ESE 448 Systems Engineering Laboratory

Final Project Report

Tracking of the position of a ball on a remote sensor system (SS01) ball by a ball on a Ball and Beam system (BB01).

By

Manohar Pradhan
Billy Habimana Cyusa

1. Introduction

1.1 BB01 & SS01 position control experiment using practical PD controller

Goals: The goal of our experiment was to design a practical PD controller for the Ball and Beam system (BB01) to meet the following transient performance requirements while the ball on the beam is tracking the movement of the ball on the remote sensor system (SS01):

- Percentage Overshoot (PO): $\leq 10\%$
- Steady state error (e_{ss}) 0.001m (with the SS01 ball moving at a constant speed on the trail)
- Settling time (t_s) $\leq$ 3.5s

1.2 BB01 & SS01 position control experiment using ideal PD controller

Goals: The goal of our experiment was to design an ideal PD controller for the Ball and Beam system (BB01) to meet the following transient performance requirements while the ball on the beam is tracking the movement of the ball on the remote sensor system (SS01):

- Percentage Overshoot (PO): $\leq 10\%$
- Steady state error (e_{ss}) 0.001m (with the SS01 ball moving at a constant speed on the trail)
- Settling time (t_s) $\leq$ 3.5s

2. Analysis and Procedure

2.1 BB01 & SS01 position control experiment using practical PD controller

Procedure:

The PV and PD controllers were designed and incorporated into the Simulink model as shown in Figure 1 with the parameters of the PD controller (K_c and z) and the parameters of the PV controller (the proportional and velocity gain) tuned properly to meet the required transient performance requirements. The HIL Read block reads from the Encoder channel whereas the HIL Write block writes to the Encoder channel everytime the block is executed. A calibration factor (BB calibration) was incorporated to measure the ball position from the voltage output. Another channel, ie channel 1 was linked to the HIL read block from which the output signal was sent through another calibration factor(b_{sp}) before being recorded at the scope (tracking performance) where the position of the ball on the remote sensor system was measured (SS01).

In order to investigate the transient performance, we had two different test conditions:

- SS01 ball at different static locations on the trail
- SS01 ball moves at a constant speed on the trail

To evaluate the OS% and t_s, we simulated a step input signal by putting the SS01 ball at different static positions on the trail and observing the BB01 system response. We did this a total of three times and averaged the values of percentage overshoot as well as the settling time.

To evaluate the steady-state error, we simulated a sawtooth input signal by moving the SS01 ball at constant speed on the trail and observing the response from the BB01 system and then estimating the steady-state error value. Once again, we did three trials and took an average to find the steady-state error.

Analysis: Practical PD Controller Design

The derivative operator/block in the ideal PD controller is replaced with a derivative filter with the following transfer function:

$$H(s) = \frac{\omega_f s}{s+\omega_f} \qquad \text{Eqn(1)}$$

Where w_f is the cut-off frequency and equal to 6.28 rad/s. Also, a weight parameter, b_{sp} is added which varies the amount of set point that is used to compute the error velocity. The closed loop transfer function of the Ball and Beam system is as follows:

$$\frac{X(s)}{X_d(s)} = \frac{K_{bb}K_c\left[(z+\omega_f b_{sp})s + z\omega_f\right]}{s^3+\omega_f s^2+(K_{bb}K_c\omega_f+K_{bb}K_c z)s+K_{bb}K_c z\omega_f} \qquad \text{Eqn(2)}$$

A third order characteristic polynomial can be written as:

$$s^3+\frac{(2\zeta\omega_n T_p+1)s^2}{T_p}+\frac{({\omega_n}^2 T_p+2\zeta\omega_n)s}{T_p}+\frac{{\omega_n}^2}{T_p} \qquad \text{Eqn(3)}$$

And the characteristic polynomial of the closed loop system using a practical PD controller is as follows:

$$s^3+\omega_f s^2+(K_{bb}K_c\omega_f+(K_{bb}K_c z))s+K_{bb}K_c z\omega_f \qquad \text{Eqn(4)}$$

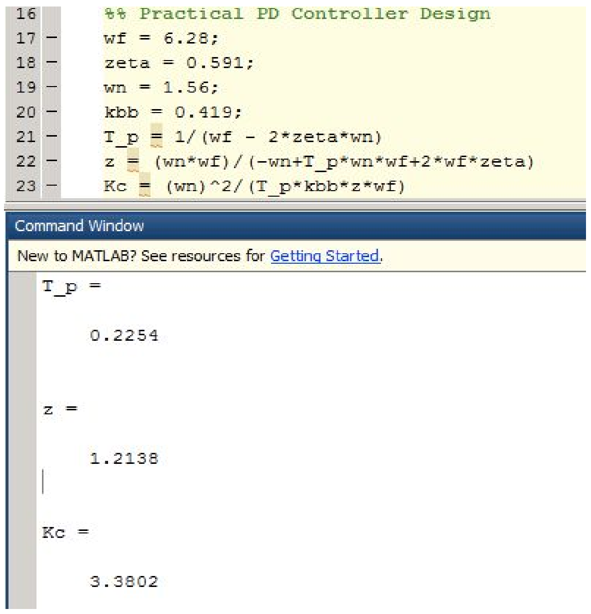
Comparing Eqn(3) and Eqn(4) we find that,

$$T_p = \frac{1}{\omega_f - 2\zeta\omega_n} \qquad \text{Eqn(5)}$$

Where T_p is the pole decay in seconds. Similarly, parameters, K_c and z can be found using the equations below:

$$K_c = \frac{{\omega_n}^2}{T_p K_{bb} z\omega_f} \text{ and } z = \frac{\omega_n\omega_f}{-\omega_n+T_p\omega_n\omega_f+2\omega_f\zeta} \qquad \text{Eqn(6)}$$

The values we obtained for T_p, K_c and z are through the following MATLAB script.



```
%% Practical PD Controller Design
wf = 6.28;
zeta = 0.591;
wn = 1.56;
kbb = 0.419;
T_p = 1/(wf - 2*zeta*wn)
z = (wn*wf)/(-wn+T_p*wn*wf+2*wf*zeta)
Kc = (wn)^2/(T_p*kbb*z*wf)
```

Command Window

New to MATLAB? See resources for Getting Started.

```
T_p =

    0.2254


z =

    1.2138


Kc =

    3.3802
```

The calculated values had to be further tuned in order to satisfy our performance requirements when doing the simulation both with and without the servo dynamics.
However, these values had to be tuned in order to meet our transient performance requirements Thus, our final value for parameters are summarized below in table 1:

Table 1: Summary of Parameter Tuning for Practical PD Controller

K_c	z
2.4	1.1

Aside from the parameters K_c and z, the practical PD controller consists of a derivative filter whose transfer function is given by:

$$H(s) = \frac{\omega_f s}{s+\omega_f s}$$

Eqn(7)

Where w_f is set to 6.28 rad/s. The purpose of using the derivative filter is to reduce the grinding noise caused by the amplified high frequency signal.

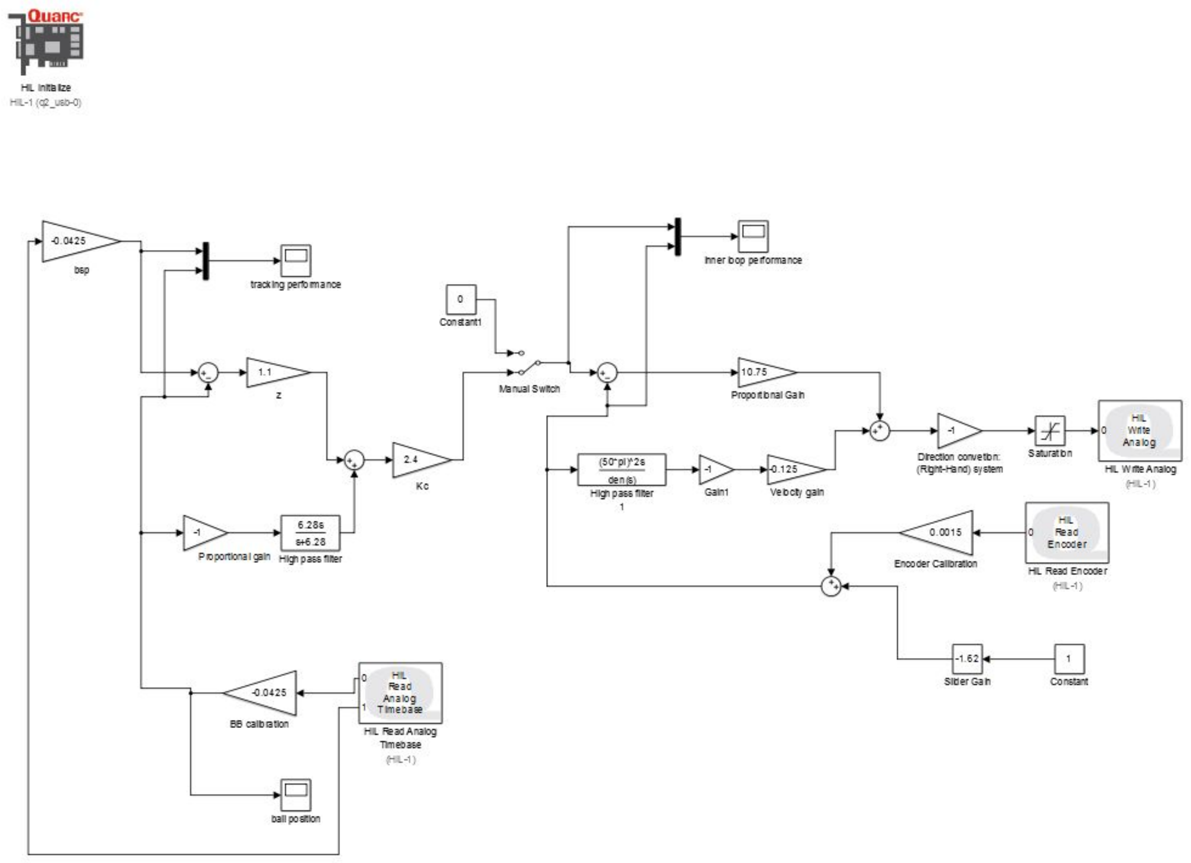


Figure 1: Simulink model of the Practical PD controller with SRV02 system

2.2 <u>BB01 & SS01 position control experiment using ideal PD controller</u>

<u>Procedure</u>

The PD and PV controllers were designed and incorporated into the Simulink model as shown in Figure 2 with the parameters of the PD controller (K_c and z) and the parameters of the PV controller (the proportional and velocity gain) tuned properly to meet the required transient performance requirements. The HIL Read block reads from the Encoder channel whereas the HIL Write block writes to the Encoder channel everytime the block is executed. A calibration factor (BB calibration) was incorporated to measure the ball position from the voltage output. Another channel, ie channel 1 was linked to the HIL read block from which the output signal was sent through another calibration factor(b_{sp}) before being recorded at the scope (tracking performance) where the position of the ball on the remote sensor system was measured (SS01).

In order to investigate the transient performance, we had two different test conditions:

- SS01 ball at different static locations on the trail
- SS01 ball moves at a constant speed on the trail

To evaluate the OS% and t_s, we simulated a step input signal by putting the SS01 ball at different static positions on the trail and observing the BB01 system response. We did this a total of three times and averaged the values of percentage overshoot as well as the settling time.

To evaluate the steady-state error, we simulated a sawtooth input signal by moving the SS01 ball at constant speed on the trail and observing the response from the BB01 system and then estimating the steady-state error value. Once again, we did three trials and took an average to find the steady-state error.

Analysis: Ideal PD Controller Design

The traditional PD controller has the following form:

$$G_{PD}(s) = K_c(s + z)$$

Eqn(8)

Where K_c and z are the gain and open loop pole respectively. Due to noise in the signals, the traditional PD controller is slightly modified where a differentiator block is added to get rid of the noise signals. The closed loop transfer function of the Ball & Beam system is as follows:

$$\frac{X(s)}{X_d(s)} = \frac{K_{bb}K_c z}{s^2 + K_{bb}K_c s + K_{bb}K_c z}$$

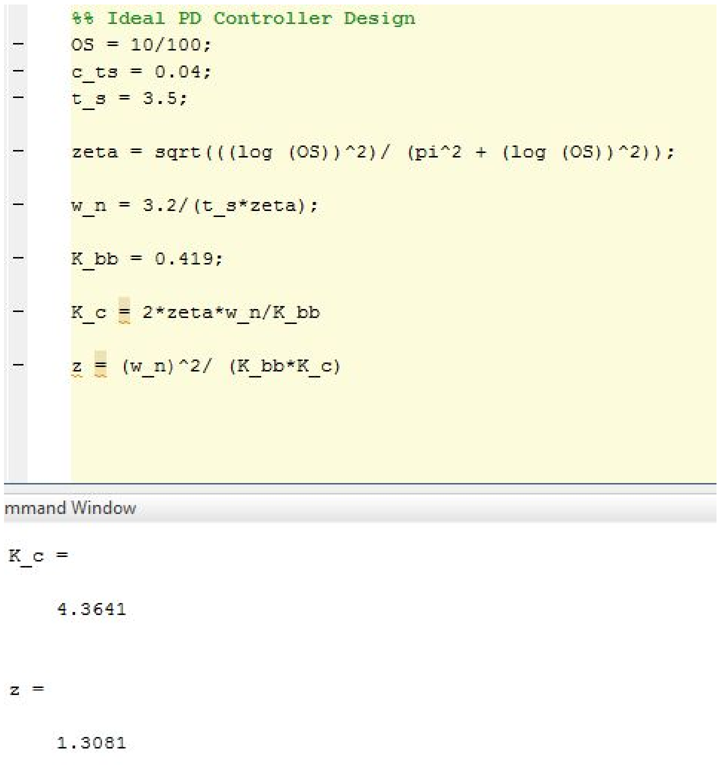
Eqn(9)

Comparing this with our second order underdamped canonical system, we get

$$K_c = \frac{2\zeta\omega_n}{K_{bb}}, z = \frac{{\omega_n}^2}{K_{bb}K_c}$$

Eqn(10)

The following MATLAB script shows our calculations on the controller parameters in order to satisfy the transient performance requirements:



```
%% Ideal PD Controller Design
OS = 10/100;
c_ts = 0.04;
t_s = 3.5;

zeta = sqrt(((log (OS))^2)/ (pi^2 + (log (OS))^2));

w_n = 3.2/(t_s*zeta);

K_bb = 0.419;

K_c = 2*zeta*w_n/K_bb

z = (w_n)^2/ (K_bb*K_c)
```

mmand Window

```
K_c =

    4.3641


z =

    1.3081
```

However, these calculated values had to be further tuned in order to meet our transient performance requirements. Thus, our final value for parameters are summarized below in Table 2:

Table 2: Summary of Parameter Tuning for Ideal PD Controller

K_c	z
2.5	1.1

The ideal PD controller does not incorporate a derivative filter and has a derivative block instead of the derivative filter in its design.

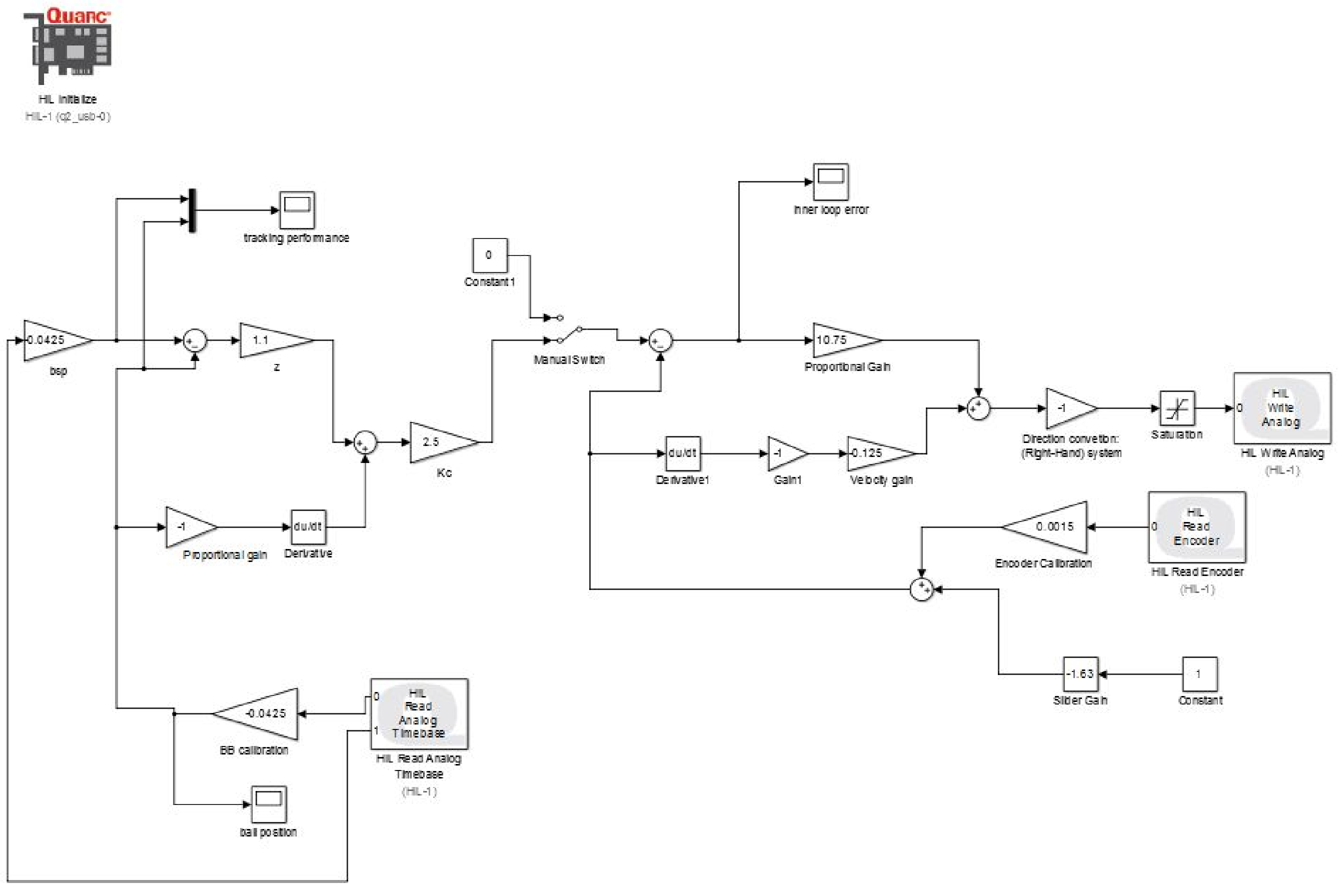


Figure 2: Simulink model of the ideal PD controller with servo dynamics

3. Results

3.1 BB01 & SS01 position control experiment using practical PD controller

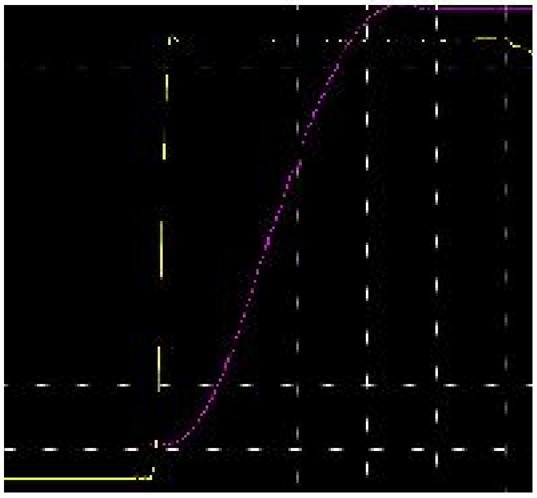


Figure 3: Step response using the practical PD controller

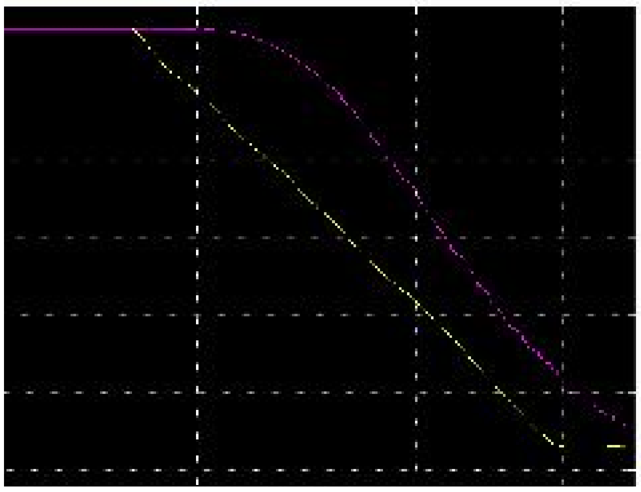


Figure 4: Ramp response using the practical PD controller

3.2 BB01 & SS01 position control experiment using ideal PD controller

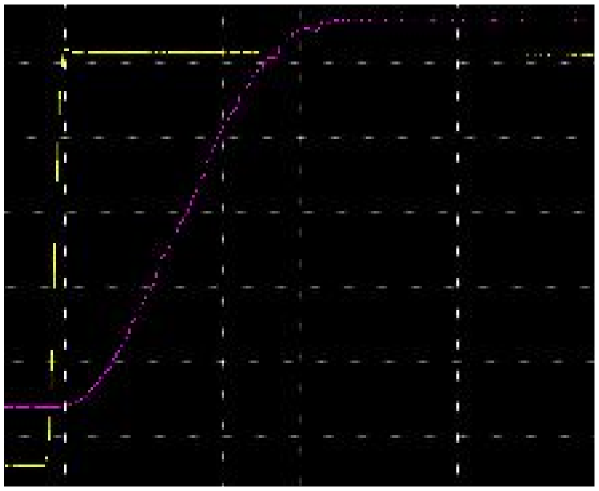


Figure 5: Step response using the ideal PD controller

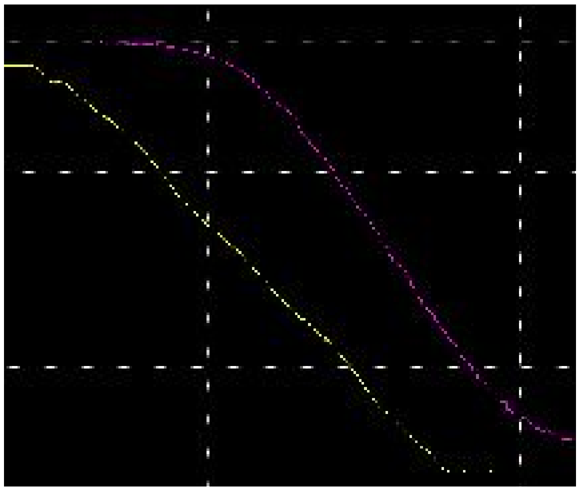


Figure 6: Ramp response using the ideal PD controller

Table 3: Transient performance of the system

Controllers	OS% (average of 3 trials)	t_s (average of 3 trials)
Ideal PD controller	11% (±0.5%)	3.3s (±0.05s)
Practical PD controller	10% (±0.5%)	3.44s(±0.1s)

Note: The steady-state error was constant and *ess* ≤ 0.01m for all 6 trials.

4. Conclusions

After performing the experiment and evaluating the transient performance using both the practical and ideal PD controllers, we conclude that the overall performance given by the practical controller satisfied our transient and steady state error requirements. We were able to attain both the required overshoot percentage as well as meet the settling time requirements and also able to narrow the constant steady state error to less than 0.01m in all of our trials. However, the performance from the ideal PD controller was not far off and there was a slight trade off in trying to meet the settling time and overshoot requirements concurrently. In other words, the incorporation of the derivative filter might have contributed greatly to improve the transient performance in the practical PD controller design.

4.1 Comments on performance requirements of the system

Based on the results summarized in Table 3, using the practical PD controller the system satisfies the performance requirements. As can be seen in Table 3, we were able to tune the system so that it had a percentage overshoot of 10% and a settling time of 3.44s.

For the ideal PD controller, while tuning we were unable to successfully find values of K_c and z so that the system would meet all performance requirements. Therefore, we decided to tradeoff OS% for t_s. The system settled within a satisfactory amount of time: 3.3s but the OS% was 10.56%.

Both the ideal and practical PD controllers satisfied the steady-state error requirement. The steady-state error was ≤ 0.01m.

The reason why the ideal PD controller could not quite meet all the performance requirements is because it was only using a derivative block. So the system was not filtering for noise to the best of its ability. Once, we replaced this block by a derivative (high-pass) filter, the system met the performance requirement.

4.2 Problems encountered and how we solved them

As the experiment involving the movement of the ball on the SS01 had to be performed at a constant speed, moving the ball at a constant speed manually was inaccurate and might have affected our performances. Repeated trials were performed until a constant looking sloped input was produced to solve this issue (though not exact).
There was also a significant challenge moving the ball manually the same distance to generate identical inputs when performing different trials of the experiment. This was solved by assigning markers and moving the ball from one end of the marker to the other.

4.3 Do the filters for ball position sensors improve the performance of the system? Discuss this based on experimental data

We used the derivative filter instead of the derivative block to improve performance when experiment was performed with the practical PD controller. This enabled us to achieve our transient and steady state error requirements. The filter was able to reduce the grinding noise caused by the amplified high frequency signal on the analog sensor.