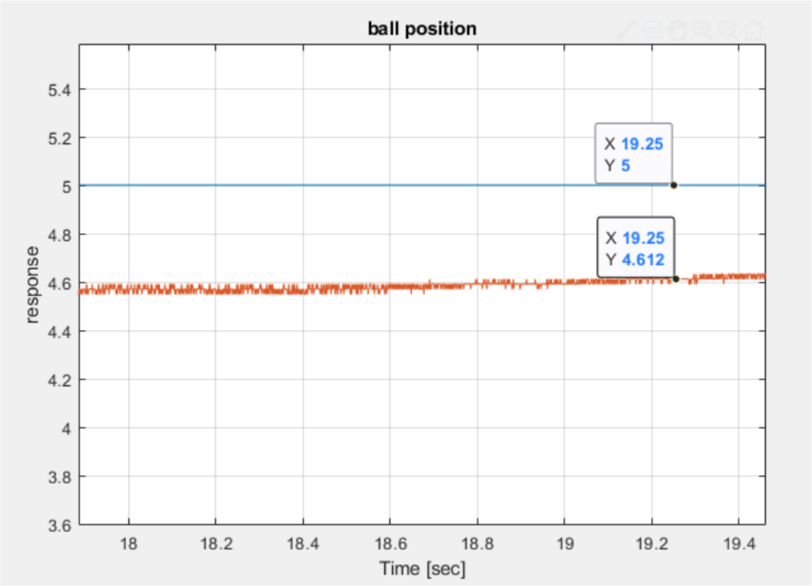
7. The plots:



8. In MATLAB:

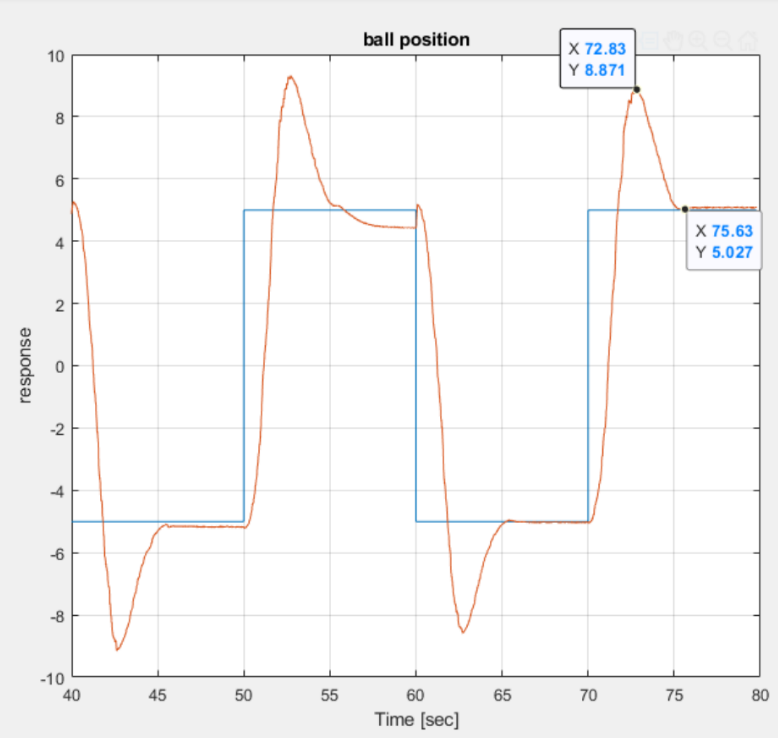


The error:



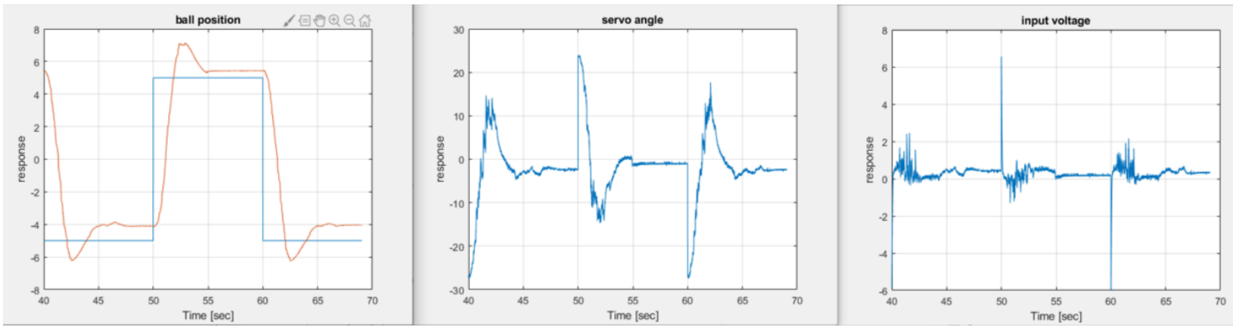
The error is: $5 - 4.612 = 0.388$

10. We add $K_i = 0.95$ and got:

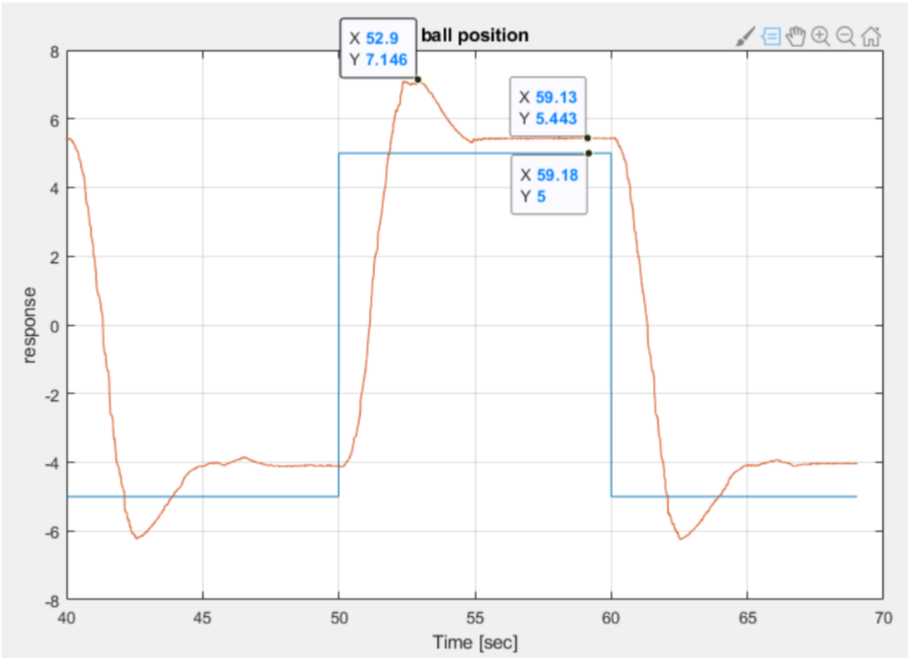


The OS%: $OS = \frac{8.871-5}{5} * 100 = 77.42\%$
The settling time = $5.63\ sec$
The error is 0.027

11. The MATALB plots for PD#2 Control:



12.



The OS% - $OS = \frac{7.146-5}{5} * 100 = 42.92\%$

The settling time = 9 sec although due to an error it doesn't converge enough

The error is $5.443 - 5 = 0.443$

4.3 Results

Section	Description	Symbol	Value	Unit
Question 11	**Pre-Lab: Ideal PD Control Design** Compensator Gain Kc Compensator Zero	K_c z	$K_c = 4.1$ $z = 1.227$	$[cm^{-1}]$ $[cs^{-1}]$
Question 12	**Pre-Lab: Practical PD Control Design** Compensator Gain Compensator Zero Compensator Pole Time Constant	K_c z T_p	$K_c = 3.5$ $z = 1.28$ $T_p = 0.23$	$[cm^{-1}]$ $[cs^{-1}]$ $[sec]$
4.1.1	**In-Lab Simulation: Cascade ideal PD with no servo dyn.** Steady-state error Settling time Percentage overshoot	e_{ss} t_s PO	$e_{ss} = 0.001$ $t_s = 4.56$ $PO = 9.56$	$[m]$ $[sec]$ $[\%]$
4.1.2	**In-Lab Simulation: Cascade ideal PD with servo dyn.** Steady-state error Settling time Percentage overshoot	e_{ss} t_s PO	$e_{ss} = 0.001$ $t_s = 6.32$ $PO = 10.07$	$[m]$ $[sec]$ $[\%]$
4.2.1	**In-Lab Simulation: Cascade Practical PD** Steady-state error Settling time Percentage overshoot	e_{ss} t_s PO	$e_{ss} = 0.00003$ $t_s = 3.819$ $PO = 6.817$	$[m]$ $[sec]$ $[\%]$
4.2.1(step 13)	**In-Lab Simulation: Cascade Tuned Practical PD #1** Compensator Gain Compensator Zero Steady-state error ess Settling time Percentage overshoot	K_c z e_{ss} t_s PO	$K_c = 3.57$ $z = 1.71$ $e_{ss} = 0.003$ $t_s = 2$ $PO = 12$	$[cm^{-1}]$ $[cs^{-1}]$ $[m]$ $[sec]$ $[\%]$
4.2.2(step 9)	**In-Lab Implementation: Tuned Practical PD #1** Steady-state error Settling time Percentage overshoot	e_{ss} t_s PO	$e_{ss} = 0.027$ $t_s = 5.63$ $PO = 38.71$	$[m]$ $[sec]$ $[\%]$
4.2.2(step10)	**In-Lab Implementation: Tuned Practical PD #2** Compensator Gain Compensator Zero Steady-state error ess Settling time Percentage overshoot	K_c z e_{ss} t_s PO	$K_c = 3.57$ $z = 1.71$ $e_{ss} = 0.443$ $t_s = 9$ $PO = 21.46$	$[cm^{-1}]$ $[cs^{-1}]$ $[m]$ $[sec]$ $[\%]$

Analysis:

We employed the same methods for calculating SS error, settling time, and overshoot:

SS error: We took a point's y-coordinate from the response graph and subtracted the step response value from it.

$$e_{ss} = |y_{res} - y_{step}|$$

PO overshoot: we measured the maximal value of the step response and divided it by the change in the step response.

$$PO = \frac{y\_(\max res) - 5}{10} \cdot 100\%$$

Settling time: the duration it takes for the response to enter and remain within 5% of the steady-state value.

5. Problems

From the plots above, it is evident that the specifications were not met. One possible reason for this could be that the values of K_i were not accurately calculated. The initial plots in the post-lab section show better convergence compared to those obtained at the end of the lab, indicating that the values of K_p and K_d were sufficiently accurate. We wanted to improve the performance of the system However, due to time constraints, we were unable to test the system with different values of K_i.

6. Conclusion

During this experiment, we learned how to implement a PD controller to balance the ball on the beam. Additionally, we observed the differences between an ideal PD controller implementation and a practical PD controller implementation.