UNIVERSITY OF WATERLOO

University of Waterloo
Faculty of Mechatronics and Mechanical Engineering

ECE 484
Digital Control Applications

Lab 3: Major Loop Controller Design

Professor John W. Simpson-Porco

Group 89
Station 17
Alistair Fink: 20601925
Michael HoSue: 20579851

November 21, 2019

Statement of Originality

The experiments conducted in this lab report were conducted on station 17.

The authors of this report acknowledge that (a) they have jointly authored this submission, (b) this work represents their original work, (c) that they have not been provided with nor examined another person's ball & beam project report, either electronically or in hard copy, and (d) that this work has not been previously submitted for academic credit.

Alistair Fink

Michael HoSue

1.0 Introduction

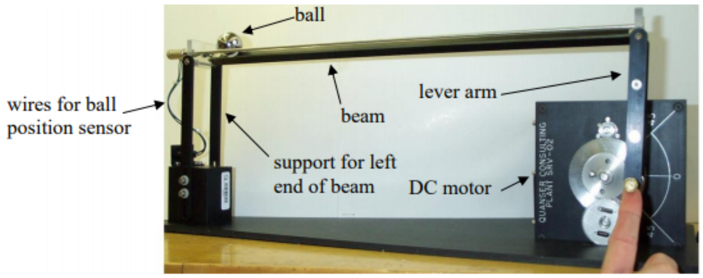


Figure 1: Experimental Apparatus [1]

As shown in figure 1, the experimental apparatus is divided into two sections, the beam, and the gearbox connected to the DC motor. The gearbox is connected to the beam through a lever arm and allows rotational movement from the motor to be translated into tilting movement of the beam. This movement can then be leveraged to balance a ball on the beam at a particular destination on the beam.

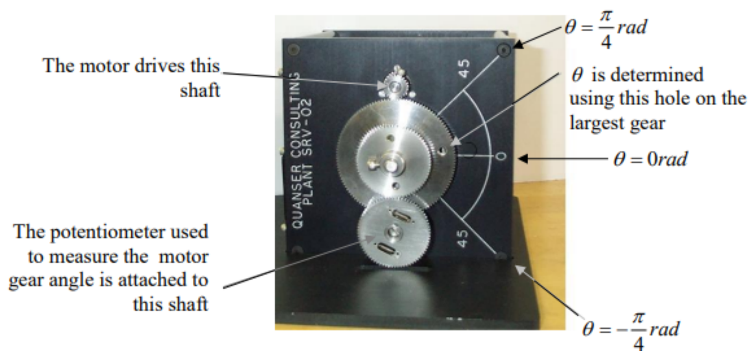


Figure 2: Gearbox [1]

The gearbox is powered by a single DC motor as seen in figure 2. This motor turns the upper most shaft in figure 2 and allows the speed of the entire system to be controlled depending on the voltage applied to the shaft. This torque is transferred to the middle most shaft, as seen in figure 2, which also contains a threaded hole that the lever arm can be attached to as seen in figure 1. This torque is once again transferred to the bottom most shaft which is connected to a potentiometer. This potentiometer can then be utilised to read the angle of the middle gear.

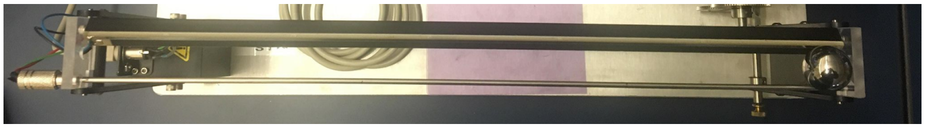


Figure 3: Ball and Beam

As seen in figure 3 the ball and beam system is a much simpler system than that of the gearbox. The beam simply consists of two rails with a metal ball allowing the circuit between the two rails to complete. Since the ball shorts the circuit, the resistance of the system varies depending on the position of the ball. Essentially, this acts as a potentiometer where the voltage drop of the two rails can be measured to determine where the ball is positioned on the beam.

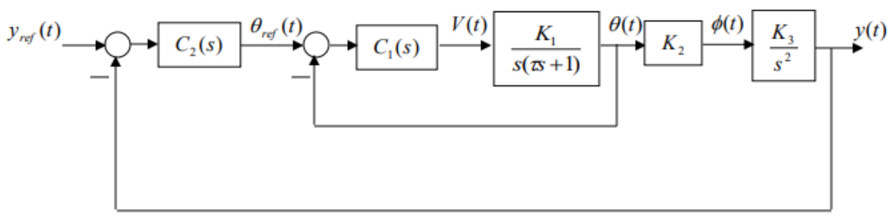


Figure 4: Experimental Apparatus Control System Block Diagram [1]

As previously discussed there are two measurements measured by the system. The first is the potentiometer reading of the gearbox and the second is the potentiometer reading of the beam itself. In order to utilise these we can design our control system using major and minor loop design by first designing for the gearbox only then designing for the beam and ball. If the closed loop bandwidth of the outer loop is five to ten times lower than that of the inner loop then we can safely say that the inner loop is seen as a gain of one to the outer loop. [2] Using this approach the two sections can be separately designed for which greatly reduces the complexity of the design process.

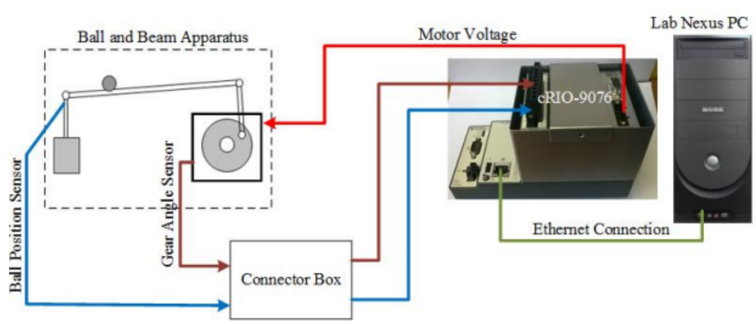


Figure 5: Experimental Apparatus and Computer Connections [1]

Once the system is designed for, the controllers can be discretized and utilising a control computer connected to a pc as seen in figure 5 the system can be digitally controlled from the computer.

2.0 System Modelling

2.1 Gearbox Modeling

In order to design a controller for the inner loop there are three components that are required. These components include the angle scaling of the gears, the gear static friction offset, and the frequency domain model of the system. These values were experimentally determined in lab 1.

The first component, the angle scaling, allows us to linearize the voltage measurement of the gear's potentiometer allowing us to formulate a single conversion equation to find the gear's angle in radians. This resulted in an offset of 6.287 V and a gain of -0.7334 V/rad. As a result we find that the equation to convert the measured voltage to radians is as follows in equation 1. These values remained the same throughout this lab and lab 2.

$$\theta_{Gear} = \frac{V_{Gear} - 6.287}{-0.7334} \tag{1}$$

The second component, the gear stiction, accounts for the nonlinearity introduced by static friction. These values were experimentally determined by slowly increasing the applied voltage on the motor until movement occurred in each direction. This resulted in a clockwise stiction value of 0.48 V and a counterclockwise stiction value of -0.48 V. These values were eventually modified to 0.25 V and -1 V in lab 2 then to 0.15 V and -0.15 V in this lab as these new values seemed to make the performance of the system better with the beam attached to the experimental apparatus.

The last component, the system model, was experimentally determined by commanding a particular angle and analyzing the system response. The experimentation resulted in a K_1 of 1.88 and a τ of 0.068. Putting these together resulted in a gearbox plant model as shown in equation 2. These values remained the same throughout this lab and lab 2.

$$P_{Gear}(s) = \frac{1.88}{0.068s^2 + s} \tag{2}$$

2.2 Beam Modeling

For the beam and ball there are four components that are required in order to adequately design the outer loop. These components include the ball position scaling, K_2, K_3, and the inner loop controller. The values were experimentally determined or formulated in lab 2.

The first component, the ball position scaling, was experimentally determined by measuring the beam's potentiometer voltage when the ball was at various positions on the beam. These values were linearized using linear regression to find a beam offset of -30.56 cm and a gain of 9.375 cm/V. These values were then used to formulate a singular equation to convert between the measured beam voltage to the ball position in meters as shown in equation 3. These values were not changed for this lab.

$$BallPosn\ [m] = \frac{9.357 \cdot V_{Beam} - 30.56}{100} \tag{3}$$

The second component, K_2, was given by the lab manual and was geometrically confirmed as the ratio between the gear's angle and the beam's angle. This value is 0.061 and can be used to convert the gear angle to the beam's angle. This value was not changed for this lab.

The third component, K_3, was also given by the lab manual and was experimentally confirmed through an experimental vs simulated analysis. Through this we found that the simulation closely matched what we saw in the experiment which confirmed that the value of 4.78 closely matched the physical properties of the beam. This value was not changed for this lab.

The fourth component, the inner loop controller, was formulated using the data outlined in section 2.1. Using pole placement and the bilinear discretization we formulated the controller as shown in equation 4 with closed loop transfer function poles at $s = -8 \pm 5j,\ -60 \pm 5j,\ -90$. This controller was not changed for this lab.

$$C_1[z] = \frac{2.619z^2 + 1.281z - 1.255}{z^2 - 0.1333z + 0.03625} \tag{4}$$

3.0 Lead Controller Simulation

As instructed by the lab manual, the lead controller, as shown in equation 5, was implemented in Simulink so that the entire system looked like figure 6.

$$C_{2LD}(s) = 7\frac{s+0.35}{s+2.5} \tag{5}$$

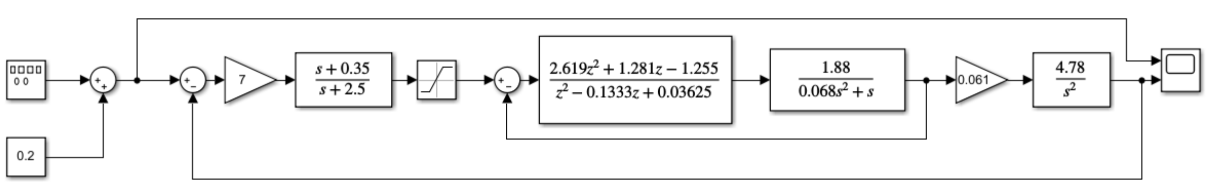


Figure 6: Lead Outer Loop Controller Block Diagram

A step input from 0.15 m to 0.25 m was then applied using a square wave signal of magnitude 0.05 m with a DC offset of 0.2 m. The obtained response is shown in figure 7.

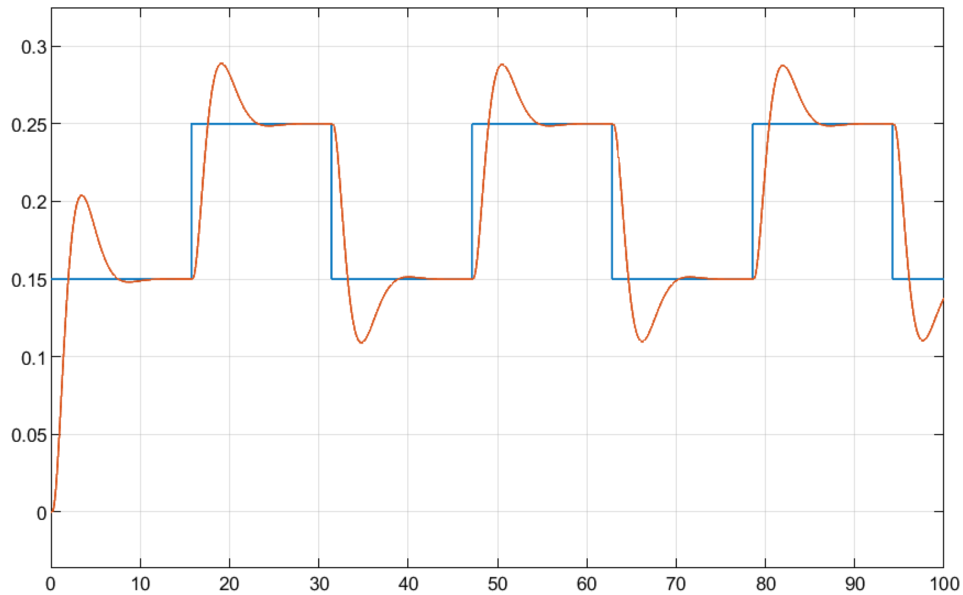


Figure 7: Lead Outer Loop Controller Continuous Simulation Step Response

As seen in figure 7 above, the step response had an overshoot of approximately 0.03 m, a settling time of approximately 9 s and virtually 0 steady state error as expected due to the presence of the double integrator in the plant.

4.0 Lead Controller Implementation

Next, the lead controller was discretized by using the bilinear transform and the step response of 0.15 m to 0.25 m (like part 3) was obtained from Simulink as shown below in equation 6 and figure 8.

$$C_{2LD}(s) = 7\frac{s+0.35}{s+2.5} \rightarrow C_{2LD}[z] = \frac{5.987z - 5.657}{z - 0.6634} \quad (6)$$

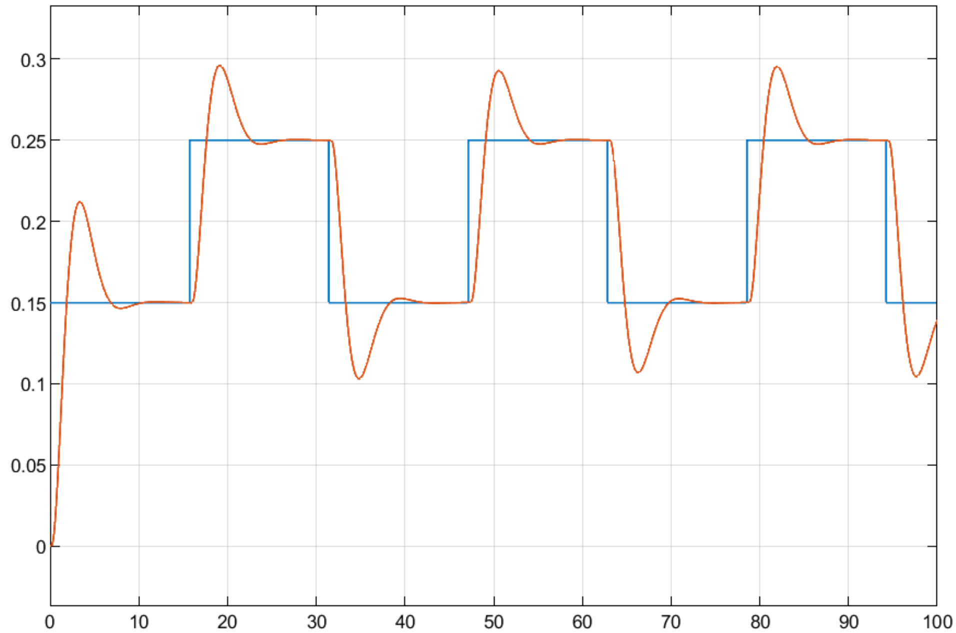


Figure 8: Lead Outer Loop Controller Discrete Simulation Step Response

It can be observed that the discretized controller had almost the same step response as the simulated continuous time model. The only noticeable difference is that the discrete controller had slightly higher overshoot of 0.045 m.

The discretized outer controller was then changed into a difference equation (equation 7) and implemented in the labview code.

$$C_{2LD}[z] = \frac{5.987z - 5.657}{z - 0.6634} \rightarrow \theta ref = 5.987\, Err - 5.657\, Err_1 + 0.6634\, \theta ref_1 \quad (7)$$

With the gear angle saturator in place, the new outer loop controller was tested with a square wave input from 0.15m to 0.25m. The response is shown below in figure 9.

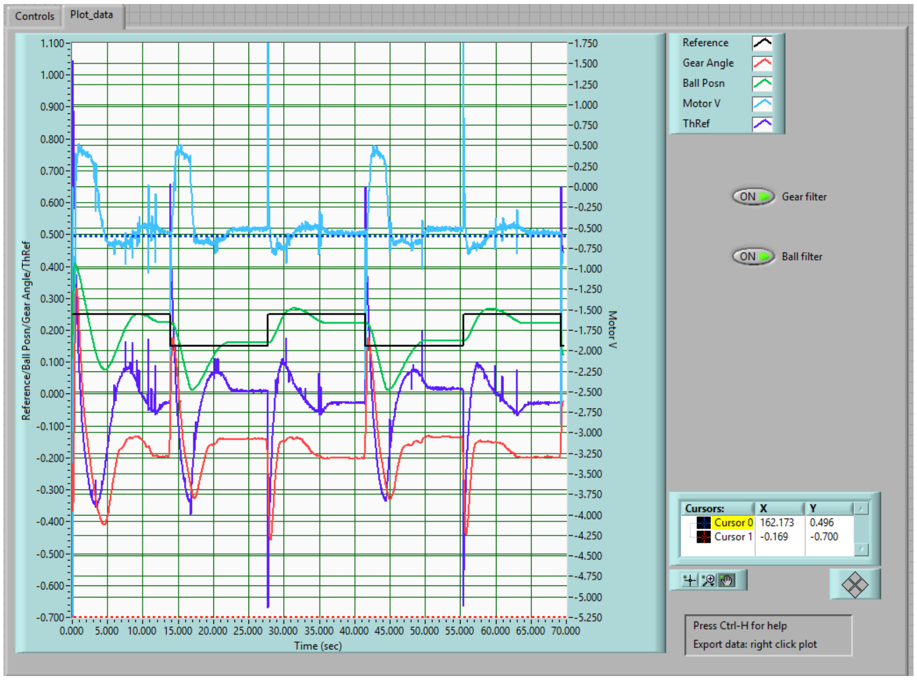


Figure 9: Lead Outer Loop Controller Experimental Step Response

It was found that there was neither a symmetrical response, nor zero steady-state error, despite the presence of the double integrator in the plant. The rising overshoot was only 0.02 m while the falling overshoot was 0.15 m (the ball rolled back to the end of the beam). When tracking the 0.25 m signal there was also consistently slightly higher steady state error than when tracking the 0.15 m input. This may be attributed to random non-linearities such as the presence of ball stiction (discussed and corrected later in the report) preventing the controller from correcting small errors in the position of the ball.

5.0 Outer Loop Controller Design

As previously discussed the lead controller laid out in the lab manual seems to suffer from poor tracking with a lot of steady state error. In order to overcome this, a new controller is required accurately track the ball's position. To do this several constraints for a square wave with peak to peak amplitude of 0.03 m are laid out in the lab manual as follows:

- Zero Step Response Steady-State Error
- Step Response 2% Settling Time of No More than 7 Seconds
- Step Response Overshoot of 45% or Less
- Command Angle Does Not Saturate
- Integrator Included in Controller

Before designing the controller we know from the course notes that if the outer loop closed loop bandwidth is five to ten times lower than that of the inner loop's closed loop bandwidth then the outer loop will see the inner loop simply as a gain of one. [2] From lab 2 we know that our inner loop's closed loop bandwidth is 8.6627 rad/s. This means that if we can attain an outer loop closed loop bandwidth of 1.73254 rad/s or lower the inner loop can safely be ignored.

Since an integrator is required in the controller for proper tracking we know that our controller should take the form as shown in equation 8.

$$C_2(s) = \frac{1}{s} C_{2-Interim} \tag{8}$$

In order to account for this we instead design $C_{2\text{-Interim}}$ for an augmented plant as shown in equation 9.

$$P_{aug} = \frac{1}{s} P(s) \tag{9}$$

In order to design according to these specifications we used an emulation approach coupled with pole placement as shown in appendix B. We found that with a closed loop transfer function with poles at $s = -0.01 \pm 0.01j,\ -1.7 \pm 0.9j,\ -7.5$ results in an overshoot of about 2.3156% and a settling time of about 6.4 seconds. These poles result in a controller as shown in equation 10.

$$C_{2-Interim}(s) = \frac{97.18s^2+1.923s+0.01903}{s^2+10.92s+29.42} \tag{10}$$

To discretize this controller we used the bilinear approximation. For the sampling period, T, we would usually calculated it based on the closed loop bandwidth of the system as shown in equation 11. Since the system in LabView only seems to sample at one sampling period for both controllers we opted to use the sampling period of the inner loop controller. This is because of the fact that the inner loop controller is required to sample at higher speeds in order to have the higher bandwidth. In this case using a slower sampling period, derived from the outer loop, may negatively impact the performance of the inner loop controller. As discussed in lab 2 the inner loop's sampling period is 0.029012 seconds.

$$\frac{2\pi}{T} \geq 25\omega_{bw} \tag{11}$$

Substituting the bilinear transformation for C_2 results in a discretized controller as shown in equation 12.

$$C_2[z] = \frac{1.211z^3-1.21z^2-1.211z+1.21}{z^3-2.707z^2+2.435z-0.728} \tag{12}$$

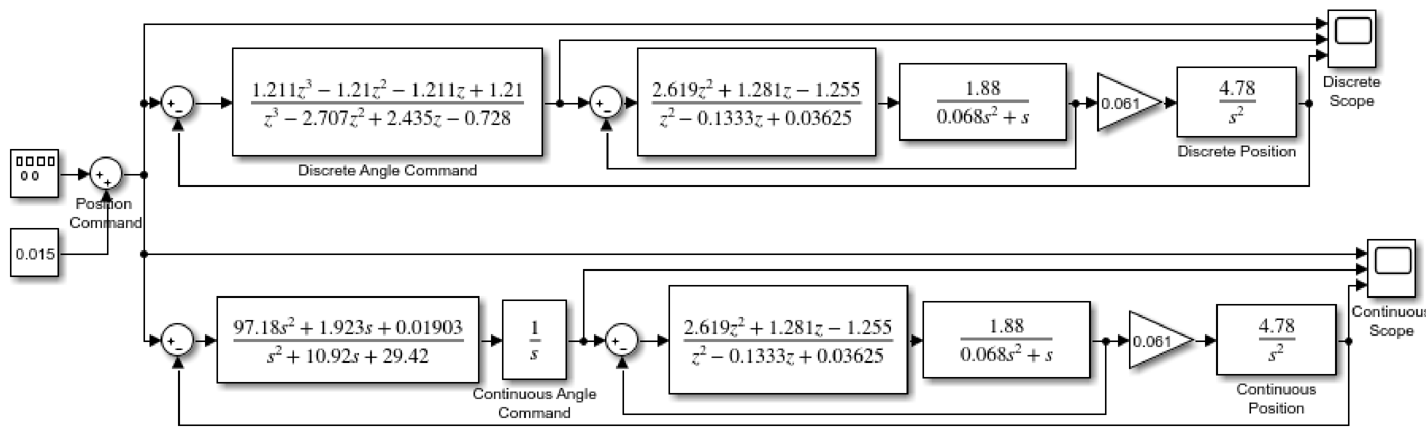


Figure 10: Outer Loop Controller Design Continuous and Discrete Simulation

As shown in figure 10 we can form a simulink simulation to confirm this controller. As shown, we can split the input into a discretized controller system and a continuous controller system allowing us to analyze both system's performance simultaneously. One aspect of the simulation to note is that while we multiply the continuous controller by the integrator we don't do this in the discretized controller as it is already accounted for when discretizing, as shown in appendix B.

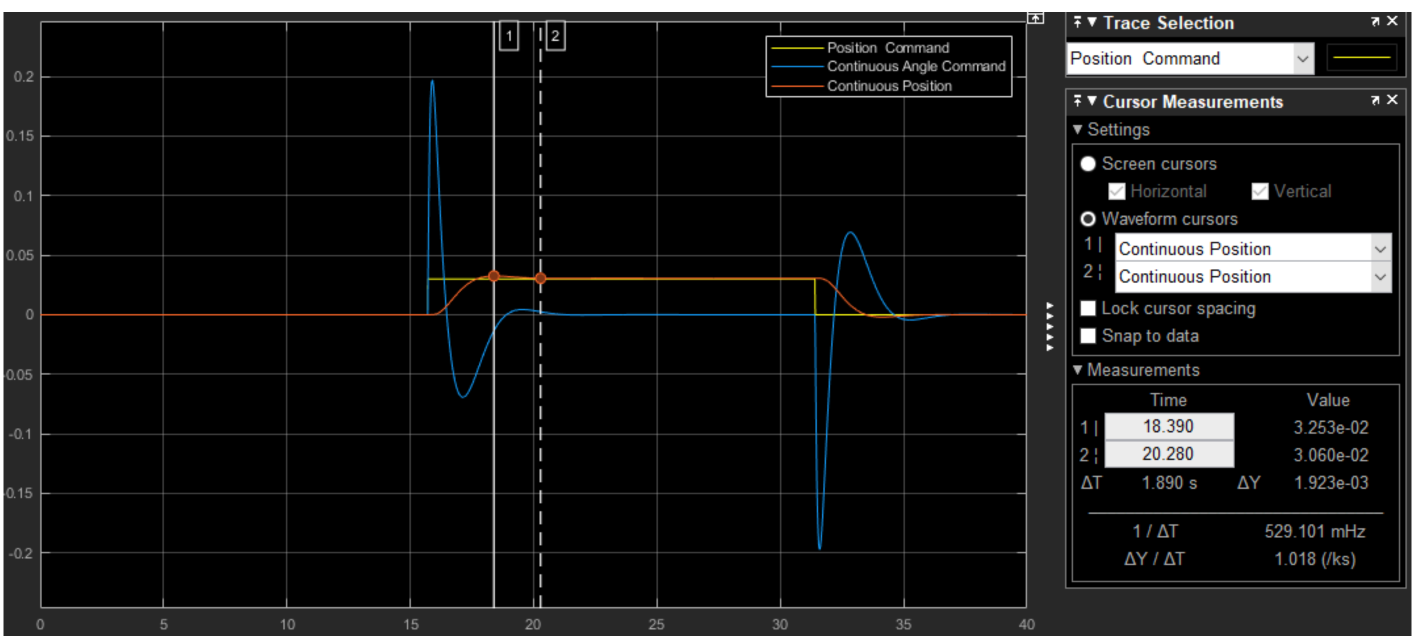


Figure 11: Step Response of Continuous Controller

As can be seen from figure 11 we find that the continuous controller appears to have an overshoot of about 0.03253 m which translates to about 8.4% well below our given specification of 45%. We also find that the settling time where the position stays within

2% of the command signal seems to occur at 20.28 seconds. This translates to about 4.572 seconds which is also well below our given specification of 7 seconds. In addition to this we find that the command signal for the angle is well within our saturation of ±0.7 rad. The controller also seems to settle to within very little steady state error which means that this controller satisfies all given specifications.

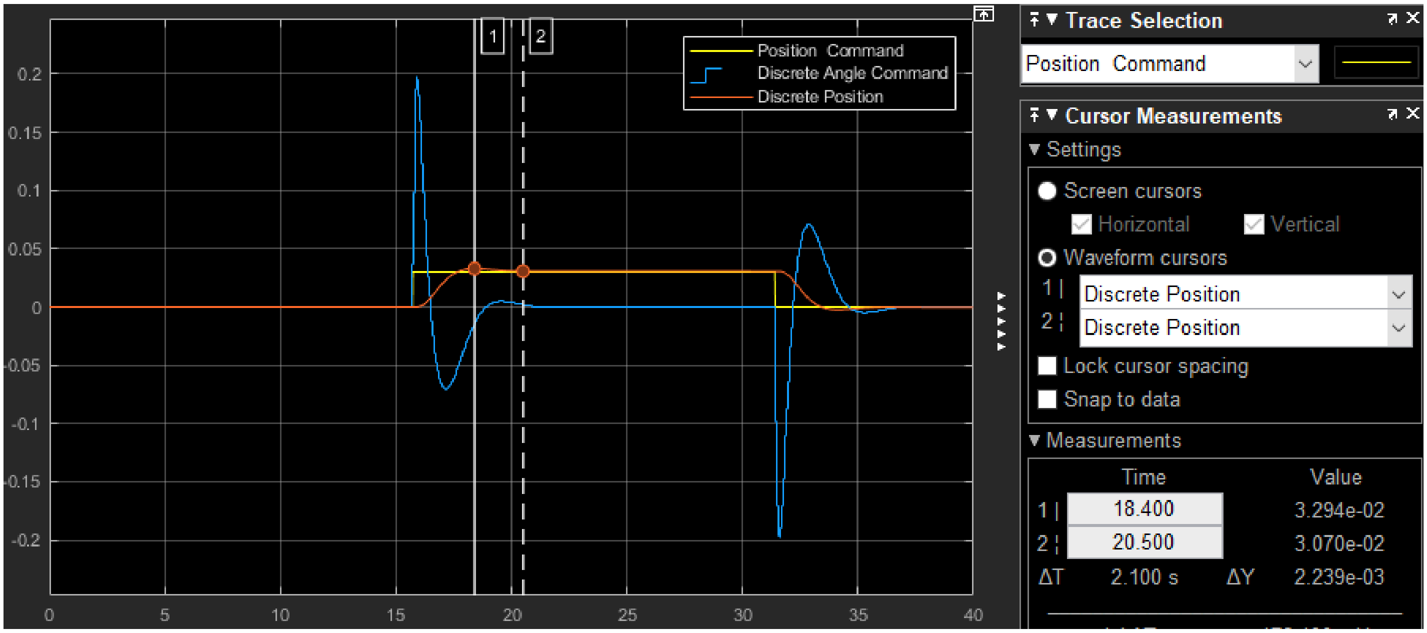


Figure 12: Step Response of Discrete Controller

As shown in figure 12, simulating the discrete controller returns similar results to that of the continuous controller with a settling time of 4.792 seconds and an overshoot of about 8.8% which are both within specification. In addition to this we also find that the discrete command signal to be well within our saturation limits. This means that this discrete controller satisfies all given specifications.

6.0 Outer Loop Controller Simulation and Implementation

To investigate the performance of the outer loop controller designed in the previous section, the response to a larger square wave from 0.1 m to 0.25 m was simulated. This was done using the same system shown in figure 10 with added saturator blocks as shown in figure 13.

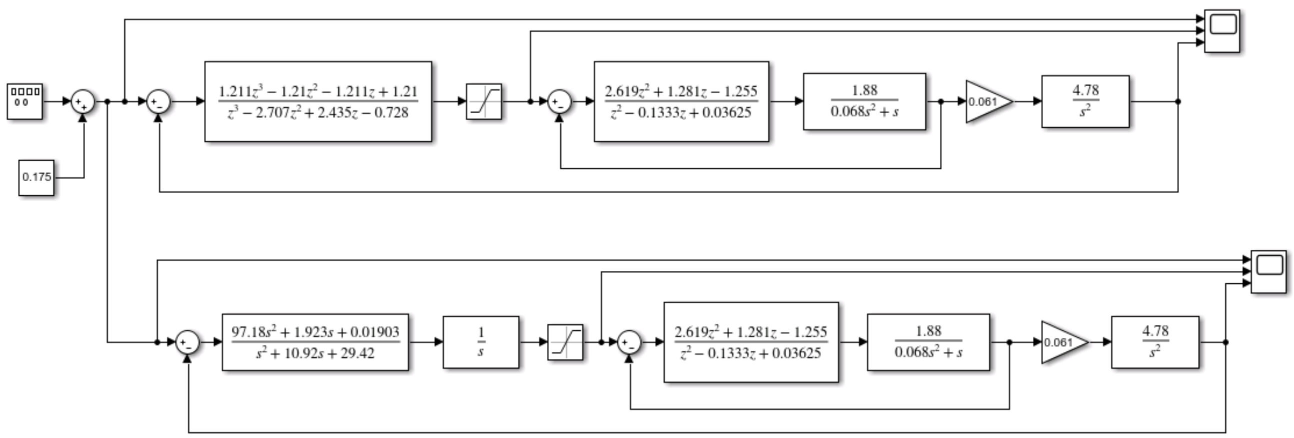


Figure 13: Outer Loop Controller Design Continuous and Discrete Simulation

The obtained step response from the discrete time controller is shown below in figure 14.

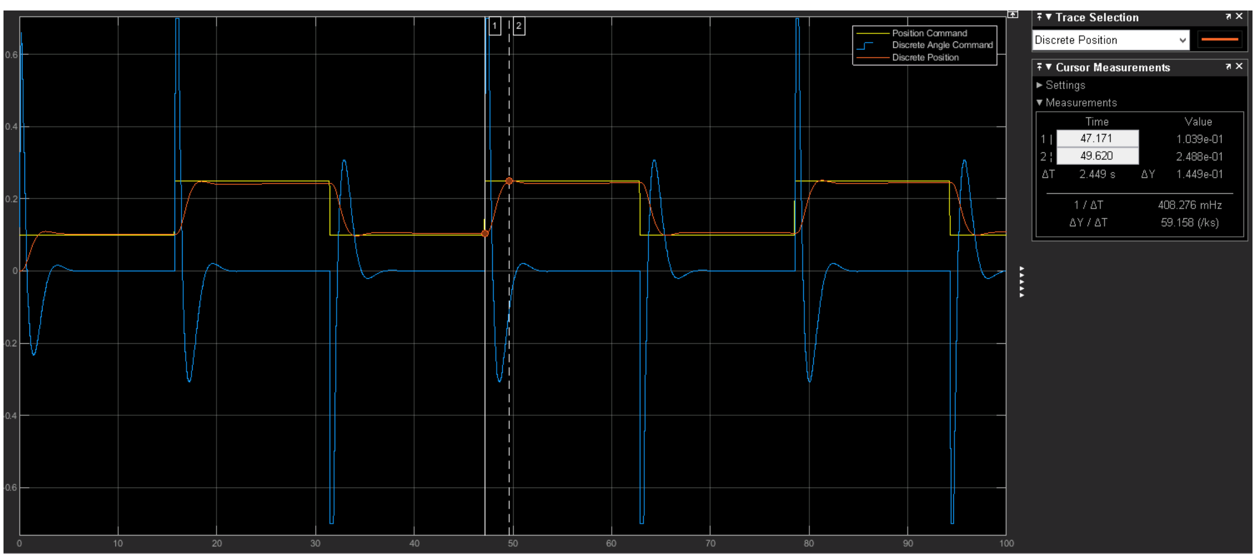


Figure 14: Simulated Step Response of Discrete Controller to 0.1 m to 0.25 m Sqwv

The simulation shows a step response with almost 0 steady-state error, 0.25 s settling and 0 overshoot which are all within the specifications outlined in section 5. However, the step response to the larger square wave does cause the reference angle, θ_{ref}, to saturate. This is because of the larger immediate error present on rising and falling edges of the input. Regardless, this should not be a problem for the apparatus due to the presence of the gear angle saturator.

Next the discrete controller was changed to a difference equation as shown in equation 13.

$$\theta ref = 1.211\,Err - 1.21\,Err_1 - 1.211\,Err_2 + 1.21\,Err_3 + 2.707\,\theta ref_1 - 2.435\,\theta ref_2 + 0.728\,\theta ref_3 \tag{13}$$

The difference equation was then implemented in the lab view code and the response of the new outer loop controller was tested using a square wave input from 0.1 m to 0.25 m. The obtained step response is shown below in figure 15.

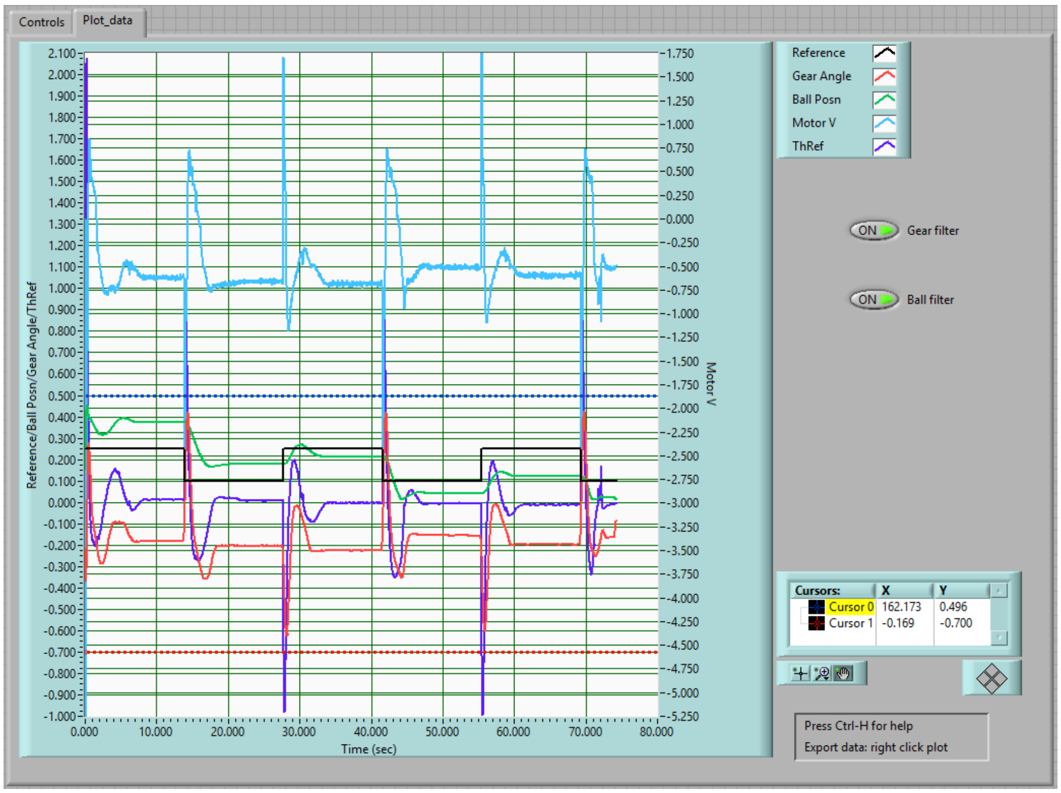


Figure 15: Experimental Step Response of Discrete Controller to 0.1 m to 0.25 m Sqwv

It was found that none of the specifications outlined in section 5 were met by the experimental steady state response except for overshoot (which was inconsistent). In general the experimental response tracking was very poor with large and inconsistent steady state error as well as settling times. As expected based on the simulation, the reference angle, θ_{ref}, was also saturating on rising and falling edges of the square wave input.

The most challenging aspect of designing this controller for the smaller square wave input in section 5 was ensuring θ_{ref} did not saturate while still maintaining a settling time less than 7 s. Thus it was expected before carrying out the experiment that the reference angle would exceed 0.7 rad. However the poor steady state tracking was not expected but is likely due to ball stiction which is corrected in the next section.

7.0 Outer Loop Stiction

It was found that the controller would not be able to correct for very small errors in the ball position. This was because very small reference angles would not be enough to cause the ball to roll on the beam, thus introducing non-linearity into the control system.

In order to correct this, the minimum gear angles required for the ball to roll were determined. This was done experimentally by using only the inner loop controller to set a small gear angle. The ball would then be released to roll freely on the beam. This was repeated using increments of the gear angle until the ball started to roll when released.

The minimum angles required to make the ball roll both left and right on the beam were determined to be ±0.4 rad and implemented in the labview code. The acceptable steady state error (as outlined in the lab manual) was 4% = 0.006 m. In order to prevent small fluctuations in the ball position during steady state, stiction offsets were wrapped in an if statement so they would only be added to the reference angle if the error in position was greater than the allowable error. See the labview code in appendix A for implementation.

8.0 Controller Redesign

Due to the linearization of the ball position on the beam which was linearized in section 7.0 a new controller is required. New specifications for a square wave input with peak to peak amplitude of 0.15 m are given by the lab manual below.

- Step Response Steady-State Error Less than 4%
- Step Response 2% Settling Time No More than 9 Seconds
- Step Response Overshoot Less than 45%
- Commanded Angle can Saturate as Step Size is Large

Since we are redesigning the previous controller the same integrator approach as before is used. This means that the controller will be designed in the same fashion as equation 5-0.5 using an augmented plant as in equation 5-0.6.

An emulation approach coupled with pole placement is also used again using the same script section 5.0 used in appendix B. Through pole placement we find that placing the poles of our closed loop transfer function at $s = -0.7 \pm 0.4j,\ -8 \pm 0.5j,\ and\ -10$ yields a step response with settling time of 4.6285 seconds, under our specification of 9 seconds, and a max overshoot of 27.9175%, also under our specification of 45%. This results in the controller as shown in equation 14.

$$C_2(s) = C_{2-Interim}(s) \cdot \frac{1}{s} = \frac{3338s^2+3585s+1432}{s^2+27.4s+261.3} \cdot \frac{1}{s} \tag{14}$$

In order to discretize this controller we can use the c2d function with the bilinear transform. As we did in section 5.0 we again will use the inner loop's sampling time of 0.029012 seconds since LabView only samples as a single sampling time. Doing this discretization for our new $C_2(s)$ results in the controller as shown in equation 15.

$$C_2[z] = \frac{33.86z^3-32.81z^2-33.85z+32.82}{z^3-2.301z^2+1.754z-0.4527} \tag{15}$$

Running the bandwidth function in matlab for our closed loop shows us that the closed loop bandwidth of this system is about 6.957118. This is much higher than the factor of five of the inner loop bandwidth that indicates the zone where the inner loop can be safely ignored. This means that this controller may not perform as expected on the experimental apparatus so a simulation is required to fully test this controller.

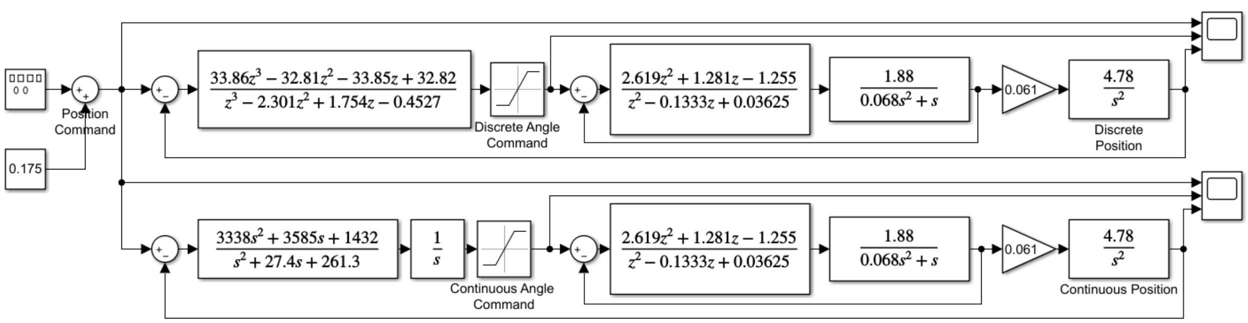


Figure 16: Simulink Simulation for Discrete and Continuous Versions of Controller

In order to check whether or not this controller will satisfy the specifications outlined a simulation is required. As shown in figure 16 a simulink simulation was constructed to simulate the square wave specified in the specifications. This square wave then inputs into a simulation for the discrete version of this controller and a simulation for the continuous version of this controller.

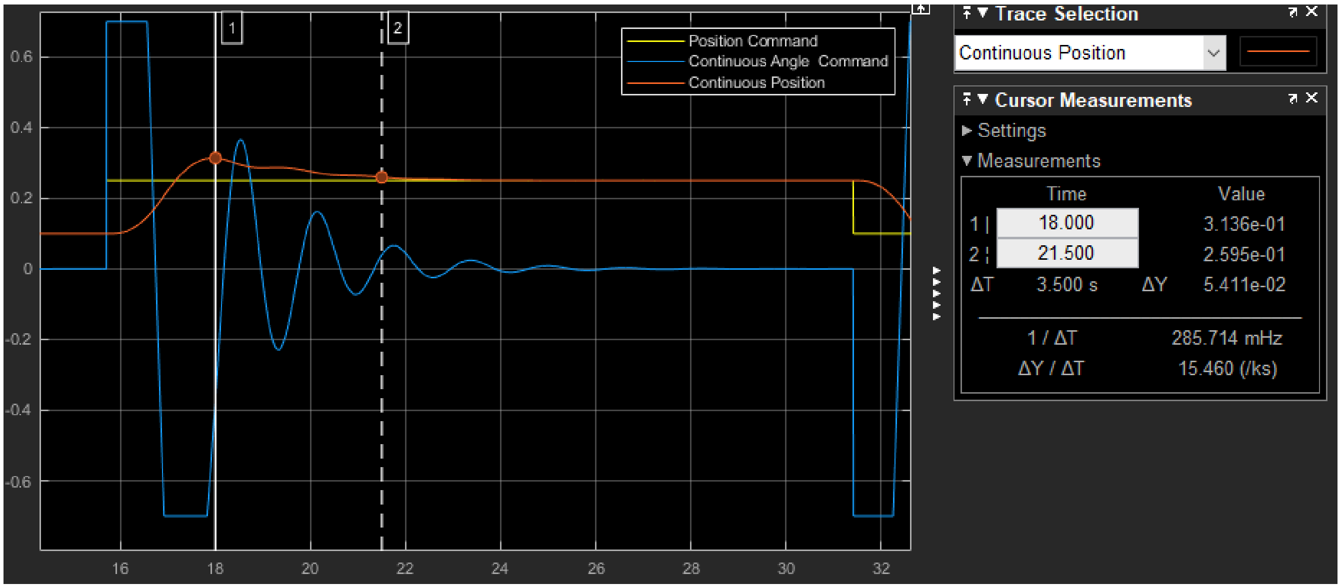


Figure 17: Continuous Controller Redesign Simulation Results

Running the simulation for the continuous controller results in an overshoot of about 42.4% with a settling time of about 5.792 seconds which are both within our given specifications. As we can see from figure 17 we see that the command signal for the angle seems to saturate at the 0.7 rad and -0.7 rad but as we know from the specifications this is ok since the step size is large. We also find that the steady state error of the simulation is well within the 4% specification which means that the continuous version of this controller satisfies each of the given specifications.

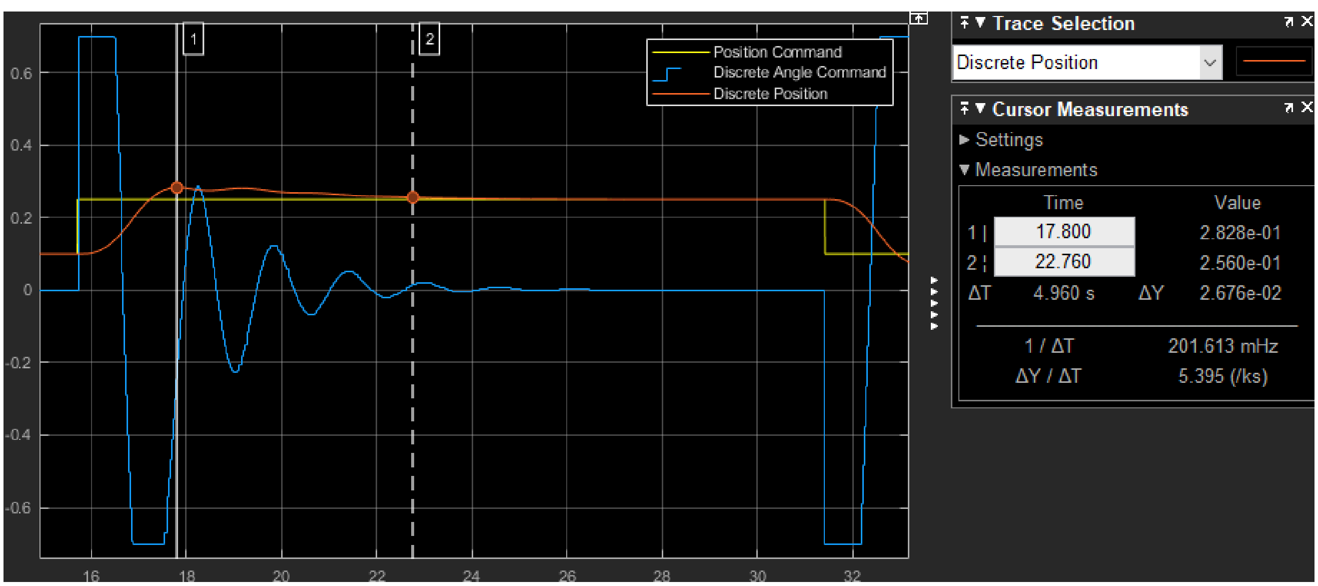


Figure 18: Discrete Controller Redesign Simulation Results

As shown in figure 18, running this simulation for the discrete version of this controller results in an overshoot of about 21.66% with a settling time of about 7.052 seconds. We find that the overshoot of our discrete controller is much lower than the continuous controller while the settling time is much higher. While the overshoot is better but the settling time is worse, we find that both of these values are within our specifications which satisfies our overshoot and settling time specifications. Again, we find that the command angle saturates and that the steady state error is well within 4%. This means that all the specifications for our discrete controller are satisfied.

Now that we have our discrete controller in order to implement it on the experimental setup we must first find the difference equation. This difference equation is as shown in equation 16 and was implemented into LabView as shown in appendix A.

$$\begin{aligned} u[k] = 33.86 \cdot e[k] - 32.81 \cdot e[k-1] - 33.85 \cdot e[k-2] + 32.82 \cdot e[k-3] \\ + 2.301 \cdot u[k-1] - 1.754 \cdot u[k-2] + 0.4527 \cdot u[k-3] \end{aligned} \tag{16}$$

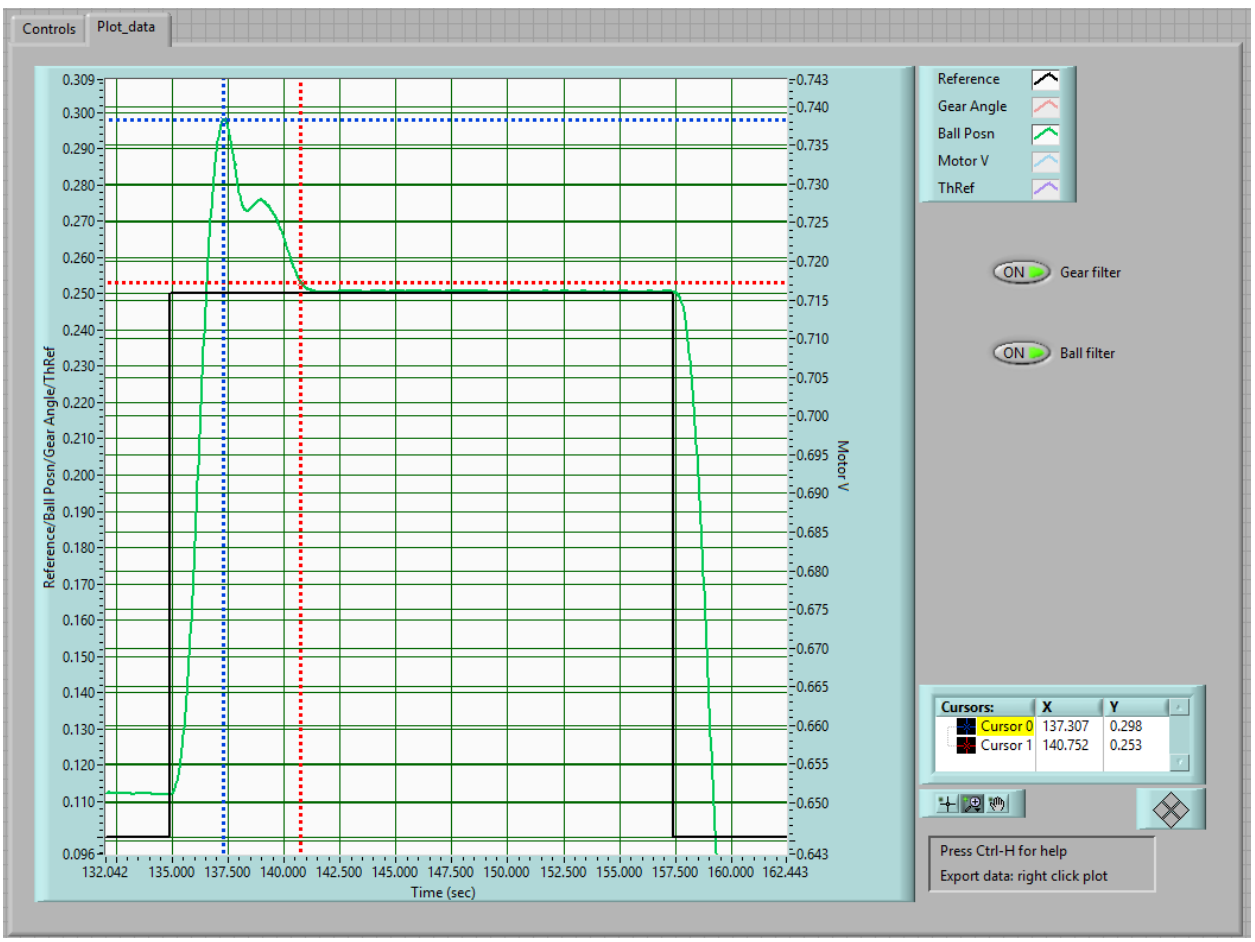


Figure 19: Controller Redesign Implementation Results

As shown in figure 19 once we implemented our controller into LabView we found that when stepping from 0.1 m to 0.25 m the system had an overshoot of about 32% and a settling time of about 5.752 seconds. This means the settling time and overshoot specifications are satisfied. As shown in figure 19 we know that the horizontal red line is the 2% mark of the command and we find that the steady state error of our system is will below that line. This means that the steady state error of our system is well within the 4% specification that is required. This means that this controller satisfies all the specifications required by our system.

9.0 Full Control System Block Diagram

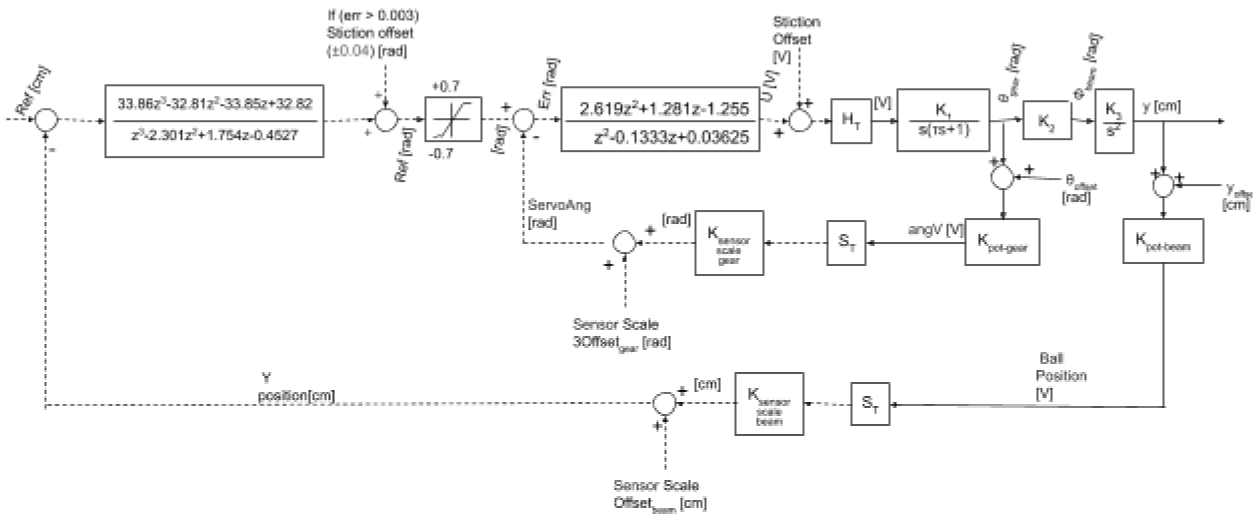


Figure 20: Full Control System Block Diagram

10.0 Conclusion

The outer loop controller was successfully designed and implemented using a pole placement method similar to the inner loop controller. The controller was designed to allow for tracking of a square wave input from 0.01 m to 0.25 m according to the specifications outlined in the lab manual.

In addition to the controller, the required reference angle to overcome ball stiction was experimentally determined and implemented in labview, further removing steady state error. In the end, the controller was able to track the reference signal with 32% overshoot, 5.752 s settling time and <4% steady state error.

To conclude, the final controllers used in the system were:

$$C_{inner}[z] = \frac{2.619z^2+1.281z-1.255}{z^2-0.1333z+0.03625} \tag{17}$$

$$C_{outer}[z] = \frac{33.86z^3-32.81z^2-33.85z+32.82}{z^3-2.301z^2+1.754z-0.4527} \tag{18}$$

And the parameters used in the system are shown in table 1.

Table 1: System Parameters

Parameter	Value
Ball Stiction [rad]	±0.04
Motor Stiction [V]	0.15 (CW), -0.15 (CCW)
K_1	1.88
τ	0.068
K_2	0.061
K_3	4.78
$K_{sensor\ scale\ gear}$	-0.7334
$K_{sensor\ scale\ beam}$	9.375
Sensor Scale $Offset_{gear}$ [rad]	-6.287
Sensor Scale $Offset_{beam}$ [m]	-0.3056

References

[1] C. Caradima, *Ball and Beam Lab Project Student Handout*. 2019.

[2] J. Simpson-Porco, "ECE 484: Digital Control Applications Lecture Notes", 2019.

Appendix A

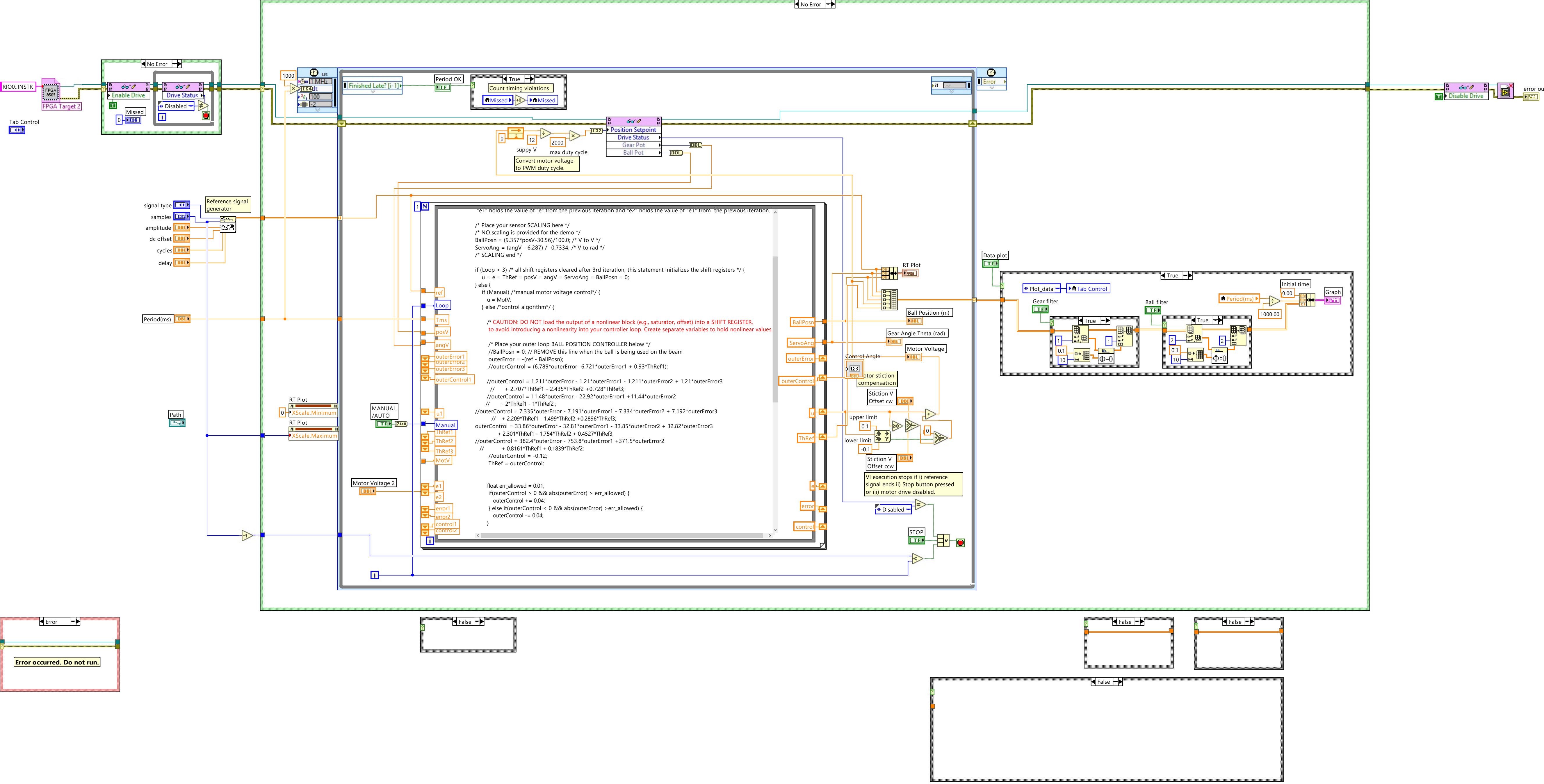
Formula Node Code:

```
float Temp1;
float eGearAng;
BallPosn = (9.357 * posV - 30.56) / 100.0;
ServoAng = (angV - 6.287) / -0.7334;
if (Loop < 3) {
  u = e = ThRef = posV = angV = ServoAng = BallPosn = 0;
} else {
  if (Manual) {
    u = MotV;
  } else {
    outerError = -(ref - BallPosn);
    outerControl = 33.86 * outerError - 32.81 * outerError1 - 33.85 *
                   outerError2 + 32.82 * outerError3 + 2.301 * ThRef1 - 1.754
                   * ThRef2 + 0.4527 * ThRef3;
    ThRef = outerControl;
    float err_allowed = 0.003;
    if (outerControl > 0 && abs(outerError) > err_allowed) {
      outerControl += 0.04;
    } else if (outerControl < 0 && abs(outerError) > err_allowed) {
      outerControl -= 0.04;
    }
    if (outerControl >= 0.7) {
      outerControl = 0.7;
    } else if (outerControl <= -0.7) {
      outerControl = -0.7;
    }
    error = outerControl - ServoAng;
    float b0, b1, b2, a0, a1, a2;
    b0 = 1;
    b1 = -0.1333;
    b2 = 0.03625;
    a0 = 2.619;
    a1 = 1.281;
    a2 = -1.255;
    control = -(-b1 * control1 - b2 * control2 + a0 * error + a1 * error1 + a2
            * error2) / b0;
    u = control;
  }
}
```

Appendix B

Matlab Controller Design Script

```
format long;
s = tf('s');

k2 = 0.061;
k3 = 4.78/(s^2);
p2 = k2*k3/s;

% Part C Controller
real1 = -0.01;
img1 = 0.01;
real2 = -1.7;
img2 = 0.9;
pole3 = -7.5;

% Part E Controller
% real1 = -0.7;
% img1 = 0.4;
% real2 = -8;
% img2 = 0.5;
% pole3 = -10;

Lambda2 = [real1+img1*1i, real1-img1*1i, real2+img2*1i, real2-img2*1i,
            pole3];
C2 = pp(p2,Lambda2)

stepinfo(feedback(p2*C2, 1))
bw = bandwidth(feedback(p2*C2, 1))
T = 0.029012;

discreteController = c2d(C2/s, T, 'tustin')

[y,t] = step(minreal(feedback(p2*C2,1)));
figure('Position', [300, 100, 800, 395]);
plot(t, y, 'b', 'linewidth', 2);
grid on;
set(gca, 'FontSize', 16);
xlabel('Time (s)','interpreter','latex','FontSize',20);
ylabel('Angular position (radians)','interpreter','latex','FontSize',20);
xlim([0,max(t)])
```


No Error
RIO0::INSTR
FPGA Target 2
Tab Control
Enable Drive
Drive Status
Disabled
Missed
1000
us
1 MHz
dt
Finished Late? [i-1]
Period OK
True
Count timing violations
Missed
Error
Position Setpoint
Drive Status
Gear Pot
Ball Pot
suppy V
12
2000
max duty cycle
Convert motor voltage to PWM duty cycle.
signal type
samples
amplitude
dc offset
cycles
delay
Reference signal generator
Period(ms)
RT Plot
XScale.Minimum
XScale.Maximum
Path
MANUAL /AUTO
Motor Voltage 2
"e1" holds the value of "e" from the previous iteration and "e2" holds the value of "e1" from the previous iteration.
/* Place your sensor SCALING here */
/* NO scaling is provided for the demo */
BallPosn = (9.357*posV-30.56)/100.0; /* V to V */
ServoAng = (angV - 6.287) / -0.7334; /* V to rad */
/* SCALING end */
if (Loop < 3) /* all shift registers cleared after 3rd iteration; this statement initializes the shift registers */ {
u = e = ThRef = posV = angV = ServoAng = BallPosn = 0;
} else {
if (Manual) /*manual motor voltage control*/ {
u = MotV;
} else /*control algorithm*/ {
/* CAUTION: DO NOT load the output of a nonlinear block (e.g., saturator, offset) into a SHIFT REGISTER,
to avoid introducing a nonlinearity into your controller loop. Create separate variables to hold nonlinear values.
/* Place your outer loop BALL POSITION CONTROLLER below */
//BallPosn = 0; // REMOVE this line when the ball is being used on the beam
outerError = -(ref - BallPosn);
//outerControl = (6.789*outerError -6.721*outerError1 + 0.93*ThRef1);
//outerControl = 1.211*outerError - 1.21*outerError1 - 1.211*outerError2 + 1.21*outerError3
// + 2.707*ThRef1 - 2.435*ThRef2 +0.728*ThRef3;
//outerControl = 11.48*outerError - 22.92*outerError1 +11.44*outerError2
// + 2*ThRef1 - 1*ThRef2 ;
//outerControl = 7.335*outerError - 7.191*outerError1 - 7.334*outerError2 + 7.192*outerError3
// + 2.209*ThRef1 - 1.499*ThRef2 +0.2896*ThRef3;
outerControl = 33.86*outerError - 32.81*outerError1 - 33.85*outerError2 + 32.82*outerError3
+ 2.301*ThRef1 - 1.754*ThRef2 + 0.4527*ThRef3;
//outerControl = 382.4*outerError - 753.8*outerError1 +371.5*outerError2
// + 0.8161*ThRef1 + 0.1839*ThRef2;
//outerControl = -0.12;
ThRef = outerControl;
float err_allowed = 0.01;
if(outerControl > 0 && abs(outerError) > err_allowed) {
outerControl += 0.04;
} else if(outerControl < 0 && abs(outerError) >err_allowed) {
outerControl -= 0.04;
}
ref
Loop
Tms
posV
angV
outerError1
outerError2
outerError3
outerControl1
u1
Manual
ThRef1
ThRef2
ThRef3
MotV
e1
e2
error1
error2
control1
control2
BallPosn
ServoAng
outerError
outerControl
u
ThRef
e
error
control
RT Plot
Ball Position (m)
Gear Angle Theta (rad)
Motor Voltage
Control Angle
Motor stiction compensation
Stiction V Offset cw
upper limit
0.1
lower limit
-0.1
Stiction V Offset ccw
VI execution stops if i) reference signal ends ii) Stop button pressed or iii) motor drive disabled.
Disabled
STOP
Data plot
True
Plot_data
Tab Control
Gear filter
Ball filter
Period(ms)
Initial time
0.00
1000.00
Graph
Disable Drive
error out
Error
Error occurred. Do not run.
False