**Cart mounted Inverted Pendulum passing through Pinwheel**

ECE 4375

Date: 5/4/2021

Name: Christopher Andrew

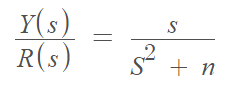
Student ID: 1885928

**1 Executive Summary**

The goal of this project was to create a control system capable of balancing an inverted pendulum mounted on top of a cart and then having the pendulum pass through a rotating pinwheel to a location on the other side.

**2 Introduction**

Most of our work for this project will be in the Laplace domain and will be in the form of taking transfer functions representations of systems and adding or multiplying them together. A typical transfer function appears in a form like

 , (2.1)

and describes the relationship between two variables in the Laplace domain. This generally consists of two polynomials with the numerator having a less than or equal order to the denominator.

These transfer functions can be assembled into blocks and summed or multiplied together to represent more complex systems. An example of this would be a generic control system with unity negative feedback, or constant negative feedback seen in figure 2.1. This basic control system attempts to bring the output Y(s) to equal the input U(s). If the performance of this system is not adequate, a variety of controllers can be inserted in between the summation block and the core transfer function to attempt to improve performance or stability.

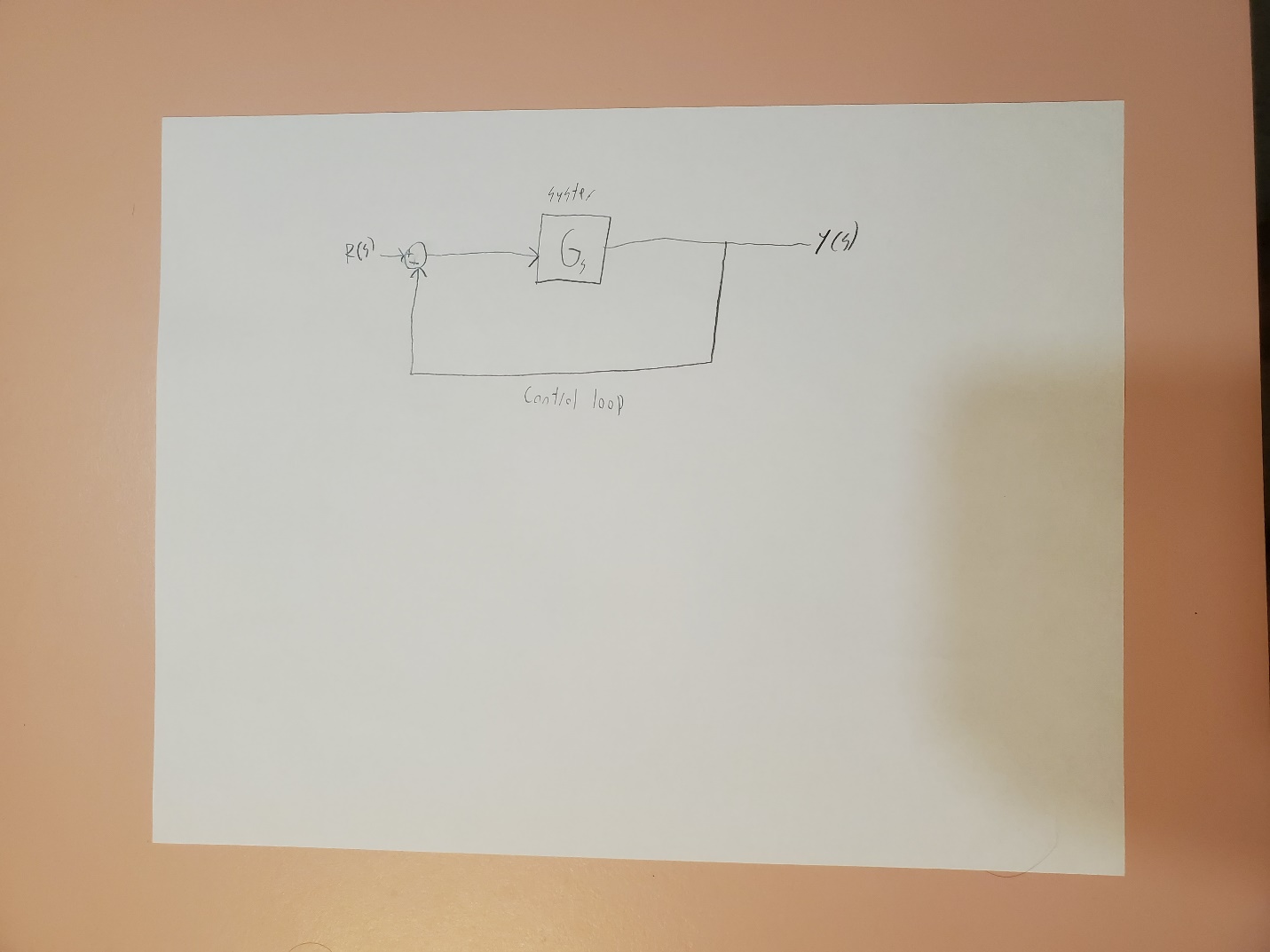


Figure 2.1 A generic Control system

We will be using PID controllers for this project which will allow us to greatly alter the performance and stability of systems.

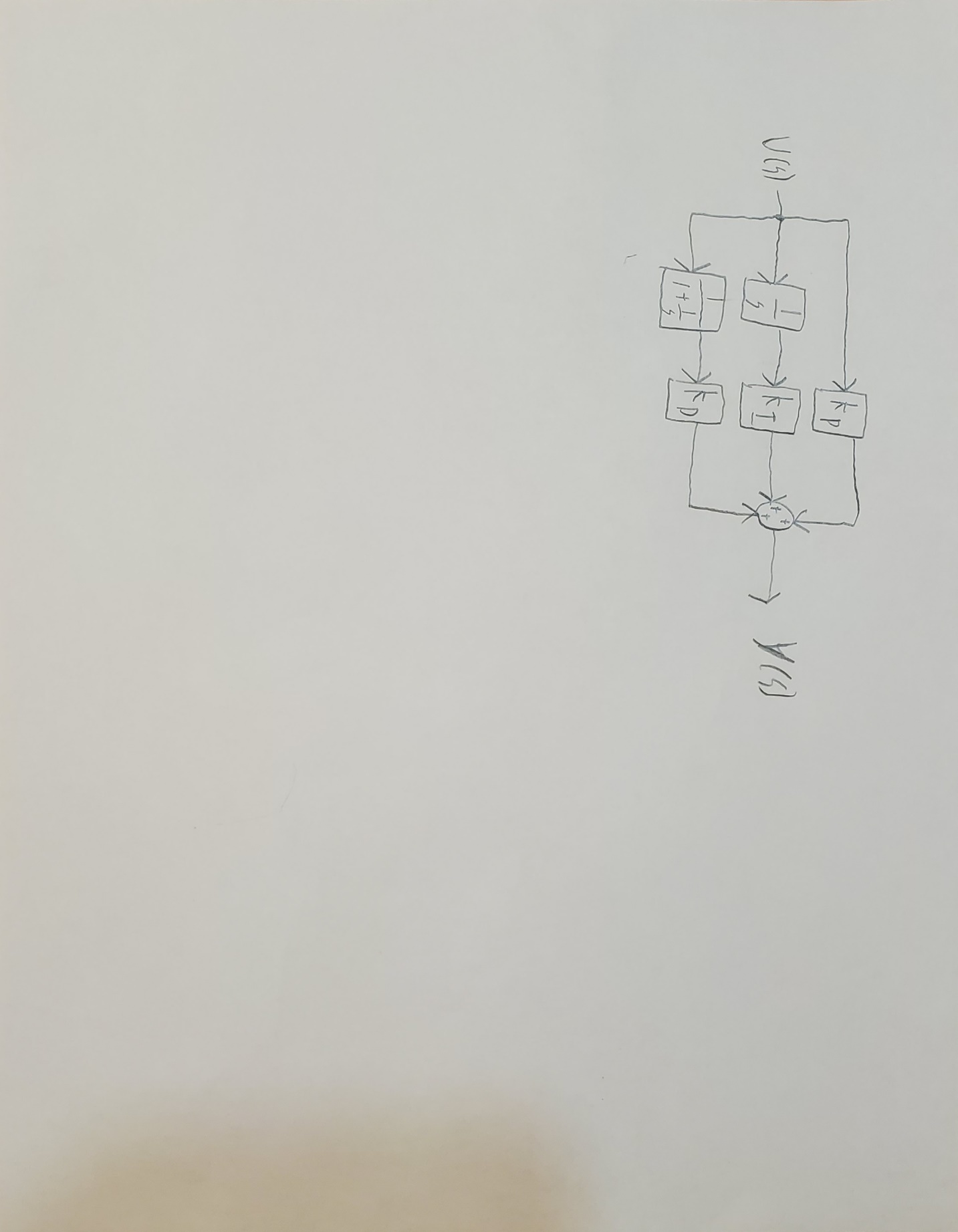


Figure 2.2 PID Controller

**3 Background**

The problem that we are here to solve has 3 core goals that must be reaches to consider our solution complete. We must have the cart reach a controlled position on the other side of the pin wheel. The second goal is that the inverted pendulum must remain upright with a minimal deviation in angle or our linearization of the physics simulation will begin to deviate significantly from reality. The 3rd goal is that the pendulum must pass through one of the slots of the pinwheel. A final constraint is that the system must not exceed the voltage and current maximums of the motor used.

**4 Solution**

4.1 Model Derivation

To model the system, we need to both derive a representation of our system and define our system. Starting with the cart and pendulum it has 3 sub parts, the motor, the torque to linear force system, the cart with pendulum.

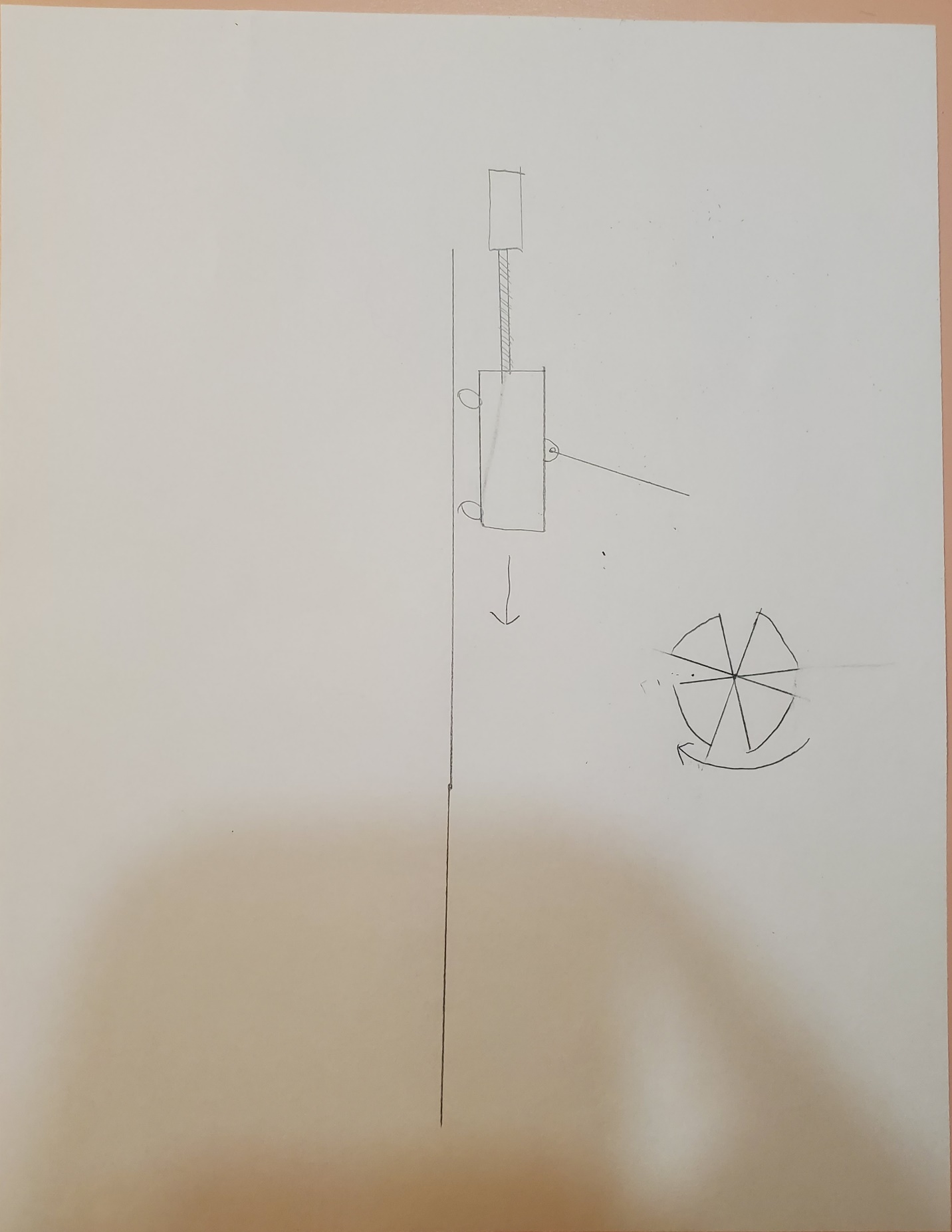


Figure 4.1.1 System Overview

In figure 4.1.2 you will see we will be using a standard 2nd order Laplace domain motor model to model our DC servo motor. [3]

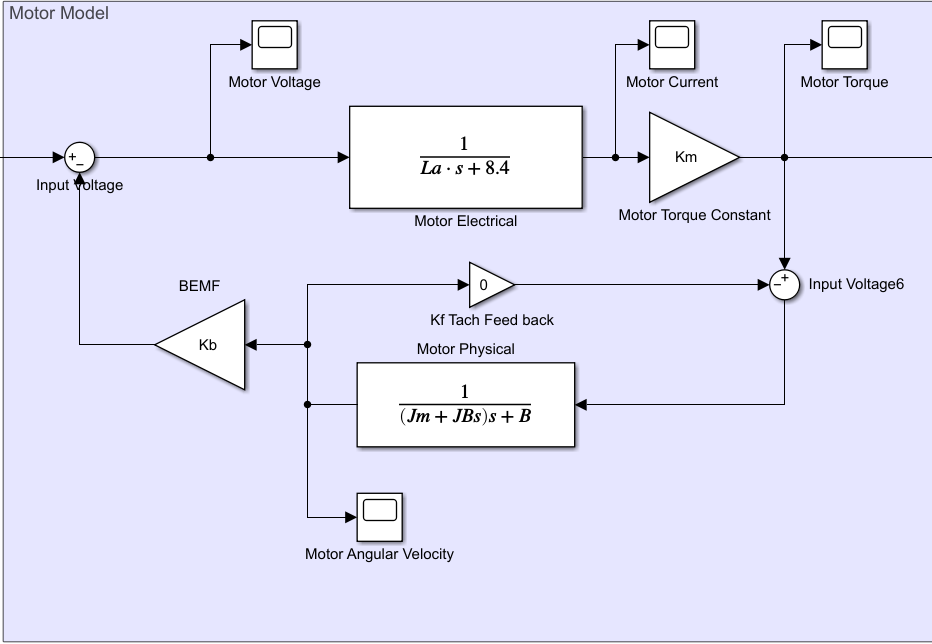


Figure 4.1.2 2nd order Laplace DC servo motor model for cart driver

To model the cart, we first constructed a free body diagram, figure 4.1.3, containing all the relevant forces required to define our cart model. Once we have this diagram, we proceeded to assemble a system of differential equations in the time domain to represent this part of our system.

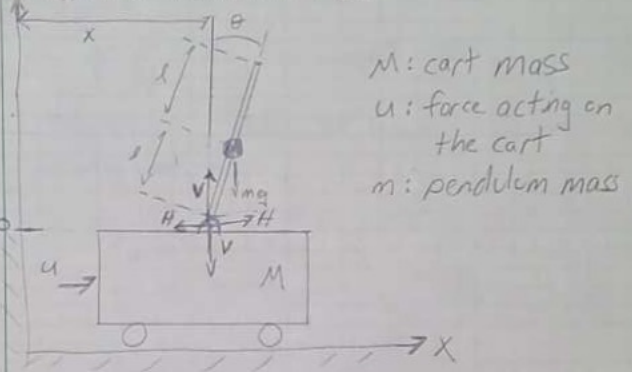
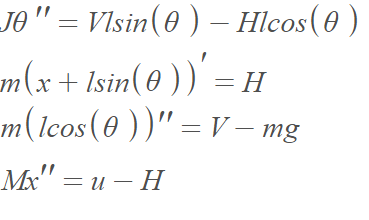


Figure 4.1.3 Cart Free Body Diagram [2]

 (4.1.1) [2]

From this system of differential equations, we then proceeded to linearize these equations and convert them into simplified Laplace domain expressions, we then used the resulting equations to form a block diagram seen in Figure 4.1.4 to represent the cart and pendulum component of the system. [2]

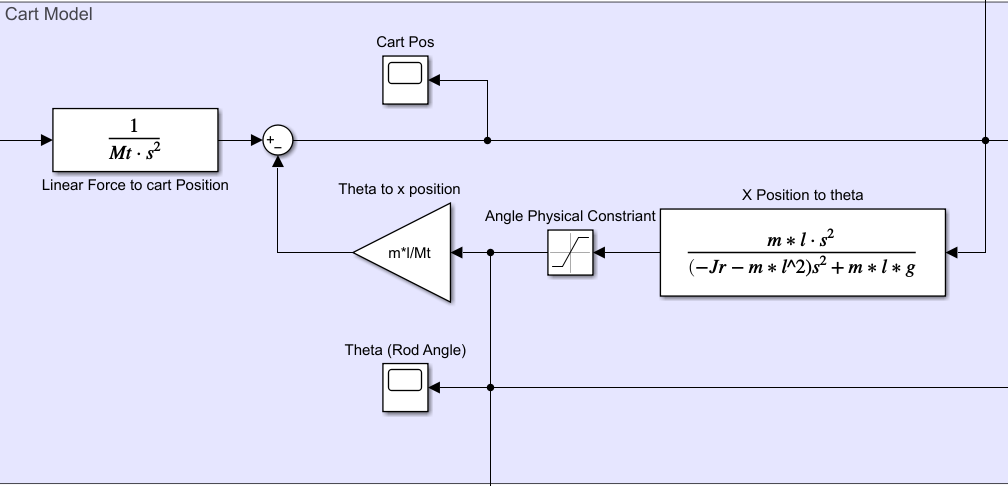


Figure 4.1.4 Resulting cart block diagram

To use our motor to apply linear force to our cart we ran into a problem in that the motor produces rotational speed and or torque, while our cart model requires a linear force to be applied. We will solve this problem by using a ball screw device to convert torque to linear force via,

Linear Force = ((Applied Torque)\*200\*pi\*0.9). (4.2.2) [1]

To simulate the pin wheel, I started with our base 2nd order model for a DC servo motor and took the output for angular velocity and converted it into angular position via integration. As we now have a rotating angle, to make the wheel rotate, we defined all the transition lines between safe and unsafe regions in terms of this changing angle. Figure 4.1.6 Using these defined transition lines, a check was implemented to determine if the 0 [RAD] position, the edge the pendulum passes through, is inside any of these safe regions. Figure 4.1.5

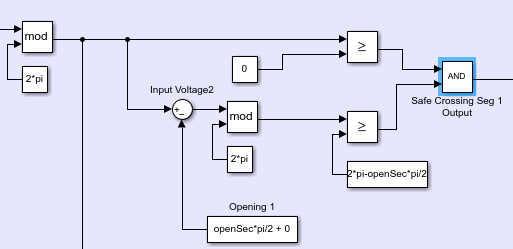


Figure 4.1.5 Pinwheel safe crossing piecewise for one segment

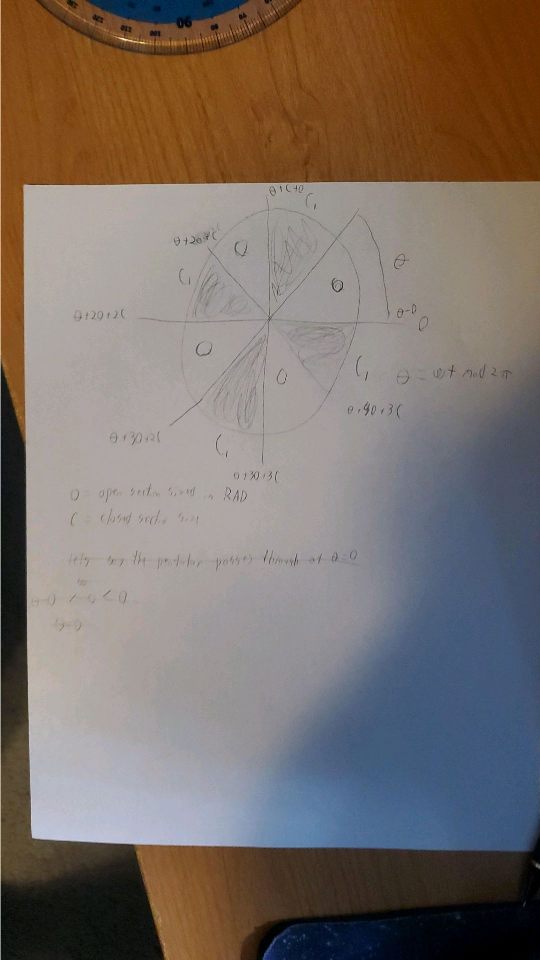


Figure 4.1.6 Pinwheel section relation chart

These checks produce a PWM, pulse width modulation, like signal which will be used latter for relative timing applications and verifying the model result that the pendulum passed through one of these safe regions.

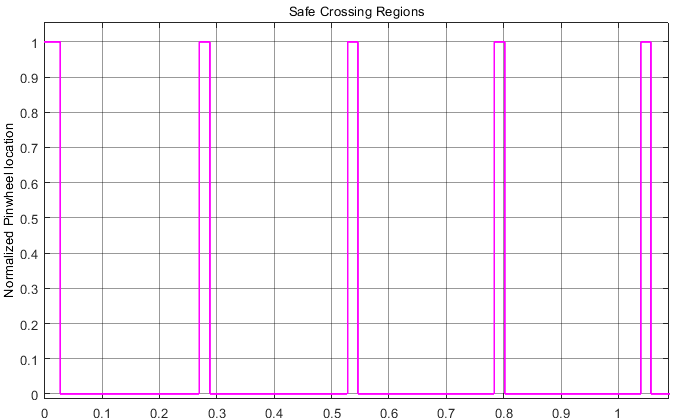


Figure 4.1.7 Safe Crossing Waveform during pinwheel motor start

Physical size and other constraints of the system were picked a random with a bias to values around 1, or other small values. This range of values were chosen due to being unsure of the power the provided servo motors could provide, in retrospect they could be much higher. The motor friction coefficients were determined by feeding the motor model 18 volts and adjusting the friction coefficient till they reached nominal speed per the motor spec sheet.

4.2 Control Approach

When looking at this problem we see that there are 3, really 4, goals that must be reached to form a complete solution. First, we must look at controlling the position of the cart and balancing its pendulum before even considering passing the pendulum through a pinwheel or limiting the voltage and current levels in the motor.

When searching for options for multivariable control systems these two controls systems options for PID system were found. Figure 4.2.1 The first option was to form concentric loops of chained summing junctions and PID controllers. The second option was assigning a PID controller to one control variable and then summing the outputs of all control variables. This option specifically was favored due to the ability to more freely tune both loops. [4]

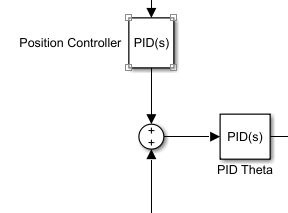


Figure 4.2.1 Two approaches to multi variable control.

Initial Values for both control loops PID controllers was generated by Simulink’s built in PID tuning feature and tuned independently before combined. These initial PID values were then tuned by had to create a starting point to step back the values to meet the voltage and current constraints.

The pinwheels control system in contrast is much simpler, it just needs to reach a target angular velocity through a simple reference track PID loop.

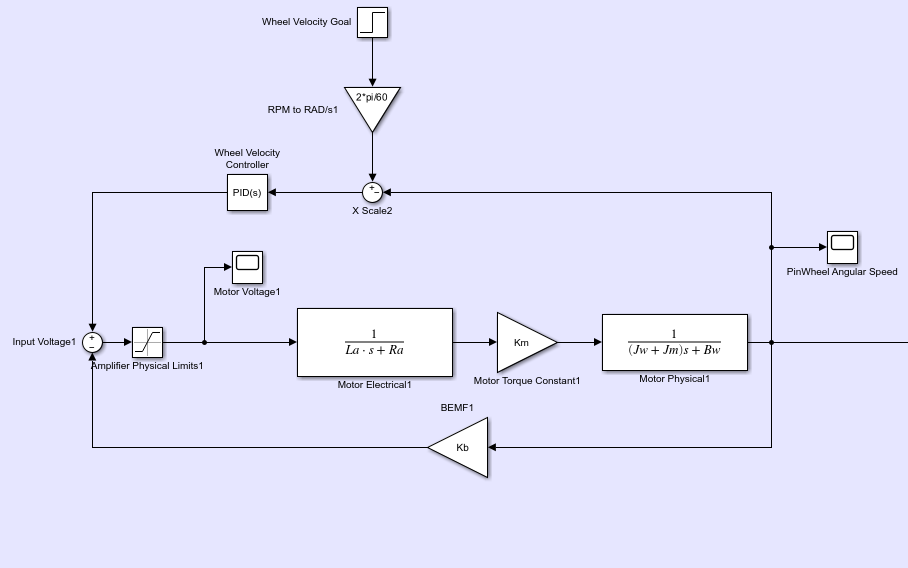


Figure 4.2.2 Pinwheel control block with motor model

The final control goal of this solution is to pass the balanced pendulum through the spinning slots of the pin wheel. This is done not by PID based systems but by measuring how long it takes the cart to pass the pinwheel position and calculating a delay value from this time to determine when the cart should be launched.

Section period \* (1 + open Fraction \* 0.5) – mod(Wheel intercept, Section period) (4.2.1)

(1/(RPM/60))\*(1/numSection)\*(1+openSec\*0.5) - mod(timeToWheel,(1/(RPM/60))\*(1/numSection)) (4.2.2)

Equation 4.2.1 is a simplified form of the expression used to generate our start delays. From this simplified form it is clear that this expression finds how far off our travel time is from the start of a safe crossing region at a given wheel speed, subtracts that from a modified section to period to shift the estimate away from the rising edge of a safe crossing region. This is the general form of what produces our relative start delay from a rising edge safe region with 4.2.2 being the messier implemented form used in MATLAB.

This relative delay value and safe crossing system is not enough on its own however, we need to change cart position goals and determine if the cart and pinwheel have reached a steady state. Multiple approaches to this were considered including waiting for the derivatives of each respected system output to approach zero. Ultimately, a simple timing system to count the number of rising edges of safe passing regions and then applying the relatively delay value to the changed x position goal was chosen instead.

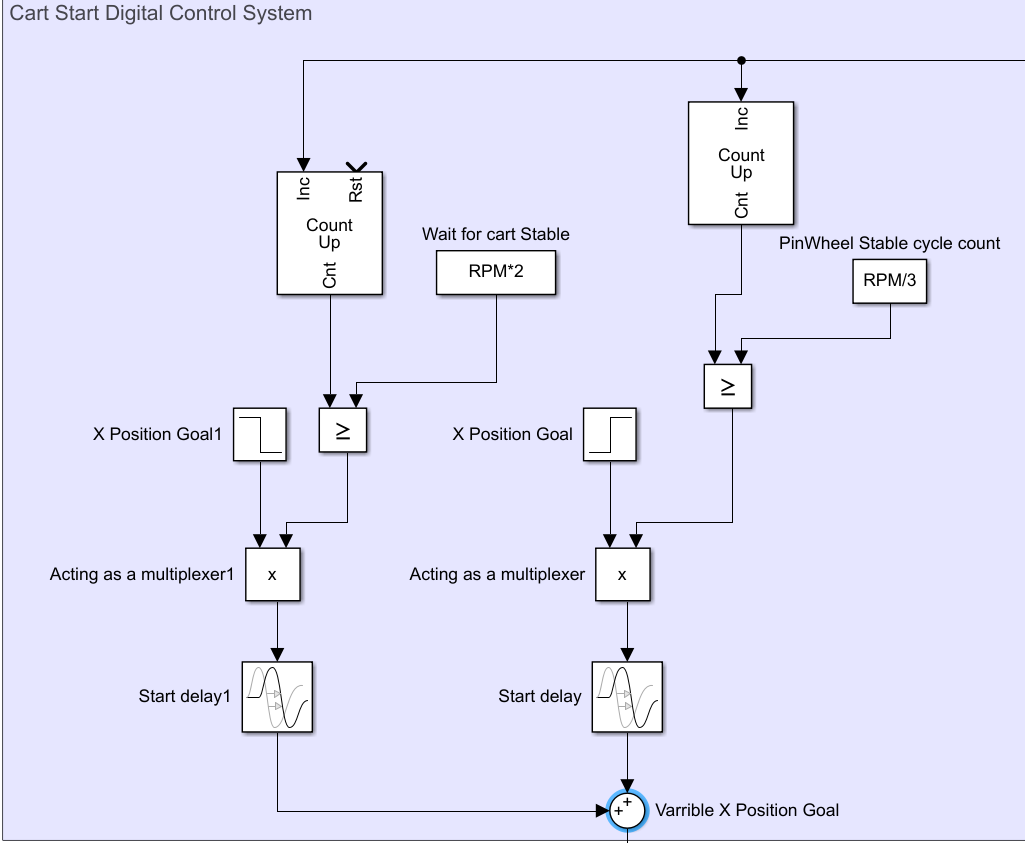


Figure 4.2.3 Safe Crossing Counting and relative delay system

4.3 MATLAB/Simulink Implementation

The software portion of this solution was divided into two part to make changing general configurations as simple as possible. The first part is a simple script, named “Lab\_Startup\_Script.m” , that loads the necessary variables into MATLAB’s workspace and calculates needed values from these constants. In addition, this script must be run again if a different RPM value for the pinwheel needs to be tested. The second part of the project is the model built in Simulink named “Inverted\_Pend\_Lab.slx” which runs the actual simulation and graphs the appropriate outputs and internal values. See the attached files for the exact files used or the appendix.

4.4 Controlled Model Results

Each of these crossing graphs consist of 3 signals, rod endpoint x position, rod start point x position, safe crossing regions. The rod endpoint is the location of the end of the rod. The rod start point is the x position of the rod where is enters the pendulum if the rod is vertical. If both lines cross the safe crossing region signal at 2 [m], the position of the pinwheel, the test case passes. In our model all 3 required speeds pass with an opening size of 7% of the sections arc length or 1.75% of the total circumference. In addition, as shown by Figures 4.4.1-3 the solution can be used to return through the pin wheel if desired. Finally, as can be seen from the voltage and current graphs each cart remains within the motor’s max values.

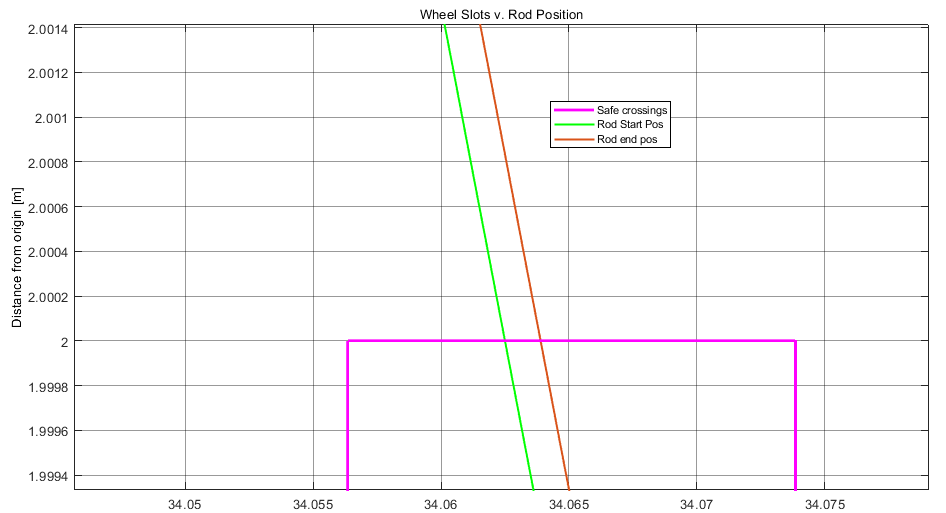
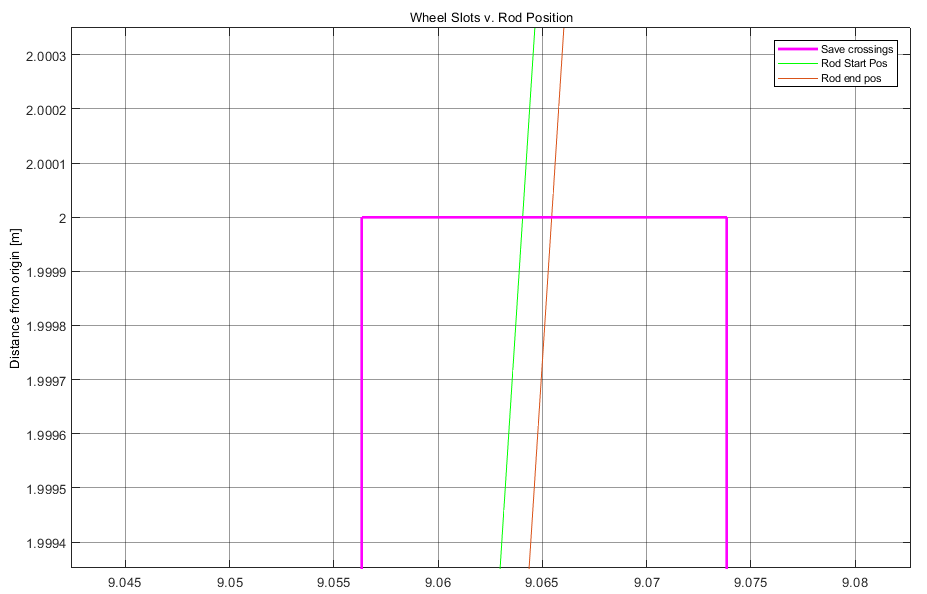


Figure 4.4.1 1 RPS Crossing (forward left, backwards right)

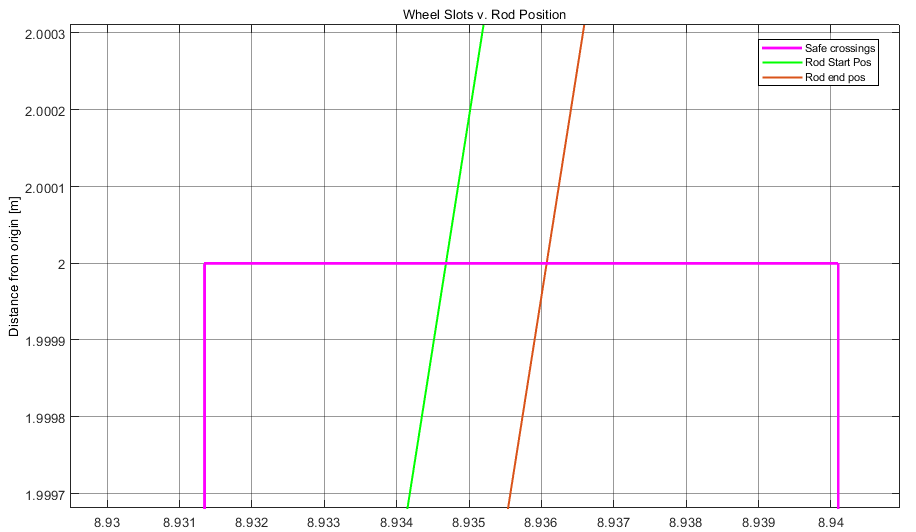
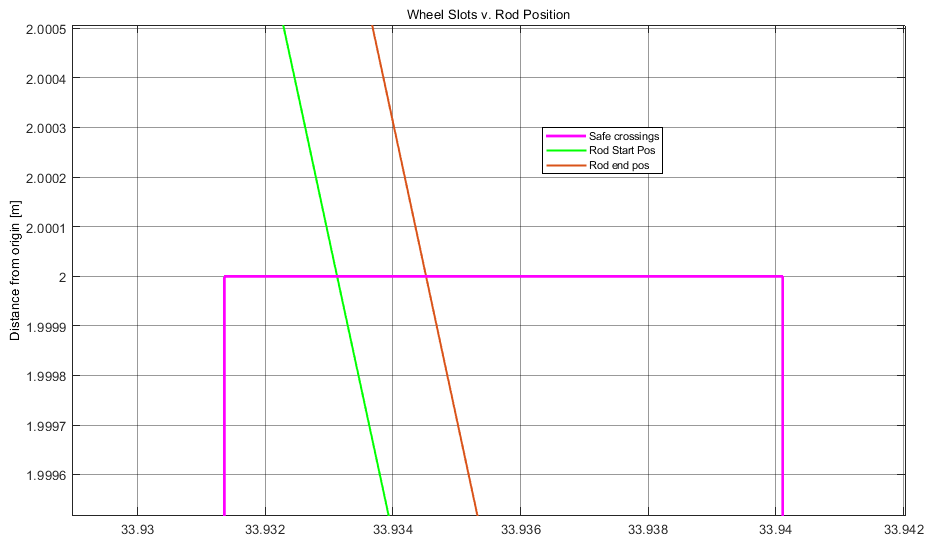
****

Figure 4.4.2 2 RPS Crossing (forward left, backwards right)

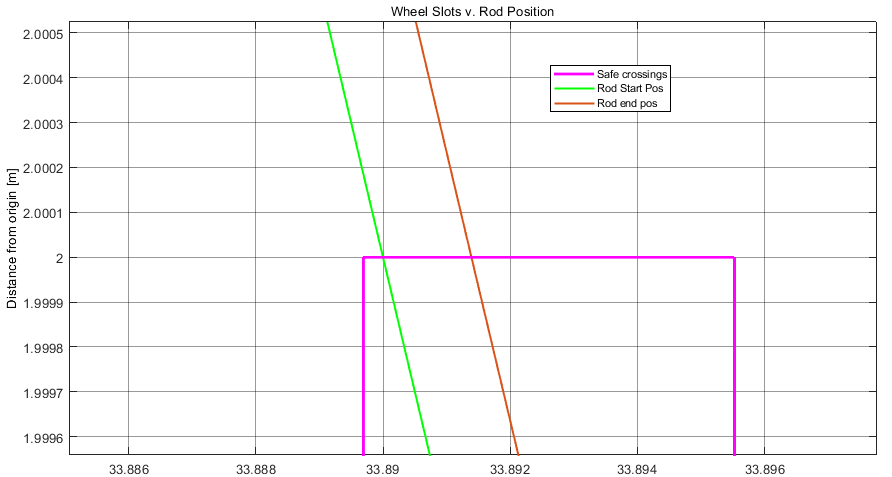
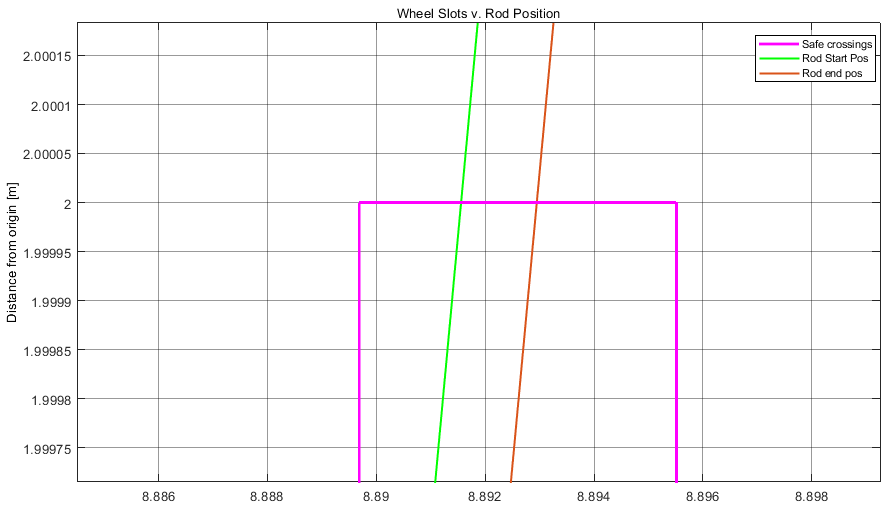


Figure 4.4.3 3 RPS Crossing (forward left, backwards right)

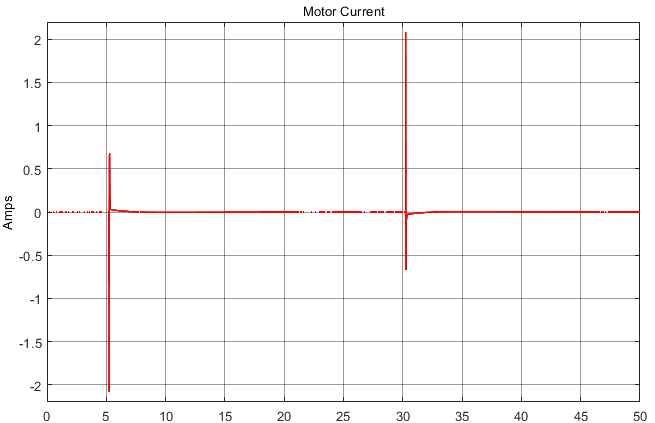
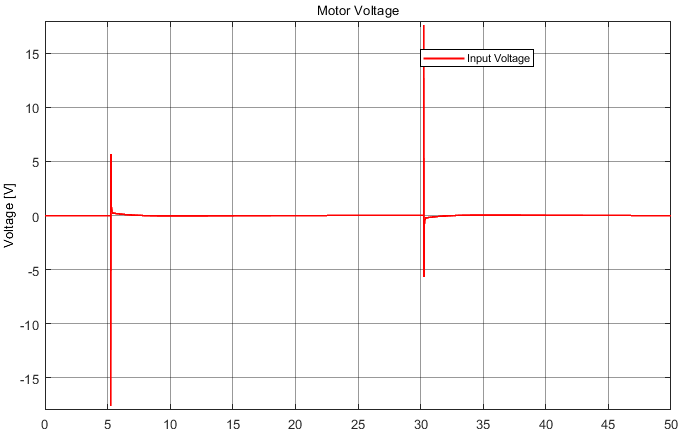


Figure 4.4.4 Motor Voltage and Current (Roughly Identical for all crossings)

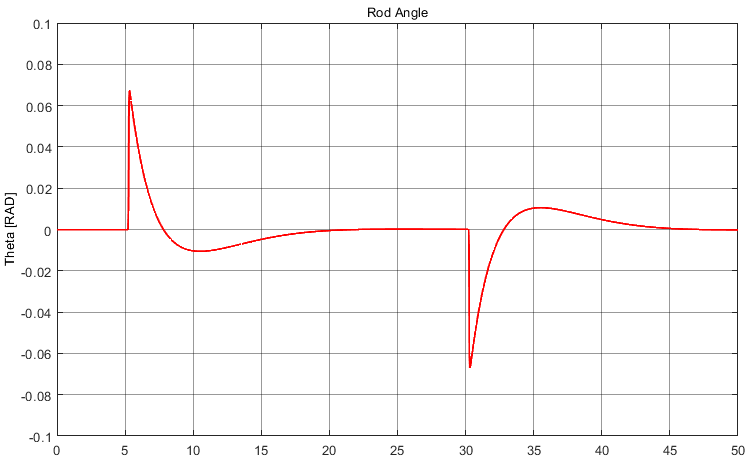


Figure 4.4.5 Rod Angle in RAD (Roughly Identical for all crossings)

5. Conclusion

The goal of this project was to create a controls solution that can control an inverted pendulum mounted on a moving cart, and then moving said cart through a rotating pinwheel. The solution created consists of 2 PID reference tracking control loops which are then summed together and fed into the carts motor. The pinwheel crossing is managed by creating a relative measure of its safe crossing regions and a calculated delay value to determine when the crossing should be attempted relative to a safe crossing region occurring. This resulted in successful crossings with the pinwheel having open slots sized to equal to 1.75% of its circumference.

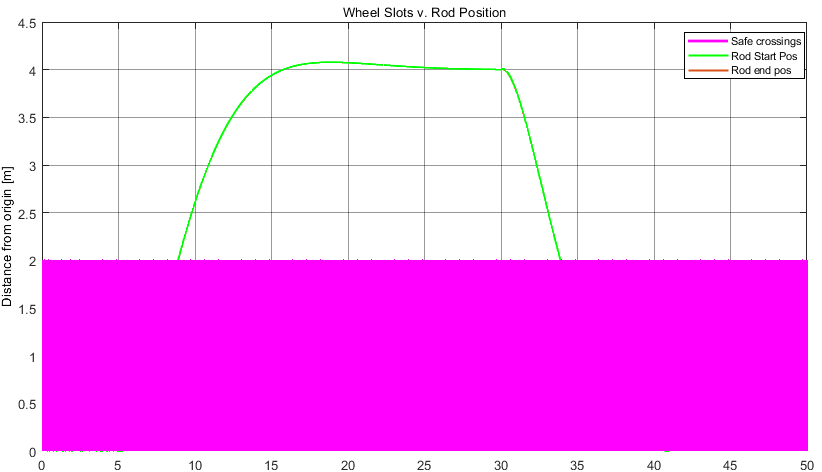


Figure 5.1 A complete Cart Circlet from start to start at 3 RPS

References

[1] “Linear Motion Systems,” Linear Rails, Guide Wheels, Steel Tracks, Bushings, Carriage Assemblies from SDP/SI. [Online]. Available: . [Accessed: 10-Apr-2021].

[2] R. S, Provence, “Cart Pendulum Derivation”, ECE 4375, University of Houston, Houston, 3/24/2021 [Accessed: 4-May-21]

[3] R. S, Provence, “dc servo motor models”, ECE 4375, University of Houston, Houston, 3/31/2021 [Accessed: 4-May-21]

[4] “Multivariable Control - Part 1,” YouTube, 06-May-2019. [Online]. Available: https://www.youtube.com/watch?v=mMtFuYeJp5A. [Accessed: 20-April-2021].

Appendix A: MatLab script

%created by: Christopher Andrew

%purpose: Loads the needed varribles into matlabs workspace to allow

%simulink to run

%motor Coefficients

B = 7.1600e-05 %Friction Coefficient

La = 1.16\*10^(-3) %Inductance

Km = 42\*10^(-3) %Torque Constant

Kb = 13.2/(3000\*2\*pi/60) %Back EMF constant

Jm = 4 \* 10 ^ (-6) %motor inertia

Ra = 8.4 %terminal resistance

%linear motion parameters

Bs = (2000\*pi\*0.9)/(10) % torque to linear force

Bsd = 0.01298 %diameter of the ball screw

BS\_length = 6 %ball screw length in meters

VBs = BS\_length\*(Bsd/2)^2 %volume of ball screw rod

MBs = VBs \* 7700 %mass of ball screw rod

JBs = MBs\*0.5\*(Bsd/2)^2 %inertia of ball screw assembly

%cart parameters

M = 1 %Cart Mass

Jc = ((0.25\*0.25^3)/12)\*M %cart inertia

%Rod parameters

m = 0.05 %rod mass

l = 0.1 %pendulum half size

Jr = (1/3)\*m\*(l\*2)^2 %rod inertia, may be wrong look more

%wheel parameters

wR = 0.11 %wheel radius, rod position at vertical is just inside this

Mw = 0.001\*(wR)^2\*pi\*200 % mass of the wheel in Kg, assumed to be

%carboard

numSection = 4 %number of sections in the wheel

openSec = 0.07 % The percentage of a quarter that is open %0.2 best

RPM = 1\*60 %speed of the wheel in revolutions per second

pinDistance = 2 %pinwheel distance from origin

Jw = 0.5\*Mw\*wR^2 %inertia of the wheel

Bw = 7.1700e-05 %wheel motor friction coefficient

%Other Parameters

Mt = M + m %total mass of the cart

%Jt = Jm + Jc + Jr %total Inirtia

g = 9.8 %acceleration due to gravity

distance = 4 %distance from the origin that must be traveled

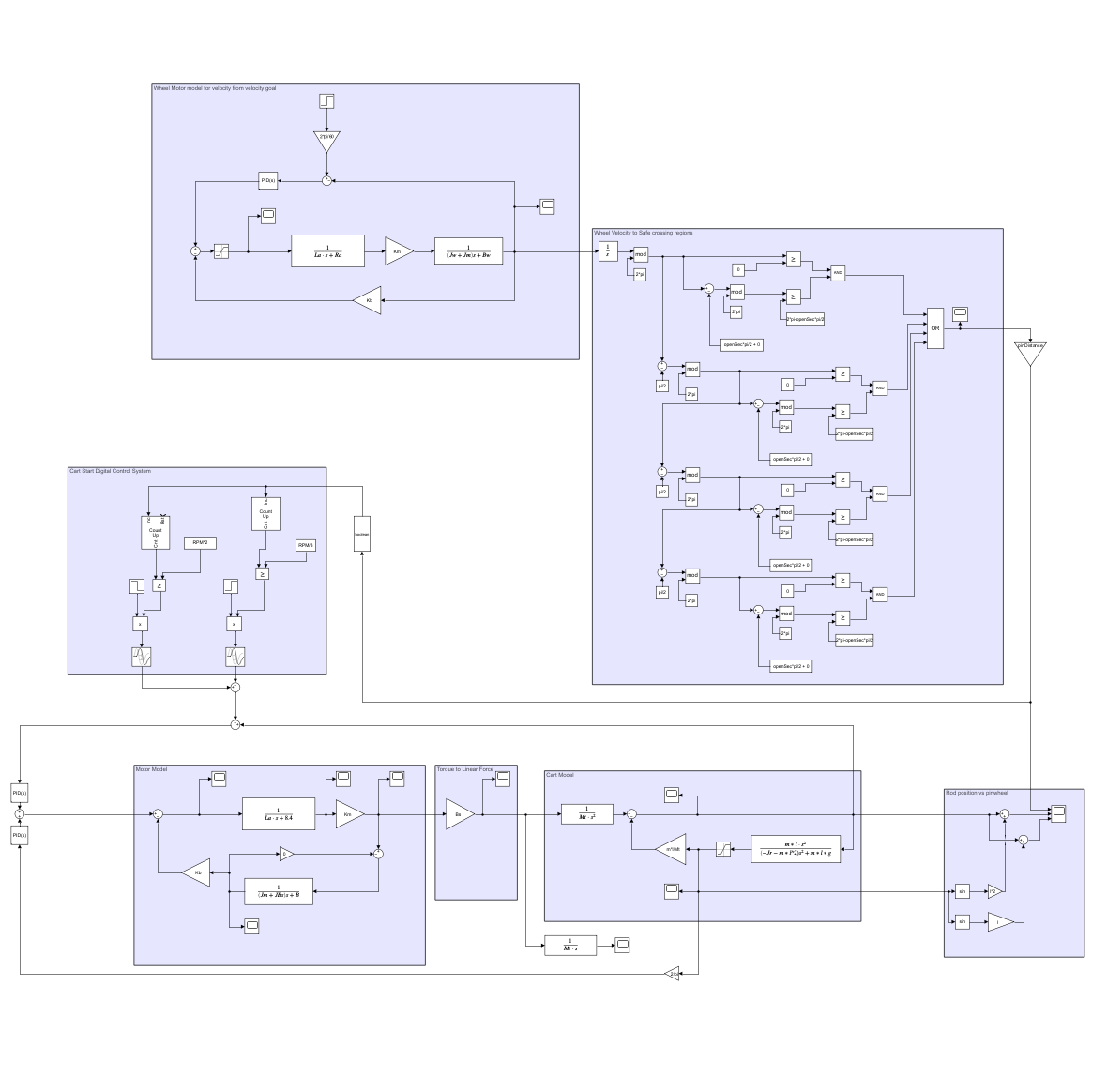
timeToWheel = 3.825 %magic number from prevous runs. time till the cart

%reaches the wheel, used to determin the start

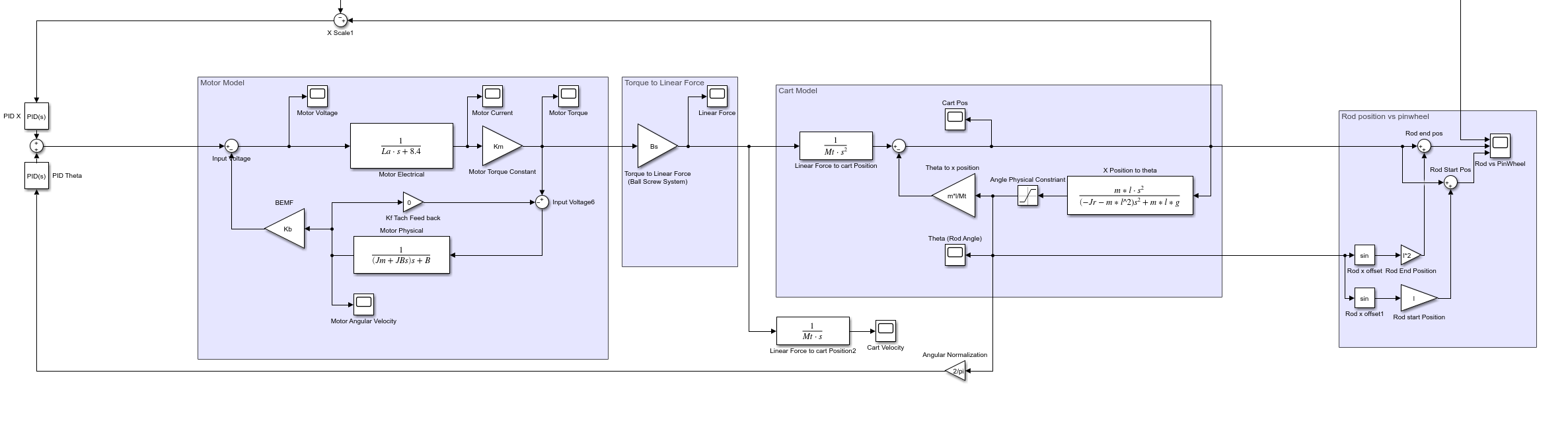
%delay

startDelay = (1/(RPM/60))\*(1/numSection)\*(1+openSec\*0.5) - mod(timeToWheel,(1/(RPM/60))\*(1/numSection))

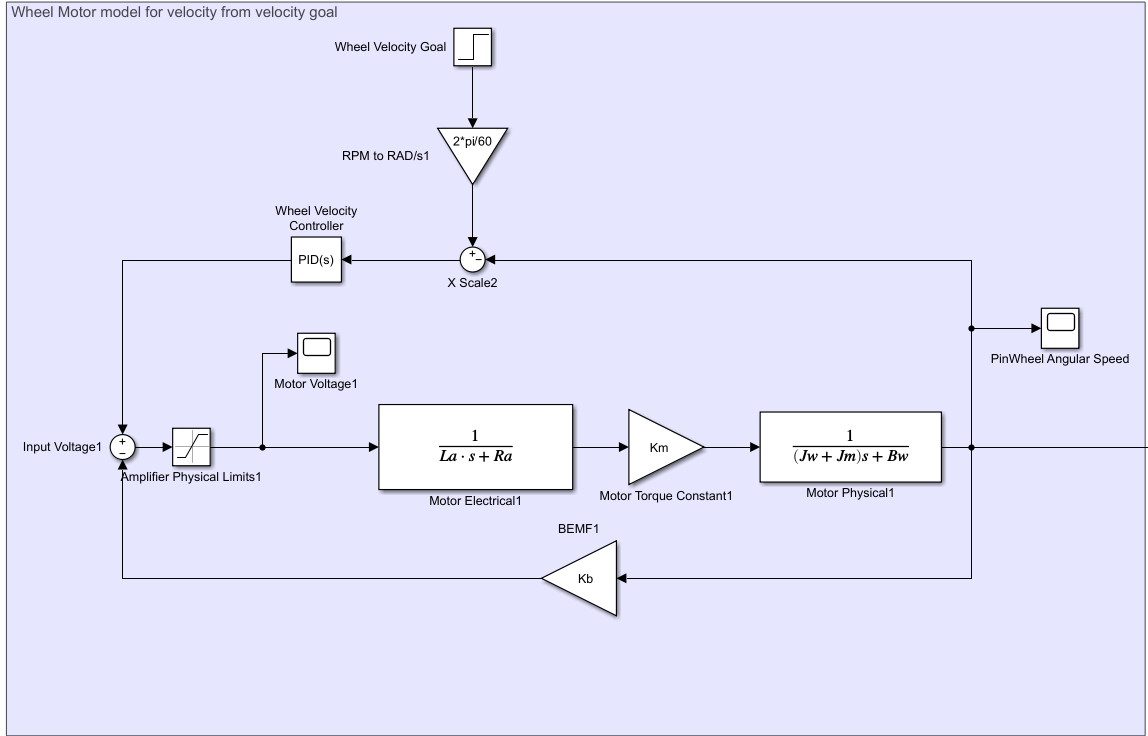
Appendix B: Simulink Model Screen Shots

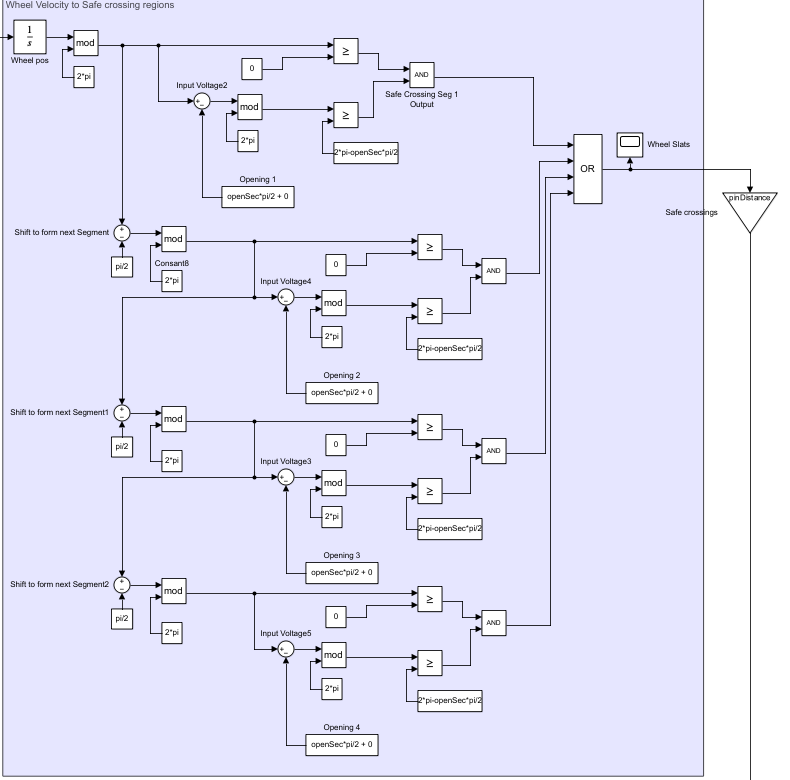


Overview of Simulink model

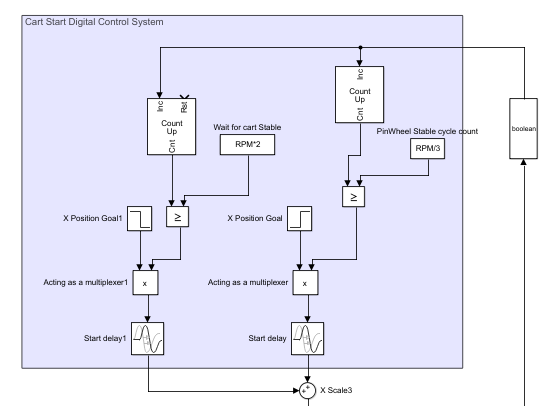


Cart Model

Wheel Motor Model



Wheel Safe crossings Generator



Cart position goal Controller