

MSE 312  
Integration Report

Anjandev Momi - 301281186  
Brindan Ramalingam - 301271568  
Neeraj Bansal - 301323495

August 15, 2019

# Contents

1	Abstract . . . . .	3
2	Introduction . . . . .	4
3	System Level Design . . . . .	4
	3.1 Demonstration Apparatus Setup . . . . .	6
4	Mechanical Design . . . . .	6
5	Electronics Design . . . . .	7
6	Controller Design . . . . .	9
	6.1 Controller PID values on the Arduino . . . . .	10
7	Software Interface . . . . .	11
	7.1 Power state Detection . . . . .	11
	7.2 Input Control . . . . .	11
	7.3 Motor Control . . . . .	13
	7.4 Motor Encoder reading . . . . .	13
	7.5 State . . . . .	15
8	Demonstration Results . . . . .	17
	8.1 Final PID Values . . . . .	17
	8.2 System Performance . . . . .	18
	8.3 Velocity and Position Plot . . . . .	20
	8.4 Electromagnet On Time plot . . . . .	21
	8.5 Demonstration Video . . . . .	21
9	Observations and Discussion . . . . .	21
	9.1 Test Reliability and selection of PID values . . . . .	21
	9.2 Discussion on system performance . . . . .	22
	9.3 Power protection, limit switch operation and system relay state . . . . .	23
	9.4 Testing the Ramp and Step Input variability . . . . .	23
10	Further Work - Additional Improvements . . . . .	24
11	Team member contribution . . . . .	25
12	Conclusion . . . . .	25
13	Appendix . . . . .	26
	13.1 Mechanical Report . . . . .	26
	13.2 Electrical Report . . . . .	50
	13.3 Control Report . . . . .	70

# List of Figures

- 1 System-level block diagram . . . . . 5
- 2 System Demonstration Apparatus . . . . . 6
- 3 Design and mechanical properties . . . . . 7
- 4 Electrical Circuit Board . . . . . 8
- 5 Electromagnet Control Circuit . . . . . 9
- 6 Power protection relay switch setup feeding system . . . . . 9
- 7 Closed loop signal flow diagram . . . . . 10
- 8 Simulink model for outputting ramp for 180 degrees . . . . . 12
- 9 Simulink model for outputting ramp for 90 degrees . . . . . 12
- 10 Voltage out of PID to PWM signal converting simulink model . . . . . 13
- 11 Simulink model for encoder reading . . . . . 14
- 12 State diagram . . . . . 15
- 13 Simulink model for driving electromagnet . . . . . 17
- 14 Simulink scope plot for required position vs. encoder value capture . . . . . 19
- 15 Simulink scope plot for system velocity and encoder position . . . . . 20
- 16 Simulink scope plot for electromagnet control signal vs. time . . . . . 21

# 1 Abstract

The MSE312 term project required teams to design, build and control a pick and place arm to rotate  $180^\circ$  pick up a metal puck and rotate  $90^\circ$  to drop the target. Mechanical design, electrical and protection circuitry, control system and integration aspects were designed and evaluated for the ultimate goal of the system to perform the required task in the fastest time possible while being robust and reliable.

The mechanical truss arm was constructed using brass rods under project constraints [2] and conceptualized around minimizing mass moment of inertia, centre of mass close to the axis of rotation and minimal deflection under FEA analysis. Concept selection analysis was performed to appropriately select the final mechanical design.

The electronics design consisted of motor circuit drive, electromagnet circuitry, power protection and configuration setup with the apparatus limit switches. The electrical configuration provided a soldered circuit board for 12V power to the system and compatibility with 5V signal which would be used for the PWM control signal of the DC motor and electromagnet.

The initial control design evaluated electrical/mechanical system continuous time transfer function which was used to plot the system root locus and evaluate PID controller parameters. Integrating the system with a digital micro controller the system is need take into consideration the Z-domain therefore PID value were tuned to operate with the set sampling period. In integration of the system using the micro-controller was inter-phased using Simulink. The logic of the control program conceptually used a finite state machine for correct sequential operation of the system. Using the integration of the program final PID values were tuned to two individual sets for each  $180^\circ$  and  $90^\circ$ . The final system performance demonstrated accurate pick and place in a avg. run time of 2.03s with a over damped system response. Designing the system to be over damped was critical as an overshooting system with varying momentum in the system resulted in unpredictable overshoot values. Attaining close to critical damping response by tuning the system, the motor current flowed in one direction for one interval of travel.

The system was assessed for further improvements on mechanical, electrical, controls and integration subsections to order to evaluate aspects of improvement for the system for further improvements in future similar projects and considerations for integrating a control system.

## 2 Introduction

MSE 312 term project requires teams to develop a one degree of freedom pick and place arm spanning 30cm to carry a payload from point A to B accurately in the fastest time. This system is electrically powered and controlled and is mechanically built using brass rods. Using the systems mechanical characteristics and the system properties provided in the data sheets, the controls section of this project allowed us to understand how the mechanical system adapts to given inputs and control parameters. It was determined a PID control was required to meet the desired output characteristics of the system. The control will be provided from an Arduino micro-controller which will be integrated using Simulink software. In developing mechanical, electrical and control sub systems of this project, we look to integrate the built components of the system to carefully control the system. The requirements of the integration are:

- Precisely move from a start position of 0 to 180 degrees CCW
- Pick up puck using electromagnet and hold until final position
- Drop puck 90 degrees CW into target position hole to trigger limit switch 3 (at velocity of 0)
- Limit 3 three activation to trigger program to stop timer and display run time to two decimal places of accuracy
- Process to complete task in fastest time possible - Integrate and tune PID controller as required
- Limit switch 1, 2 upon impact, cut power to the system using the relay operation

Overshoot of maximum 5cm

- Program to keep track of electromagnet activation time and notify upon surpassing active time

A project team initiative is to develop a robust tuning program in the Simulink interphase to be able to tune the PID values without having to re-run or re-compile the program.

## 3 System Level Design

The pick and place arm project has three distinct systems to be integrated: Mechanical system, Electrical system and Control system. These systems are interconnected as required by the system in order to provide power and control signals to the mechanical system, in order to pick and place the load from the target position to the final position.

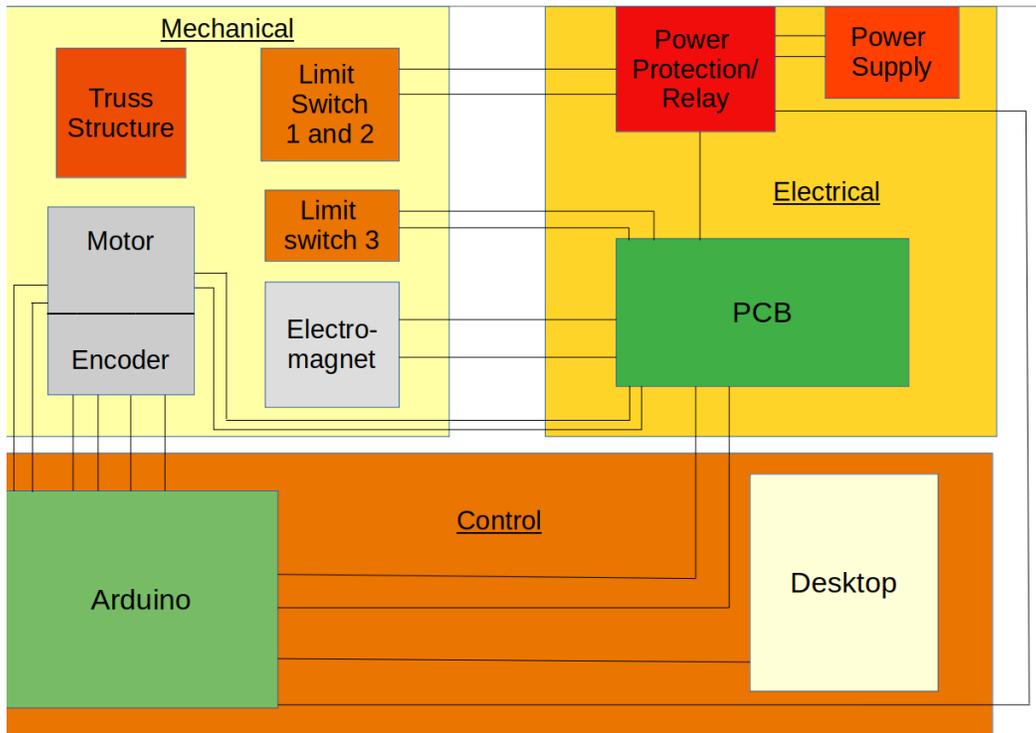


Figure 1: System-level block diagram

As observed in the system level block diagram above, the different subsystems are interconnected by functionality. From the power supply, we use the 12VDC supply which is connected through the relay (for power control/protection) to the PCB as the power input. The power protection relay is also interconnected with the two limit switches (1,2) which are connected on the test board. Using the NO an COM connection of both limit switch 1 and 2, we connect to the relay positive and negative terminals which on contact with close the switch in order to switch the relay from on to off. Limit switch 3 on the board which is actuated on the puck dropping into the target destination will be powered from the the PCB board through the pull down resistor and connected to the the Arduino controller to sense when the run time clock is to stop.

From the power supply we also utilize a secondary 5V output which is also connected through the relay at COM3 which is then used for Arduino input D44 as a means to recognize what state the relay control is at in order to use the switching of the relay buttons to reset the control system without having to restart the compilation.

From the PCB which encloses the motor drive circuit, we output to the DC servo motor the output pin 3 and 5 from the full H bridge motor driver (Toshiba TB6568KQ). PWM pin 1,2 input to the motor drive circuit is received from the Arduino board pins D11,D12. Similarly for the electromagnet drive circuit we output the power pins for the physical electromagnet attached to the end of the truss arm structure using the pins on the PCB board, and the control 5V PWM signal is from the Arduino board as an input.

The brass truss structure is mechanically secured to the motor shaft by means of a aluminum collar and set screw. Therefore acting as a single physical object on rotation. The encoder is also attached to the motor however it has its own power and communication channel wiring. Power (5V and GND) and communication terminal wiring (encoder ch.A,B) is connected directly from the Arduino board.

The Arduino micro controller is programmed and controlled using a desktop system which runs Matlab/Simulink software where Simulink/Arduino inter-phasing packages were installed for configuration. A USB connection from the desktop provides data and power to the Arduino.

### 3.1 Demonstration Apparatus Setup

The final system demonstration apparatus shown in figure ?? indicates the demonstration bed with the mechanical truss arm mounted onto the DC motor shaft via a aluminum collar which is fastened using a set screw. On the test bed is seen limit switches 1,2 which are 5 cm buffer positioned away from the 180 degree rotation which will only be impacted if the overshoot is too high. The target position hole can be observed which right underneath exists limit switch 3. The Arduino micro controller, desktop, PCB, bread board, power protection relay and power supply are all required components in the system and are arranged to the wiring side of the physical test bed.

The demonstration arm will begin 90 degrees clock wise to the final target position, swing 180 degrees to pick up the load (metal piece) using the electromagnet, and then swing 90 degrees back to drop the load into the target position in which limit switch 3 will be activated and the run time of the arm will be recorded. Limit switches 1,2 will act to shut power off to the mechanical system and electrical system via interconnection to the relay for power protection and safety.

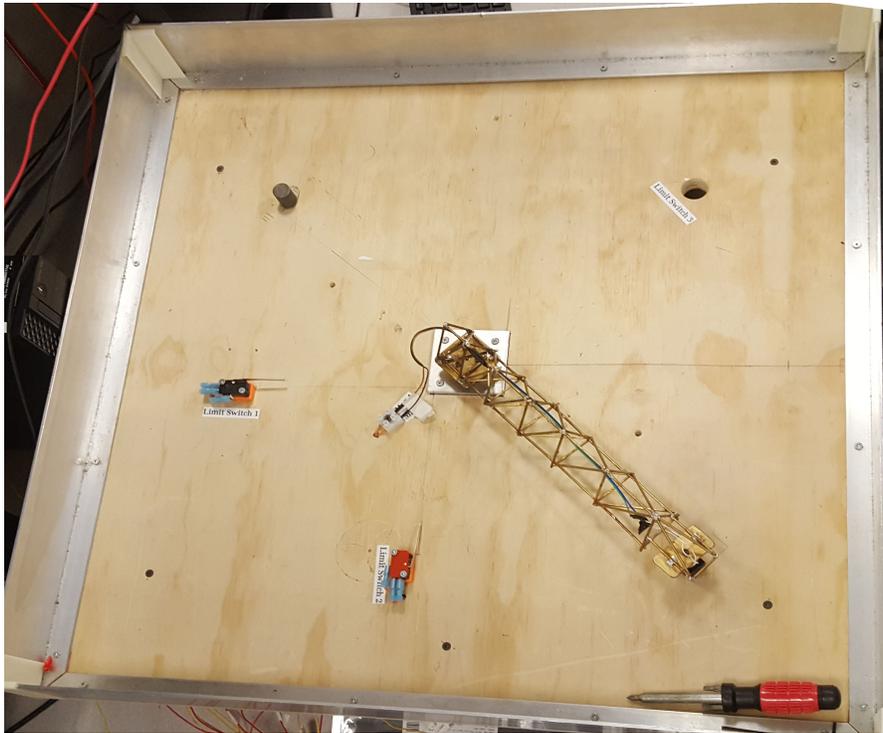


Figure 2: System Demonstration Apparatus

## 4 Mechanical Design

The mechanical design of the pick and place arm consists of a Truss structure made from brass rods. As per the requirements of the system the mechanical arm is to span 30 cm, with each truss structure

inclusive of at least 1 redundant member. Further the design of the arm was limited to a maximum width of 10 cm and max height of 5 cm, and no brass rod could be longer than 8 cm. The major concept selection parameters revolved around minimizing mass moment of inertia ( $J$ ), centre of mass close to motor shaft axis, minimal deflection under FEA loads, and ease of construction.

Figure 3 shows the mechanical arm concept design that was selected after analyzing various aspects of the design requirements with other designs. This concept rational minimizing the usage of rods by developing a triangular profile, and the spacing between the rods expanding outwards from the rotation axis lies closer to the motor axis in attempt to reduce moment of inertia ( $J$ ) and the centre of mass. Refer to the mechanical report for the decision matrix which was used to evaluate concepts. Upon selection of the concept truss structure, buckling analysis and a further rigorous FEA analysis were performed.

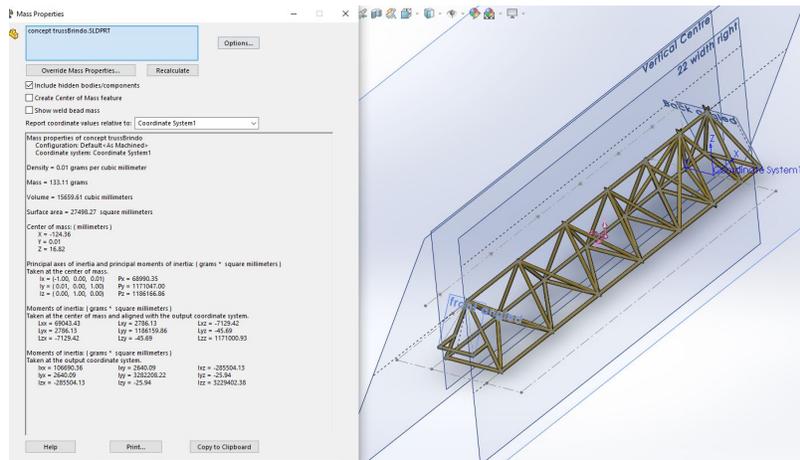


Figure 3: Design and mechanical properties

No major alterations to the mechanical brass truss arm were made. On testing the integration on the test apparatus, one of the simulations under over gain conditions resulted in the truss arm accelerating into instability and crashing into the limit switch flap. Although the limit switch did cut the power to the system, the truss structure was damaged and fell off the main connect brass plate. The arm was re-soldered onto the brass plate in relatively the same position. Since we were still tuning the PID control minor shifts in position (if any) did not alter the performance of the system or accuracy.

## 5 Electronics Design

The electronics design for the pick and place arm consists of a motor drive circuit, electromagnet drive circuit and power protection circuit which is also interconnected with test apparatus limit switches for appropriate power shut off on contact. The design of motor drive circuit revolves around the TB6568KQ full bridge DC motor-driver IC that is used to control voltage across the motor output from the electric circuit board via PWM input signals which are from the Arduino micro controller. The motor drive circuit utilizes 12V voltage for operating the DC servo motor. The control PWM input are 5V and are duty cycle controls through the Matlab algorithm. For appropriate operation of the TB6568KQ IC chip a heat sink was designed and added to the chip in order to dissipate energy.

The provided electromagnet (EM) is active low in operation and we use PWM control for operation. The electromagnet drive circuit is designed to use a BJT semiconductor to amplify the 5V PWM signal to a 12V PWM signal. To turn the EM off, we apply a 5VDC signal to the base of the BJT, and when the signal is off or minimal the EM will be on. The electromagnet has the highest force over a greater distance when the 5% signal.

The power protection circuit uses a relay switch in order to display the power state of the circuit (energized or not), allow the user to emergency shut off the power, and enable the limit switches on the test board to turn off the power on contact.

Since the H-Bridge IC was continuously burning out while tuning the system the soldered H bridge was better suited to be placed separately for easy removal and replacement. Therefore the motor drive circuit was separated from the circuit board and is connected through a breadboard in the same logic as initially designed. Another change in the electronic setup up was the use of the switching state of the power circuitry relay to control and reset the system. This a 5V connection to the comm 3 port of the relay switch was made and connected with D44 of the Arduino in order to use as a switching state variable. No further alterations were made to the electrical system.

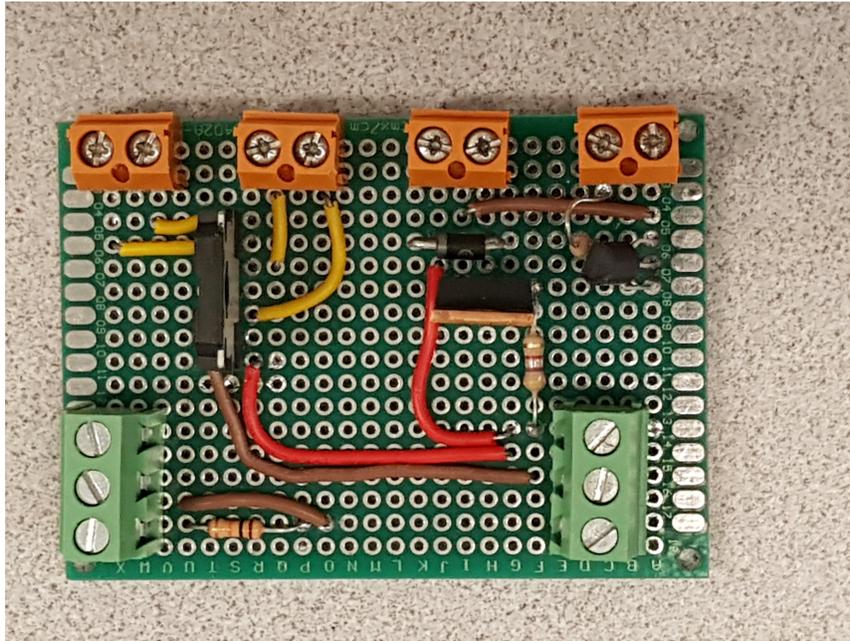


Figure 4: Electrical Circuit Board

The figure 4 shows our original configuration of the circuit board. Dedicated ports on the PCB allow for set screw connections for the circuitry as required.

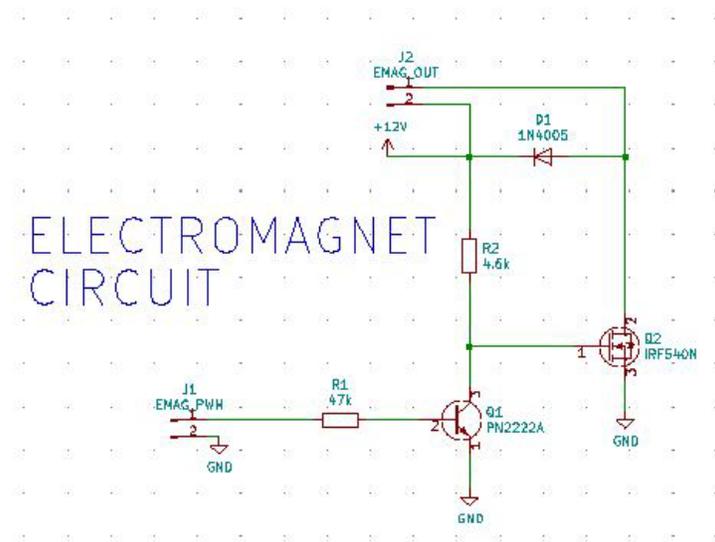


Figure 5: Electromagnet Control Circuit

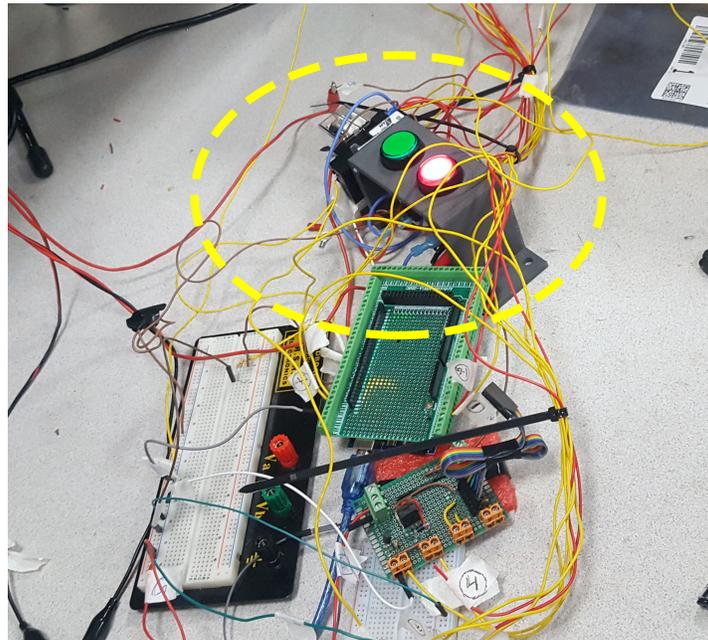


Figure 6: Power protection relay switch setup feeding system

## 6 Controller Design

In the control design of the project, we use Simulink to design a controller to smoothly move the truss for accurate rotation. We design our transfer function based on our electrical and mechanical model. Figure 7 represents our control system where  $M(s)$  is our transfer function and  $G(s)$  is the controller that we need to design.

Variable	Kp	Ki	Kd
Value	27.5	3	6

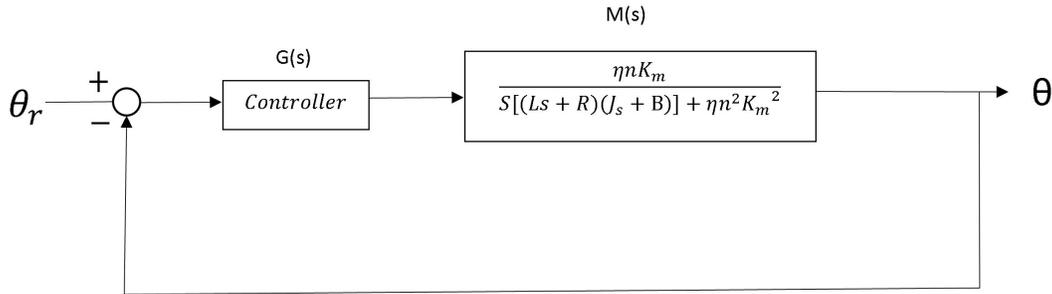


Figure 7: Closed loop signal flow diagram

After testing and running different types of controller, we decided to use PID controller to get the optimum controller. We calculate the root locus and use the Matlab PID tuner to get reasonable values for the PID constants. We simulate P, PI, PD and finally the PID controller to try different controllers. Finally, we check the step response, overshoot%, settling time, gain and phase margin to conclude our final controller.

In practice, we had to use two set of Kp, Ki and Kd. One set of PID values were for moving from 0 to 180 and another set of PID values were for moving from 180 to 90. This decoupling of PID values is a consequence of how the J of our system changes. When we are moving from 0 to 180, the J of the system is equal to the truss only. When we are moving from 180 to 90, the weight of the payload is added to the inertia of the system.

When the system is off and the arduino is on, the integral continuously adds up error. This is a consequence of noise and how we programmed our simulink. As such, our system is now dependant on the time it takes for us to press the start button on simulink. Furthermore, any integral component for moving from 0 to 180 is carried over to the component when we are moving from 180 to 90. In order to ensure consistent results for the same input, we had to program a reset for the I portion of the PID controller. When this reset was triggered, the integral portion of the PID controller was reset to 0. The integral portion of the PID controller had to be reset anytime the input changed (Certain state changes 7.5) or the system was power cycled (Section 7.1).

## 6.1 Controller PID values on the Arduino

Evaluating system performance with the theoretical PID values which were calculated in the controls report analysis of the system, resulted in almost to very little movement in the system. This can most notably be reasoned due to the implementation of a digital control system via the Arduino. When analyzing the system in the controls section of this report we analyzed a continuous time system and maintained that in the s-domain, however the system behaves different in the digital (discrete domain) and the z-domain must be used in evaluation and appropriate digital control.

Attempting to employ Ziegler-Nichols PID tuning method to define the PID values required the ability to attain the ultimate gain of the system in which the system is on the verge of instability

and is under perfect oscillatory condition. However in attempt to find the ultimate gain ( $K_u$ ) we often burnt out H-bridge devices most commonly related to over current flow. Therefore to define our system PID values for the 180 and 90 return to the target load drop position, we first employ a PD controller to find a system performance with reliably output performance and then adjust the steady state error of the system using a minimal integral value to form a PID system, which is similar to the approach taken when developing a PID controller in the controls report.

## 7 Software Interface

### 7.1 Power state Detection

Our code needs to know when the system is moved from off to on state (power protection LED switches from RED to GREEN). In order to detect the off to on state switch, we connected a 5V cable from the power supply to the COM3 of the relay. Then, we passed a cable from the NO3 of the relay to the D8 pin of the arduino. D8 pin was connected to ground through a pull down resistor in order to stop it from being floating. This state detection is critical for the integral blocks in the system. This architecture allows us to wire the limit switches to electrically change the state of the relay. As such, the arduino only needs to read the state of the relay rather than having wires and code keeping track of all the limit switches.

Throughout the rest of the section, it is discussed how the simulink model's knowledge of the power state allows us to continuously run the same code while tuning.

### 7.2 Input Control

The input into the system is determined by the state of the finite state machine. That finite state machine is discussed later. Here, we discuss how we are generating the ramping feature of 0 to 180 and 180 to 90.

We used a ramping function because it provides a good balance between tune-ability (giving us another variable to control) but does not dramatically confuse us. For example, if we used a polynomial, we would have to tune the coefficients of the polynomial and tune the values of the PID controller. We found that using a ramp function while only tuning the values of the PID controller provided us with a very good system response.

The simulink model for moving from 0 degrees to 180 degrees is shown in figure 8. The important portion of this system is the integrator. As the constant (1) gets a higher user tune-able gain (4 in figure 8), the system ramps up to 180 degrees faster. Specifically, the system reaches 180 degrees at  $t = \frac{1}{Gain}$ . At  $t = \frac{1}{Gain}$ , the system saturates with the saturation block that we placed after the integrator. In the saturation block:

- max = 1
- min = 0

When the system's power state is power cycled as discussed in Section 7.1, the integrator in figure 8 gets a rising edge and is consequently reset. Additionally, a power cycle resets the state of the finite

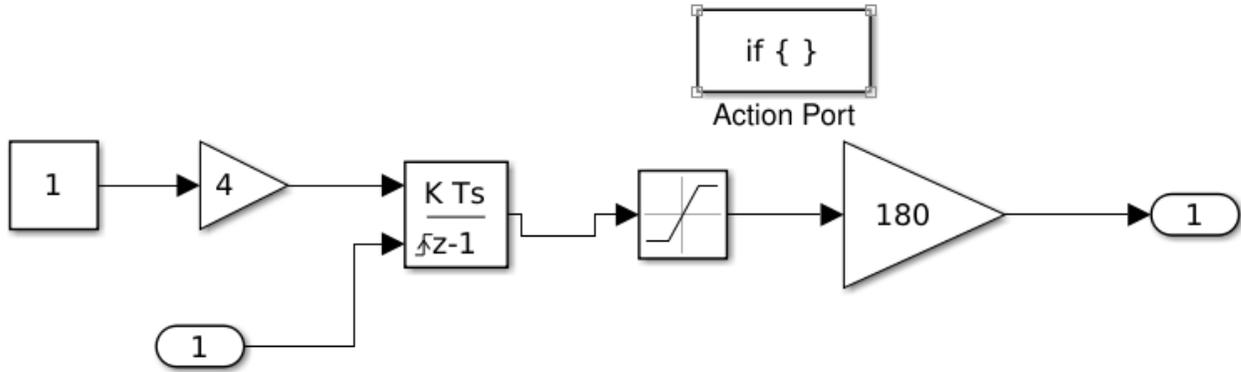


Figure 8: Simulink model for outputting ramp for 180 degrees

state machine is set to 0. As such, when the user starts the system again, the system will try to go to 180.

If the system reaches 180 degrees, has the electromagnet, the finite state machine will reach state 2. At state 2, we look to ramp down to 90 degrees. In this state, the simulink model shown in figure 9 is activated and the simulink model ramping to 180 is deactivated.

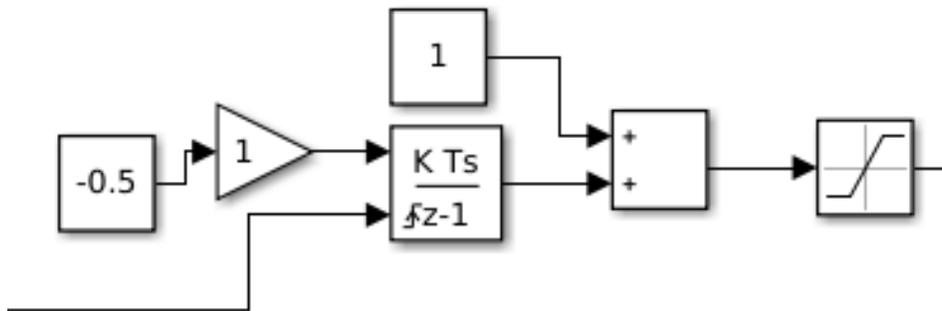


Figure 9: Simulink model for outputting ramp for 90 degrees

The logic here is similar to the ramp to 180 discussed before. However, in this case, the integrator decreases the value from 1 to 0.5. The saturation block for this system is set as follows:

In the saturation block:

- max = 1
- min = 0.5

The integrator in this case is reset when system switches from state 1 to state 2. A separate trivial simulink module was developed to create a rising edge for this state change.

### 7.3 Motor Control

In order to interface the arduino with the H-Bridge, we need to convert the voltage output of our PID controller into a PWM signal. The voltage to PWM converting simulink model is shown in figure 10.

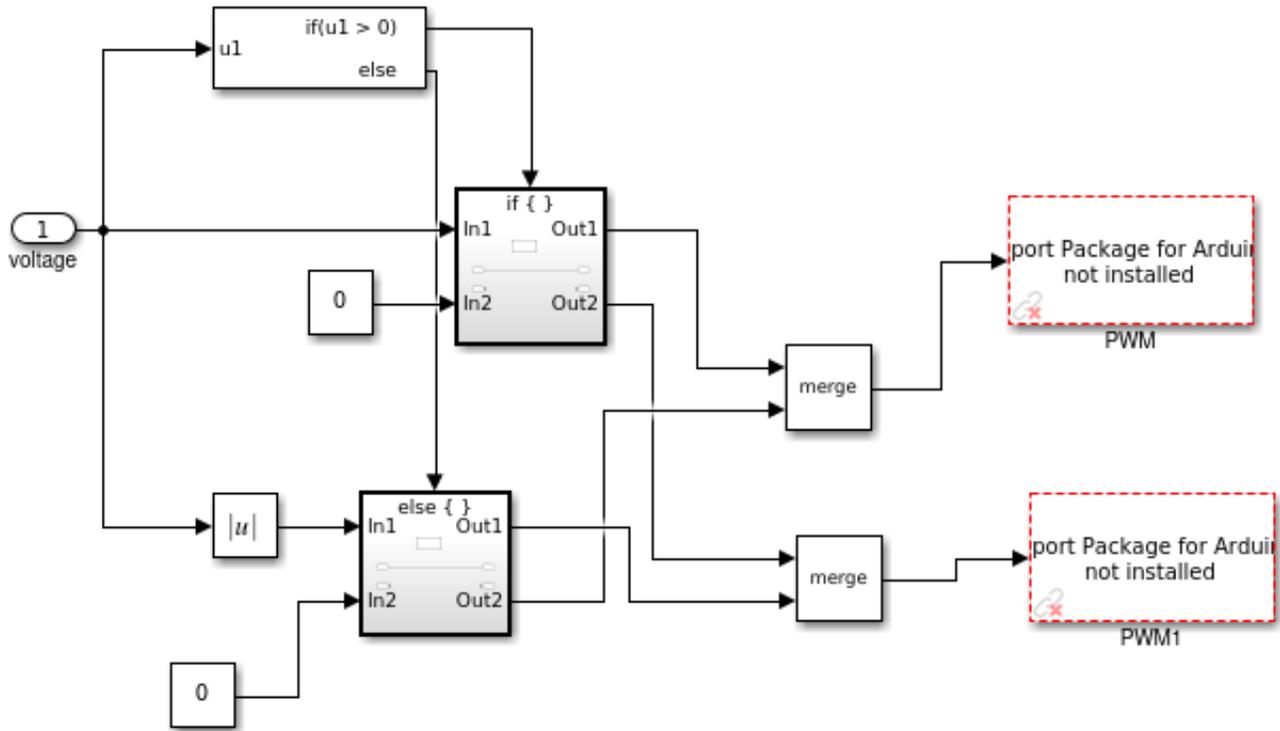


Figure 10: Voltage out of PID to PWM signal converting simulink model

If the voltage from the PID controller  $> 0$ , the PID controller wants the motor to CW. As such, PWM on the arduino gets the voltage the PID controller wants. If the voltage from the PID controller is  $\leq 0$ , the PID controller wants the motor to move CCW. As such, we flip the polarity of the average voltage being provided via the PWM to the motor.

### 7.4 Motor Encoder reading

The encoder reading simulink model is shown in figure 11.

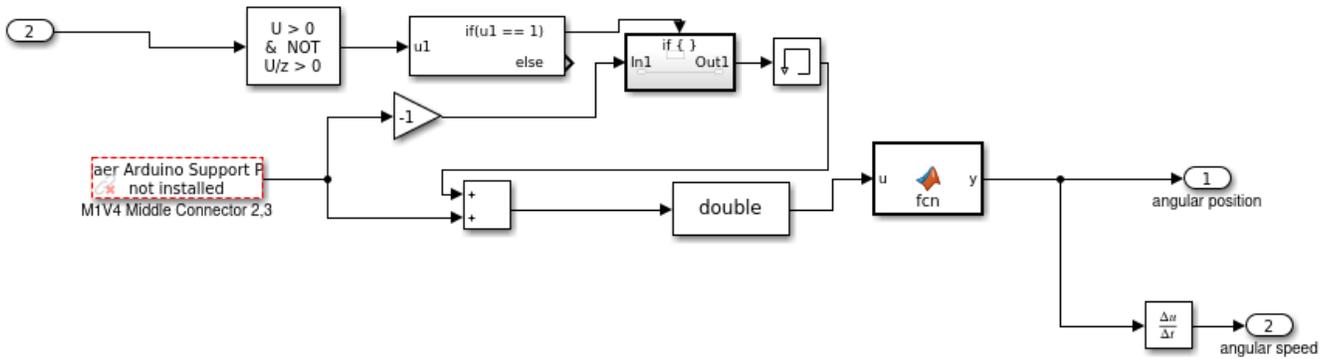


Figure 11: Simulink model for encoder reading

There is a slight modification to the simulink model for encoder reading that was recommended in lab. Rather than just outputting the encoder reading, the encoder reading is zeroed everytime there is a power cycle. This change is important because we noticed that the encoder did not report 0 when we reset in between runs. We do this zeroing by detecting when there is a power cycle (a rising edge). When there is a rising edge, the value of the encoder is stored in memory. While the system is running, this off zero error is subtracted from the current value the encoder is reporting.

All of this allows us to be more precise and removes the systematic error from the encoder between runs. We were able to be more consistent in our runs when we introduced this zeroing at the beginning of a power cycle.

The fcn converts the double to a angular position. It is important to note that throughout this report, we use degrees rather than radians because thats what we wanted the user to interface with. It's much easier for the user to see 60 degrees rather than 1.05 rad and recognize whether the system is performing as expected. The PID controller and encoder operates using radians and gains are used to convert the UI to degrees for the user.

The code for fcn is:

```
function y = fcn(u)
pos = (u *360)/(4*500*6.3);
y = pos *pi/180;
```

The equation for pos was calculated as follows

$$position = \frac{EdgeCount * 360}{xNg}$$

N = 500 - Number of pulses generated by encoder per shaft rotation

x = 4 - encoding type (4 for quadrature)

g = 6.3 - motor gear ratio

## 7.5 State

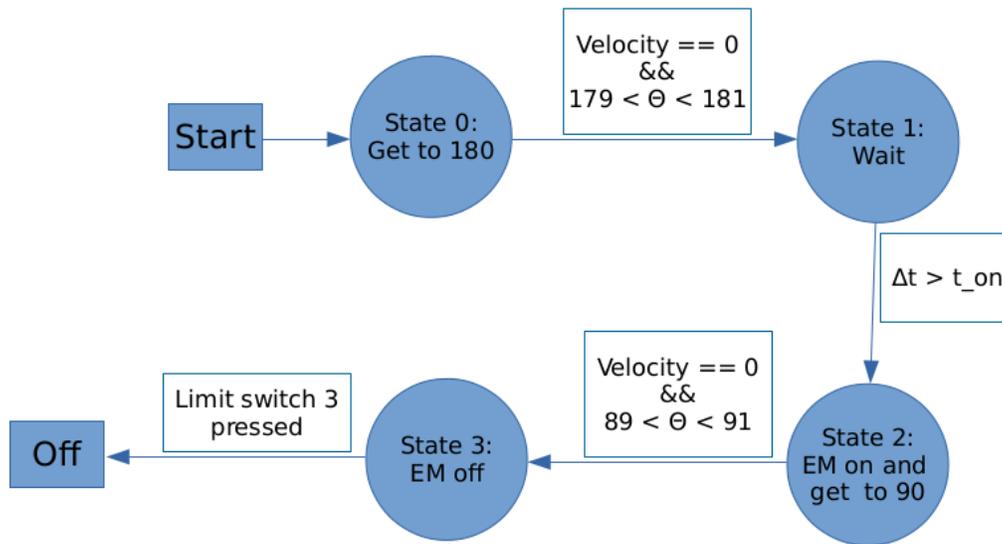


Figure 12: State diagram

### State 0

#### State Requirement

State 0 is the default state. The system has just been turned on

#### Input

In order to generate the input signal, we wanted to use a ramp signal. However, this ramp signal had to be able to be reset according to the system state. As such, we used an integrator to generate our ramp signal. When the system is turned on after being off, we get a rising edge on pin 8. The integrator's rising edge reset is connected to the D8 pin.

#### Output

The PID controller for the system has different values for when we are going to 180 or going to 90 degrees. This decoupling of the PID controller is due to the fact that the inertia of our system ( $J$ ) is different when we are moving with the payload attached (180 to 90) than when we are moving without payload (0 to 180). Since our system has two different inertias, we looked to program two different values for the PID. The PID values of the 0 to 180 movement are listed here:

The electromagnet is turned off for this state

### State 1

#### State Requirement

$$179 < \Theta < 181$$

To allow for error. With this error, we are able to ensure the truss will be over the payload.

$$v = 0$$

The system requirements require that we have  $v = 0$  when we turn on our electromagnet.

### **Input**

Input = 180 degrees. We want the motor to be steady at 180 degrees when picking up the electromagnet

### **Output**

If the system is in state 1, the truss is over the payload and the truss has stopped. As such, it is time to pick up the payload. Electromagnet is turned on.

## **State 2**

### **State Requirement**

$$\Delta t > T_{on}$$

Electromagnets are known to have slow actuation time. As such, we wait  $T_{on}$  for the electromagnet to turn on and then input movement to go to 90. In code, this waiting is started we enter state 1 (the electromagnet is over payload and turned on):

1. When we enter state 1, save the current time into memory
2. Continuously (check if current time) - (time when we enter state 1)  $\geq T_{on}$ .
3. When (check if current time) - (time when we enter state 1)  $\geq T_{on}$  is true, enter state 2.

### **Input**

Ramp down to 90 degrees. PID values for moving from 180 to 90 degrees:

### **Output**

Electromagnet on to hold payload.

## **State 3**

### **State Requirement**

$$89 < \Theta < 91$$

To allow for error. With this error, we are able to ensure the payload will be over the hole.

$$v = 0$$

The system requirements require that we have  $v = 0$  when we turn off our electromagnet.

### **Input**

90 degrees.

### **Output**

Electromagnet off. The payload is at its final location and should be dropped.

In order to simplify the number of connections to the arduino, we simply just had the limit switch 3 flip the relay's state to off as discussed in section 9.3. As such, we were able to integrate the time that the system was on and just print the runtime to the screen.



for the electrical characteristics of the DC motor that the system has slight behavioral changes in the CCW vs. CW rotation, therefore is it critical to implement two separate PID values for each motion interval. The final PID values implemented are listed in the table below:

-	$K_p$	$K_i$	$K_d$
180°	250	0	58.5
90°	150	0	60

## 8.2 System Performance

The performance results of the PID tuned pick and place arm system had a total run time of **2.03 seconds** on average. This run time value is the time when the relay button activates power to the system, and limit switch 3 cuts power to the system when the metal puck is dropped in target position.

Presented in figure 14 we observe the requested position (target value) of the system and the actual trajectory of the system due to PID control read through the encoder value.

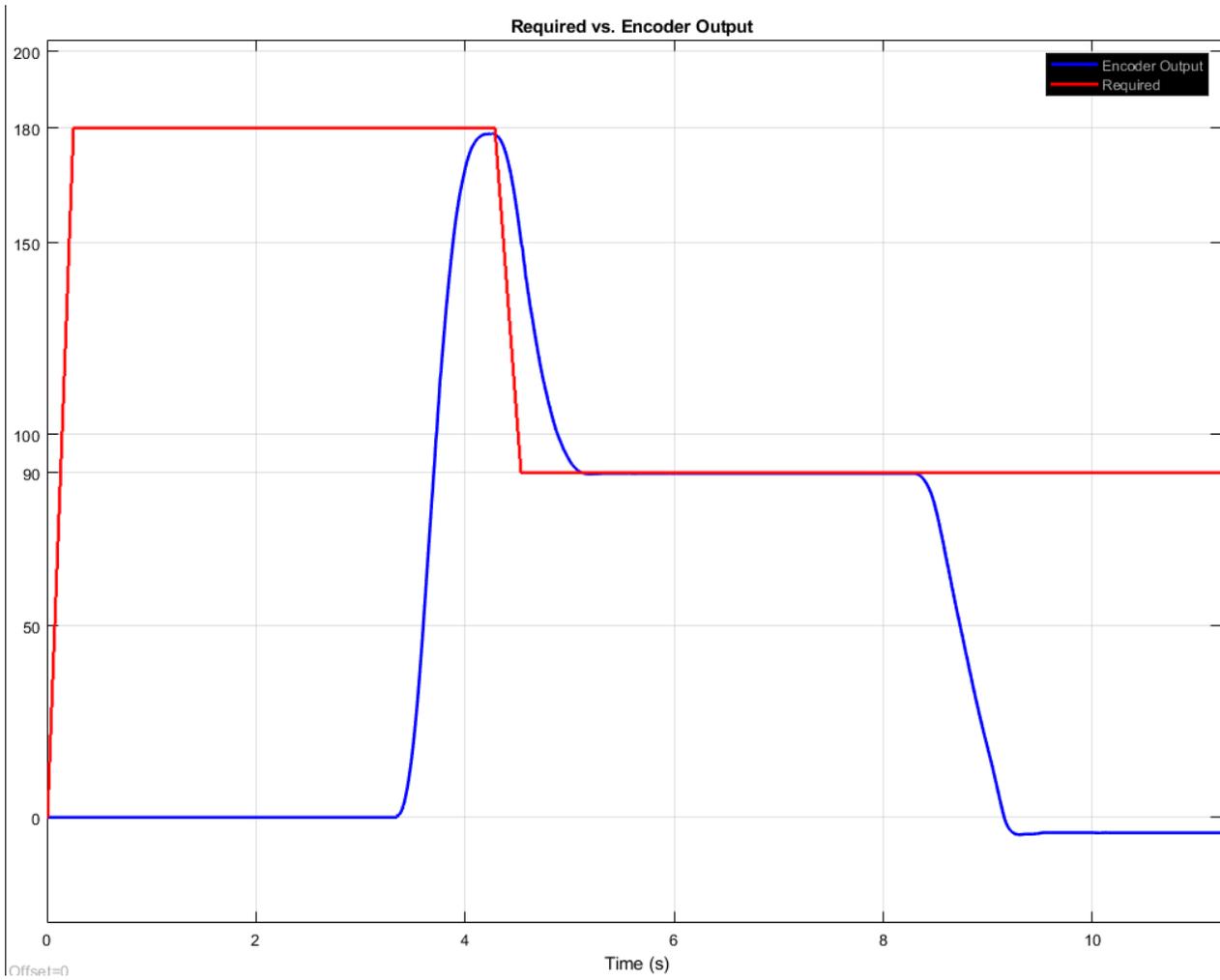


Figure 14: Simulink scope plot for required position vs. encoder value capture

It may be noted the system in the run shown has no overshoot in the system and is a over damped response curve however the acceleration of the actual system to the target position is quiet fast. Due to the logic of our system, the target value of set to  $180^\circ$  for the duration until the system is connected to power and the motor is allowed to run. The target value is shown to quickly change from  $180^\circ$  to  $90^\circ$  upon the system arriving to the position within the threshold range and critically a velocity of zero, which is a requirement before picking up the electromagnet.

It is observable the required (target) value when changing between  $180^\circ$  to  $90^\circ$  has a slight slope in the line instead of being vertical. This is due to the electromagnet pickup and wait time which we have set in the algorithm to have a visible stop when picking up the magnet. If the time for the hold value was decreased the transition from  $180^\circ$  to  $90^\circ$  would have less slope and change quicker.

Return output of the system shows the ramping of the system to a strong acceleration due to the high gain in the system, and damping to the final position value. The output of the system is very accurate to reach both  $180^\circ$  and  $90^\circ$  target values.

### 8.3 Velocity and Position Plot

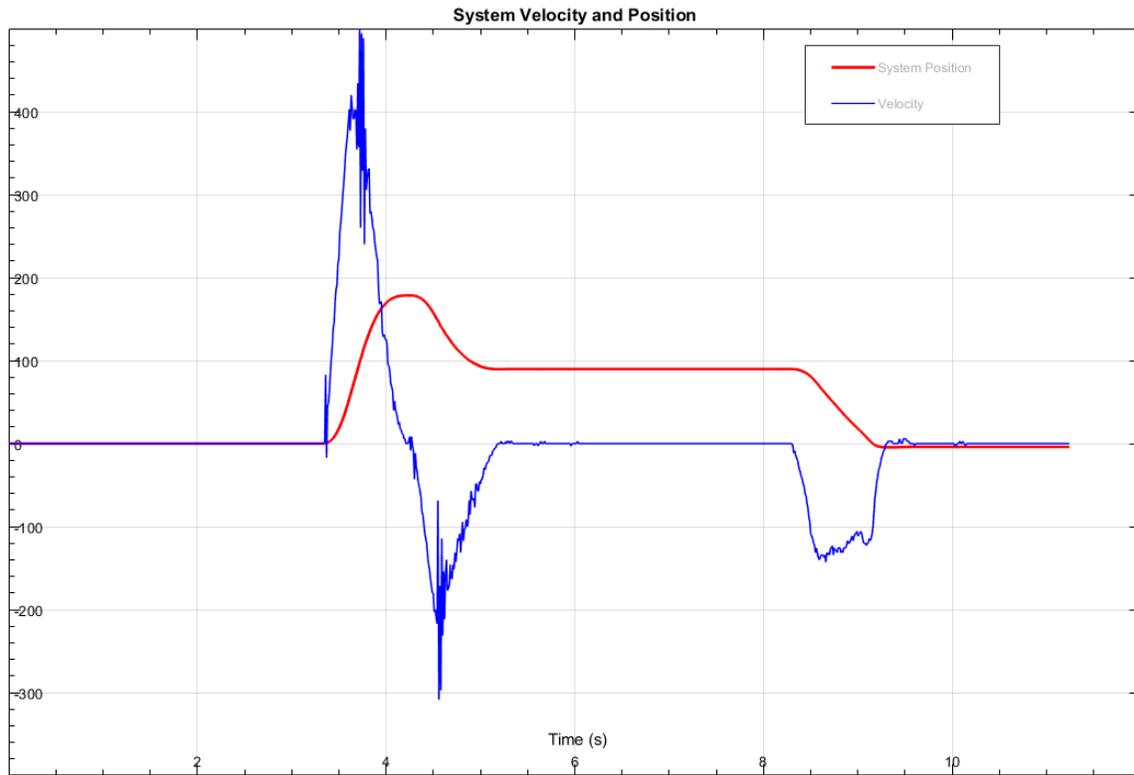


Figure 15: Simulink scope plot for system velocity and encoder position

Observed in figure 15 above the encoder position plotted against the velocity of the system. The critical point to observe is the position of the velocity when slowing down on the position side of the plot, until the 0 is crossed. Upon zero contact there is a slight pause (for electromagnet pick up and hold fractional time) until the system accelerates in the negative direction back to the target position. The switch of the encoder plot in target position to  $90^\circ$  matches the point at when the velocity begins to move in the negative direction (ie. clockwise to the target position).

## 8.4 Electromagnet On Time plot

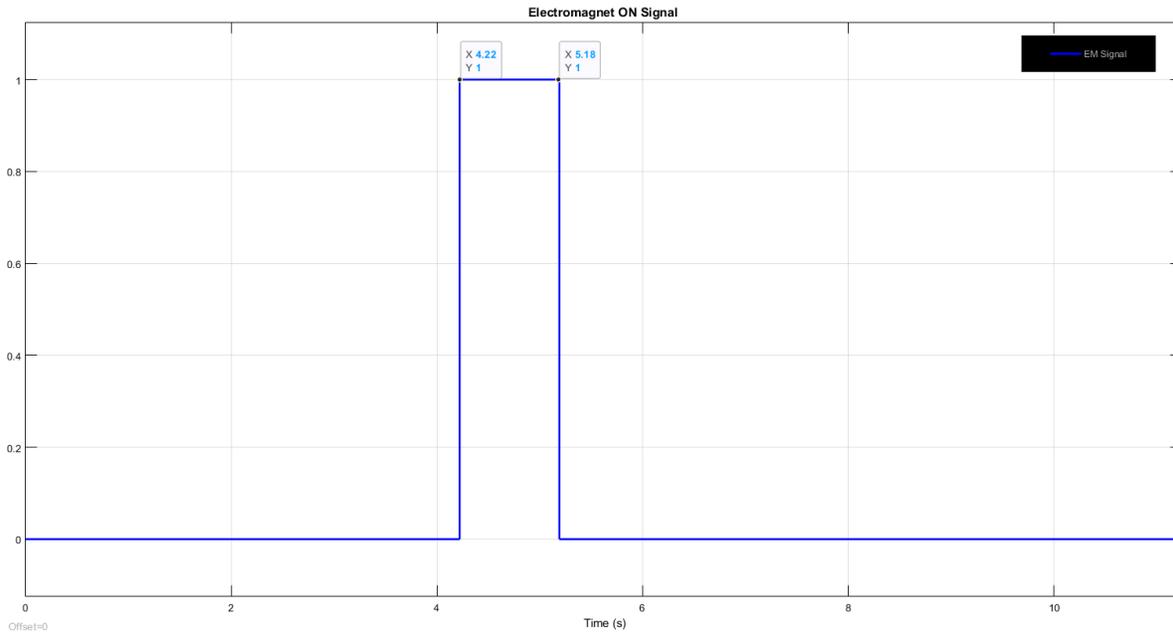


Figure 16: Simulink scope plot for electromagnet control signal vs. time

Figure 16 illustrates the control signal used for the electromagnet. The magnet control signal is ON (1) when the system reaches the pickup metal pickup location at  $180^\circ$  and continues to stay on until the system reaches the target position and reaches a velocity of 0 (stop). The data cursors in the plot show the X-axis time values at 5.18 and 4.22 seconds therefore the total on-time of the electromagnet is 0.96 seconds. This time also indicates the total return time of the run including the fractional electromagnet pickup and hold time.

## 8.5 Demonstration Video

A link to the performance demonstration video may be found at the link below:

<https://youtu.be/QqD11hy9zQI>

# 9 Observations and Discussion

## 9.1 Test Reliability and selection of PID values

In having one of the target requirements of the project to complete the pick and place operation in the fastest time possible, there are a variety of variables which can affect the total run time of the system however during PID tuning it was observed that a fine balance is required to be met: to have a reliable system and adequately high speed of operation. Initially attempting to use Ziegler-Nichols tuning method and later realizing to reach the ultimate gain of the system may inflict more

damage to the brass rod structure, we moved to the tuning method we used in the controls portion of this system. By tuning a proportional-derivative (PD) controller until a system response of desired attributes is acquired and then adding marginal  $I$  into the system to form the PID controller required to adjust the non-linear properties.

Since the system is over-damped and very close to critically damped (theoretical) response we saw that the systems performance had greater reliability then overshooting the system. When testing the system with PID values with over shoot, the system took longer time to release the current through the motor spinning in one direction to re-charge the inductive motor to spin the alternative way. We observed when there was more fluctuation, this process added more time to the system, therefore the optimal choice was to damp the system adding  $K_d$  until the system was over-damped. This was important to eliminate the overshoot of the system to eliminate wasted time reaching the target position.

In terms of system reliability, a similar characteristic was observed. When overshooting the system, there is varying momentum in the system and the damping can quiet often overshoot by  $2^\circ$  to  $3^\circ$ . System performance was more reliable with higher damping and continually spinning in one direction not to adjust the error if any, therefore we tuning the values such that the damping was just perfect to perform the  $180^\circ$  rotation.

Reliability also came from using the same system setup while performing and tuning the system. Different system performance was observed when not using the same test bed setup because of possibly mechanical or electrical build qualities of the encoder and motors. Therefore is was essential that when tuning the system we tune the same whole system consistently. The lab power supply were current limited to 3A. Therefore it was critical to test the system at the same current limited position to coordinate the PID values. Variability in lab power supply with the current limit position changed the performance characteristics, so when changing lab stations, checking the current limit position was important.

## 9.2 Discussion on system performance

The pick and place arm was able to complete the task in a total of 2.03 seconds (on average). The total time of the system was an accumulation of the following actions:

- Relay switch button switch on
- DC servo motor ramping up to  $180^\circ$  metal puck pickup
- Electromagnet energize - hold and wait time (fractional time set in code)
- DC servo motor return to  $90^\circ$  target drop position and zero velocity
- Electromagnet de-energize - metal puck drop time to limit switch 3 impact

The system has certain variables outside of the PID tuning of the system which adds to the total run time of the system, in particular the time needed to charge the electromagnet (inductive element) when reaching the  $180^\circ$  position and having a velocity of zero (full stop). A fractional wait time was thus implemented to allow the electromagnet to pick up the system which may possibly be shorter if the induction in the system was partially charged to a value which does not pick up the magnet. Further a few milliseconds are accumulated to the overall time by the time it takes for the magnet to impact the limit switch at the target position.

Our system performance has no overshoot as discussed in the selection of PID method above and figure 14 indicates actual system performance via the encoder values of the motor of the system. This meets the requirement of the project to have a max overshoot of 5cm. This also resulted in the limit switches 1 and 2 never coming close to being impacted however limit switches 1 and 2 are operational and have been tested to protect the system by turning off power on contact (switch).

Having a performance characteristic curve with no overshoot is critical for a pick and place system because we do not just want to reach the target value and hold that position, but we want a quick return in the opposite direction to another position. Attaining close to critical damp performance on the over damped side of the system response allowed the motor to behave in a natural manner and current flow was in one direction for one interval of travel which through observation had better performance characteristics then the motor having to change direction to reach a target position. Our system was successfully able pick up and hold the metal puck by using the PWM signal to control the electromagnet activity. Implementing the 5% duty cycle to control the system electromagnet ON time was observed to have the highest magnetic pull force which allowed the 2mm gap between the top of the metal puck and the electromagnet face be sufficient. The duty cycle is switched to 100% when the target position is reach and velocity is zero, and drops the puck on limit switch three. To have efficient coding strategy, limit switch 3 operated to switch the state of the relay (which in this scenario) cuts power to the system, which is read into the Arduino at port D8 to use as the point to stop the system run timer. Another timer was working to make sure the electromagnet had a max ON time of five seconds which operated to release the magnet operation when that value surpasses the threshold but was not interconnected with the rest of the systems operation.

### **9.3 Power protection, limit switch operation and system relay state**

Protection to the system is provided through the overshoot limit switches and emergency stop via the relay button. Using the logic of the relay, we use the wiring configuration to have a variable in the Simulink model to know the status of the switch. The system relay state is used in order to start and stop the run timer, as well as restart the simulation (resetting encoder values to zero and the state value of the finite state machine).

### **9.4 Testing the Ramp and Step Input variability**

Implementing the ramping function allowed tunability to the signal to the system output however we note that the value of the ramp duration itself effects the output of the system. Initially setting the speed to 1 second, significantly slowed down the system even when increasing gain to the very high value. That being known before tuning the system a test for the value of the ramping speed was tested. The system ramping speed was changed to ramping speed of 1/4 seconds for 180 degrees movement. We observed testing at near step input (very step ramp up), had poor interphase with the system, where as a ramp of 1 second had too slow of control of the rotation needed. The same value was used in the rotation back for 90 degrees of the target position.

## 10 Further Work - Additional Improvements

Designing and building a working pick and place arm with the constraints of each sub system has allowed us to learn on how the system is affected by numerous properties which will allow us to improve performance.

### Mechanical Improvements

Simulating and further testing the pick and place arm, we observed the affect of moment of inertia ( $J$ ) has on the performance of the system. In minimizing the inertia, we will be able to have a faster angular acceleration therefore speeding up the system response.

Although, a mechanical constraint was having a maximum length of 8 cm in length of the brass rods, which forced the truss structures to have multiple joints and greater member link count, decreasing the amount of joints by having longer rods would decrease the overall weight of the system and allow for better structural build quality.

Referring to the mechanical report (attached in the Appendix (13)) the system had a 4 cm vertical deflection under 1 kg weight load. Although the metal puck was 50g and the system performed well, the manufacturing quality could have been better at the joints which was difficult to join. Improvements of welds can be made by developing a jig setup to align rods at set angles.

### Electrical Improvements

After the electrical demonstration and proceeding with integration components, we ran through many H-bridge motor drives while tuning the system to be as fast as possible and understanding the characteristics of the system. Since soldering and de-soldering onto the circuit board was tedious and repetitive, a smaller breadboard with isolated power and control wiring was used to wire the H bridge. After finalizing the system, a finished circuit board could have been implemented to have less components in the final demonstration.

The terminal connections which were not soldered onto the board could have been improved. It takes 30 minutes to wire the system as the setup board is shared with other teams and these connections are made using set screw and raw wire connections. If proper connectors were implemented, it would be less time consuming and more reliable.

### Controls and Integration Improvements

The current system has a manual switch to reset and control pick and place operation however this system can be improved by implementing a return to 0 position automatically operation after the system has dropped the mechanical puck at the target position.

The system performance was quite reliable however the speed of the system does have potential for improvement. Adjustment of the PID values can make the system perform better depending on the system root locus and current values. Further the system implemented has a max time ON shutoff feature for the magnet which (based on the state of the system) turns off the electromagnet for safety consideration. The control system could be improved by having the GUI of the system

throw an error to the user that this event has occurred and the system should reset itself to initial position.

Arduino micro-controller was programmed using Simulink but it can alternatively also be programmed using a python code. An issue that teams were often facing was the run time error which was created by the Simulink which often crashed the program which may have been avoided if a different platform was used.

## **11 Team member contribution**

All team members equally contributed to all aspects of the term project. This was collectively done as a group for learning purposes as all members wanted to contribute and know how each element of this project came together and helped reinforce decisions which were required to be made while completing tasks.

## **12 Conclusion**

In MSE312, we were given the task to create a pick and place robot. By consistently evaluating each individual part of the system (mechanical and electrical), we were able to create a robust pick and place robot. When it came to the integration portion of the project, we could be certain our individual subsystems worked and moved onto creating a safe program. Using a finite state machine, we were able to easily sequence the steps and inputs needed for the pick and place robot. Finally, using the theory we learning in our discrete controls course, we were able tune the PID controlled. This led to a robust system that could be repeated various times.

# 13 Appendix

## 13.1 Mechanical Report

MSE 312  
Mechanical Report

Anjandev Momi - 301281186  
Brindan Ramalingam - 301271567  
Neeraj Bansal - 301323495

August 2, 2019

# Contents

1	Introduction . . . . .	3
2	Design Approach . . . . .	3
	2.1 Design Constraints . . . . .	3
	2.2 Design Objectives . . . . .	3
	2.3 Task Distribution Among Team Members . . . . .	4
3	Quantitative Analysis of 3 Truss Structures and Design After Final Iteration . . . . .	4
	3.1 FEA Summary of 3 Designs . . . . .	4
	3.2 Mass of System, Center of Gravity, and Inertia (J) . . . . .	5
	3.3 Buckling Analysis . . . . .	5
4	Calculation Assumptions and Effect on Results . . . . .	5
	4.1 Truss Analysis Assumptions . . . . .	5
	4.2 Finite Element Analysis Assumptions . . . . .	6
5	Evaluation of Truss Designs . . . . .	6
	5.1 Modified Final Design . . . . .	7
6	Detailed Engineering Drawing Of Final Structure . . . . .	7
7	Comparison of Deflection of FEA and Experiments . . . . .	7
8	Conclusion . . . . .	7
9	Appendix . . . . .	8
	9.1 Design 1 - Structure Properties and FEA . . . . .	8
	9.2 Design 2 - Structure Properties and FEA . . . . .	9
	9.3 Design 3 - Structure Properties and FEA . . . . .	10
	9.4 Truss Analysis: Buckling . . . . .	11
	9.5 FEA Final Design Stress and Displacement Plots . . . . .	19
	9.6 Final Design Drawings and Real Life Pictures . . . . .	21
	9.7 Mechanical Lab Demonstration Results . . . . .	22

# List of Figures

1	Design 1 - Design and mechanical properties . . . . .	8
2	Design 1 - Vertical Deflection FEA . . . . .	8
3	Design 1 - Horizontal Deflection FEA . . . . .	8
4	Design 2 - Design and mechanical properties . . . . .	9
5	Design 2 - Vertical Deflection FEA . . . . .	9
6	Design 2 - Horizontal Deflection FEA . . . . .	9
7	Design 3 - Design and mechanical properties . . . . .	10
8	Design 3 - Vertical Deflection FEA . . . . .	10
9	Design 3 - Horizontal Deflection FEA . . . . .	10
10	Free Body Diagram and Node/Truss Naming convention for Vertical Force application analysis . . . . .	11
11	Free body diagram and node/truss naming convention for horizontal force application analysis . . . . .	14
12	Stress Plot for Final Design ( $F_{App}$ in horizontal Direction) . . . . .	19
13	Stress Plot for Final Design ( $F_{App}$ in vertical direction) . . . . .	19
14	Displacement Plot for Final Design ( $F_{App}$ in horizontal Direction) . . . . .	20
15	Displacement Plot for Final Design ( $F_{App}$ in vertical direction) . . . . .	20
16	SolidWorks Model Design . . . . .	21
17	Built truss structure with mounting plates . . . . .	21
18	Measured Test results by TA . . . . .	22

# 1 Introduction

MSE 312 term project requires teams to develop a one degree of freedom pick and place arm spanning 30 cm to carry a payload from point A to B accurately in the fastest time possible. Truss structure design is a economical solution for minimal material use and high structural strength. The mechanical design of the arm will be a truss structure meeting the design constraints outlined and made of brass rods joined using solder. The truss structure will be mounted to a DC motor shaft using a mounting plate and set screws. The electromagnet will be mounted onto a smaller brass plate fixed onto the the end of the truss arm.

## 2 Design Approach

Three truss structure designs which meet the design constraints outlined in section 2.1 will be further evaluated on a developed evaluation decision matrix in order to select the most suitable design. The evaluation matrix will consist of criteria outlined in the design objectives.

### 2.1 Design Constraints

Design concepts are required to satisfy the constraints shown below to be considered one of the three designs that would be considered for further investigation.

- Span 30 cm (center of motor mount to center of electromagnet mount)
- Each truss structure must have at least one redundant member
- Tool access to mounting screws in the truss structure
- No single link longer than 8 cm
- Max width of 10 cm and max height of 5 cm
- Overhang past the motor or electromagnet (max 5 cm)
- Max clearance between electromagnet surface and object is 2mm
- No rods yield

### 2.2 Design Objectives

- Minimize Inertia

$$T = J\alpha$$

Since torque ( $T$ ) is fixed (continuous torque = 0.20208 Nm), in order to maximize  $\alpha$ , we should minimize  $J$  (Mass moment of Inertia).

- Minimize deflection in truss (x and y)  
Minimizing deflection in the system, we will be able to have higher precision in the pick and place operation.
- Center of mass close to motor  
If the center of mass of the truss is further away from the motor, it will put additional strain on the motor and the truss will be harder to control. We will have to add a counter weight

to the truss move the center of mass closer to the motor. However, this counter weight will further increase the inertia.

$$J_{addedbycounterweight} = m_{counterweight} * d_{frommotorpivot}$$

- Ease of construction

We are allowed a clearance between the electromagnet surface and object of 2 mm. Some of this clearance will be taken up by the deflection as a result of the weight at the end of the truss. However, it is possible that the tolerance stack-up of the manufacturing process decreases this clearance.

## 2.3 Task Distribution Among Team Members

- Initial design concept FEA, J, center of mass etc calculations for design matrix - Each group member is responsible for creating 1 design and doing the calculations required for evaluation
- Building final truss design - Brindan
- Buckling calculations for final truss - Anjan
- Engineering Drawing for final design - Neeraj
- Report writing - Anjan, Brindan, Neeraj

# 3 Quantitative Analysis of 3 Truss Structures and Design After Final Iteration

## 3.1 FEA Summary of 3 Designs

### Supports

Joints on given design that were to be coincident to the motor's brass plate were fixed in the FEA.

### Forces

The total load to be analyzed was a force of 500 grams in the horizontal and vertical direction. For a given design, we found the trusses on which the electromagnet would be mounted. Then we counted the number of joints joining the trusses' which held the payload. Finally, we calculated the force applied per joint using the following equation:

$$F_{Per-Joint} = \frac{0.5 * 9.81}{NumberofJoints}$$

### Final Design FEA

In evaluation of the concepts, we quantifiably selected to pursue design 2. The initial performance values for design 2 are given in section 3.2. After some manufacturing changes to design 2 as outlined in section 5.1, we re-performed our FEA according to our new model. The final FEA simulation indicates that the maximum vertical deflection with a 500 gram load is 1.746 mm, and the maximum horizontal deflection with a 500 gram load is 2.765 mm. This is reasonable because the minimum vertical clearance required is 2 mm and simulation is our worst case. Further horizontal sectional modulus is lower (distance of material from neutral axis is less). The final stress and displacement plots are shown in the appendix (Section 9.5).

### 3.2 Mass of System, Center of Gravity, and Inertia (J)

The mass of system, center of gravity, and inertia (J) were calculated using Soldiworks' evaluate function. To calculate center of gravity and inertia, we made a new coordinate system on which the origin was the motor's axis.

The values of the 3 designs are summarized in the following table:

Criterion	Design 1	Design 2	Design 3
Mass of System (grams)	146.86	133.11	172.89
Inertia (J) (grams* <sup>2</sup> square millimeters)	3959425.52	3229402.38	5332033.36
x Deflection (mm)	4.8	1.33	6.56
y Deflection (mm)	0.97	2.35	3.35
Centre of Gravity (distance from axis of rotation) (mm)	135.44	124.36	149.9

### 3.3 Buckling Analysis

Buckling was checked for our final design using method of joints. This was the iterative portion of our design process. Once we chose the best design in section 5, we did a hand calculation to check for buckling. The support locations, forces, and force applied location were calculated as discussed in the FEA summary section: 3.1.

It is shown in the appendix (section 9.4) that the final design does not have any rods that buckle. A factor of safety of 100 was used.

## 4 Calculation Assumptions and Effect on Results

Both calculation methods rely on following the engineering drawing closely when manufacturing and accurately building the model. After manufacturing, we found that dimensions were within the drawing  $\pm 0.5mm$ .

### 4.1 Truss Analysis Assumptions

With truss analysis we make the following critical assumptions:

1. **Friction-less pin joints**

2. **From (1) it follows that there is no bending moments**

Assumption 1 and 2 are not accurate as per our real life structure. The joints are soldered and not frictionless pin joints. Therefore, the joints transfer bending moments and there will be bending moments in the rods. However, the forces are primarily axial and this is a safe assumption.

3. **Small displacements**

As each rod experiences a different force as a consequence of the bridge's overall geometry, some of the rods will elongate or shrink more than the rest. As such, the overall geometry and angles will change when the structure is loaded which will cause the load distribution across the rods to change. However, as shown in the section where we loaded our bridge in real life, these displacements are small and can be largely ignored.

#### 4. **All forces applied at joints**

If there is a force transverse to the cross section is applied along a truss, we would simplify it to 2 point forces. One force at the beginning of the truss and another at the end. This is used at the end where the magnet and payload is loaded and at the supports in the beginning. The truss that bears the support forces has a pin support in the beginning and a roller at the end of the truss.

#### 5. **$E = 102$ GPa for our brass material**

Since the material was sourced from a arts and crafts store, we were not given accurate data for material properties. In class, it was stated that the young's modulus for brass depends on how the material was formed but can be considered between 102 GPa and 125 GPa. Since buckling strength is proportional to  $E$ , we opted to be conservative when checking buckling and have  $E = 102$  GPa.

#### 6. **The trusses are weightless**

The total weight of the bridge is 126 grams. This value is insignificant to the 500 grams weight applied. As such, the force developed by the payload will trump the weight of the trusses. The effect of the weight of the trusses will be not be noticeable in the final calculations.

## 4.2 **Finite Element Analysis Assumptions**

### 1. **Contacts do not change in time**

Our joints are soldered. Even though we tried to ensure all joints were equally rigid, it is possible some joints are better at transferring forces to the other links connected to the joint. As such, there will be some error in our final calculation as FEA assumes all joints are equally well made. FEA will likely underestimate our deflection.

### 2. **Rods do not buckle**

The standard FEA package in solidworks does not check for buckling. It is likely that some rods fail due to buckling and solidworks FEA overestimates our factor of safety. As such, we must do hand calculations in order to check if an of the rods fail due to buckling.

### 3. **$E = 102$ GPa for our brass material**

This is to be consistent with our truss analysis. To see justification see section 4.1.

## 5 **Evaluation of Truss Designs**

Evaluation of valid concept designs for the truss structure are compared on the evaluation decision matrix by summing the total score of each concept. The weight distribution of the score is dependant on the the importance of the criteria in achieving the goal of the project: accurately pick and place the load in the fastest time possible.

Criterion	Weight	Design 1	Design 2	Design 3
Mass of System	20	18.1	20	15.4
Inertia	20	16.3	20	12.1
x Deflection (min.)	20	5.55	20	4
y Deflection (min.)	8	8	3.3	2.3
Centre of Gravity (distance from axis of rotation)	16.5	15.1	16.5	13.7
Manufacturability	9.5	8.5	9	9.5
Number of joints (min.)	6	5.1	6	3.7
$\Sigma$	100	76.65	94.80	60.70

Evaluating the concept designs using the evaluation decision matrix, the design of optimal choice is design number 2. This design had a higher evaluation decision matrix score over the other two designs.

## 5.1 Modified Final Design

Slight modifications were made while building the structure. In order to have stability on the mounting base, the majority of the structure needed to be shifted back for level mounting and the front support structure was extended from 3cm to 5cm. Another modification included changing the position of redundant member to reduce the extra mass in the system.

## 6 Detailed Engineering Drawing Of Final Structure

A detailed engineering drawing of the final design software is included in the Appendix of this document. Please see page 18.

## 7 Comparison of Deflection of FEA and Experiments

The mechanical lab demonstration results are included in Appendix 9.7. In the vertical load test, a 500g mass in showed 3mm deflection whereas our simulation FEA had 1.746 mm therefore the % error is 71.8%. This experimental error may be due to incorrect assumption in the Young's Modulus value which could be too stiff. Another factor contributing to the error is significant flexibility in the soldered joints which is not equivalent to the weldment structure simulated by FEA. Asymmetry in the truss structure due to the slight variations in angled links may have increased non-uniformity in forces on the structure and contributed towards the greater deflection.

## 8 Conclusion

In conclusion, three valid preliminary designs were developed and evaluated based on the required criteria in the evaluation decision matrix. The design evaluation selection process allowed us to select a optimal arm design (design 2) based on the sum value on the matrix. Using Solidworks' FEA and simulation, mechanical properties were evaluated and compared to mechanical demonstration values. We found that the vertical deflection compared to theoretical had 71.8 % error which can be explained due to Young's Modulus assumption errors, Soldered joint material flexibility and manufacturing error. Considering that the actual payload is 50g in weight and the deflection will be far less, our arm will be able to pick and place the load accurately without failure.

# 9 Appendix

## 9.1 Design 1 - Structure Properties and FEA

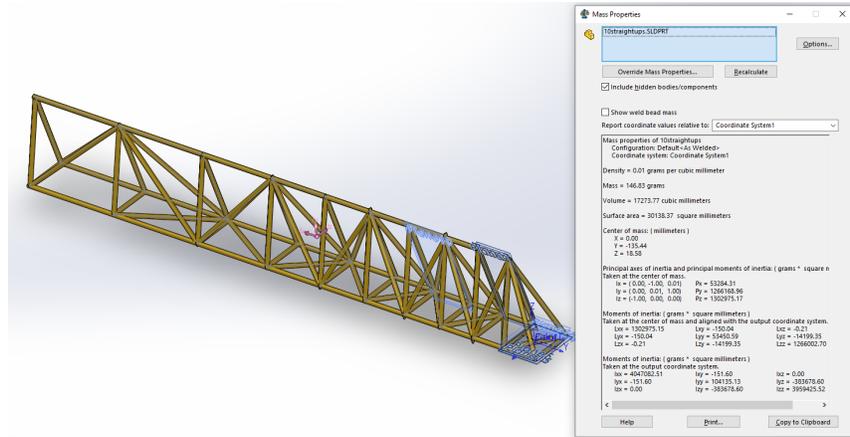


Figure 1: Design 1 - Design and mechanical properties

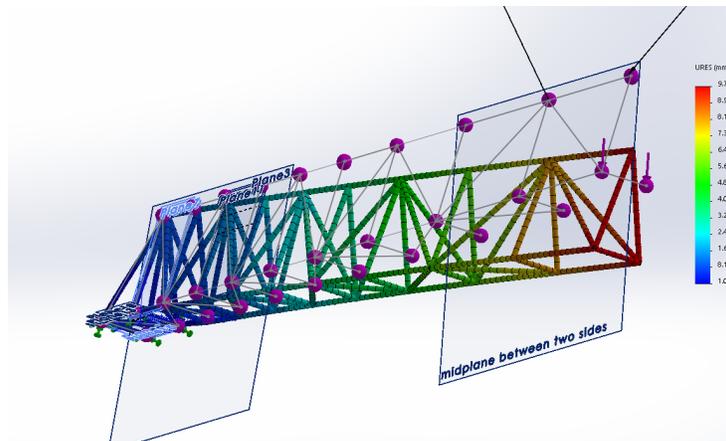


Figure 2: Design 1 - Vertical Deflection FEA

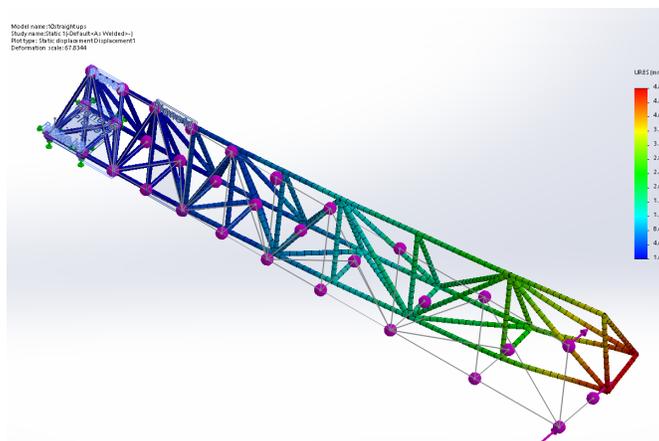


Figure 3: Design 1 - Horizontal Deflection FEA

## 9.2 Design 2 - Structure Properties and FEA

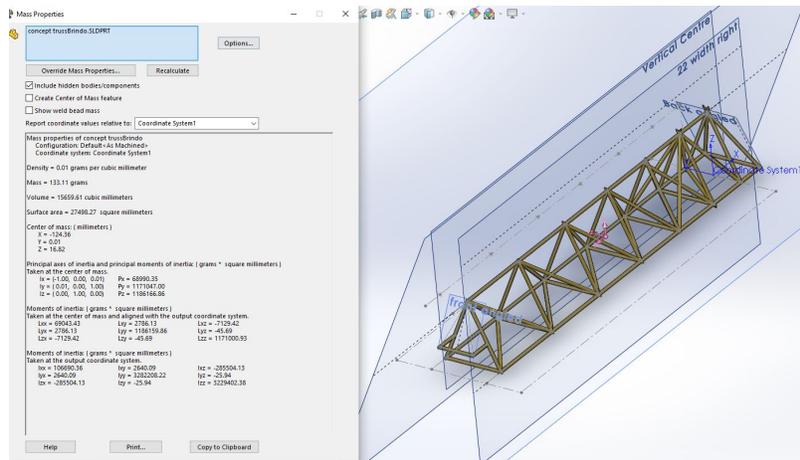


Figure 4: Design 2 - Design and mechanical properties

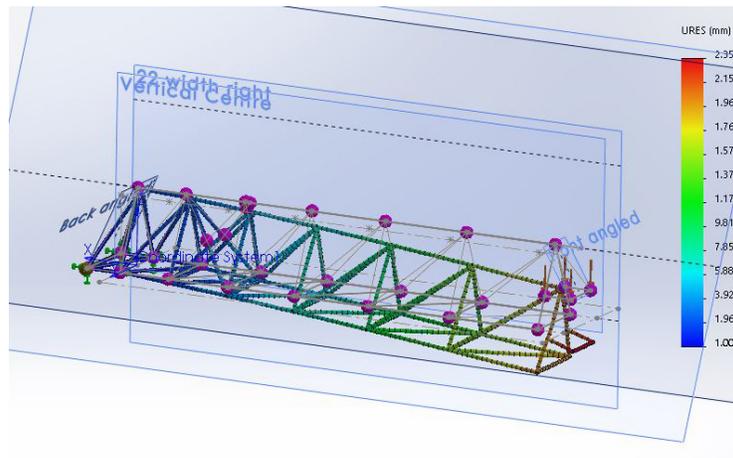


Figure 5: Design 2 - Vertical Deflection FEA

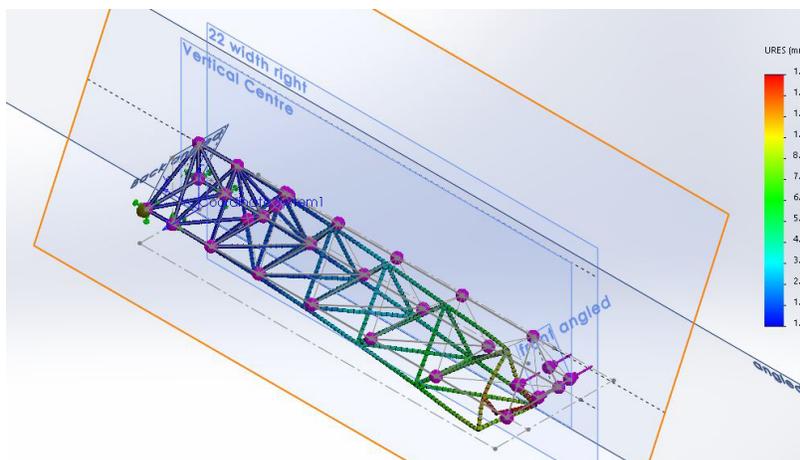


Figure 6: Design 2 - Horizontal Deflection FEA

### 9.3 Design 3 - Structure Properties and FEA

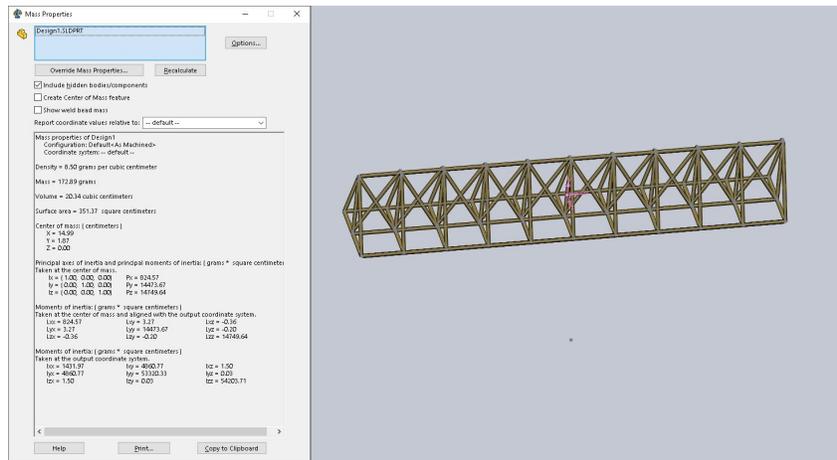


Figure 7: Design 3 - Design and mechanical properties

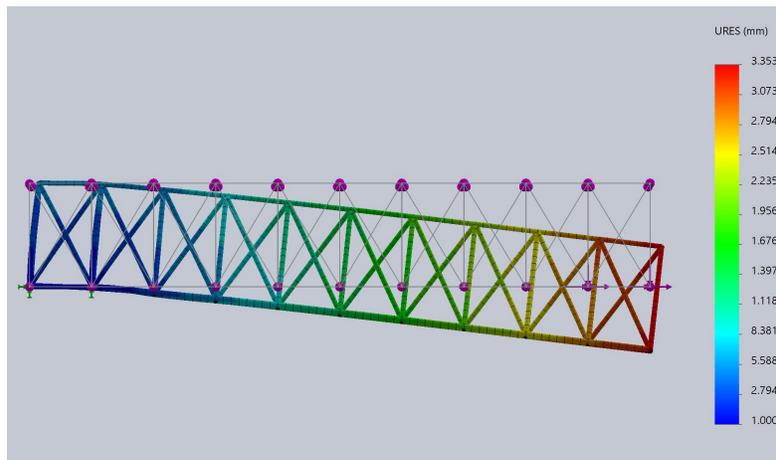


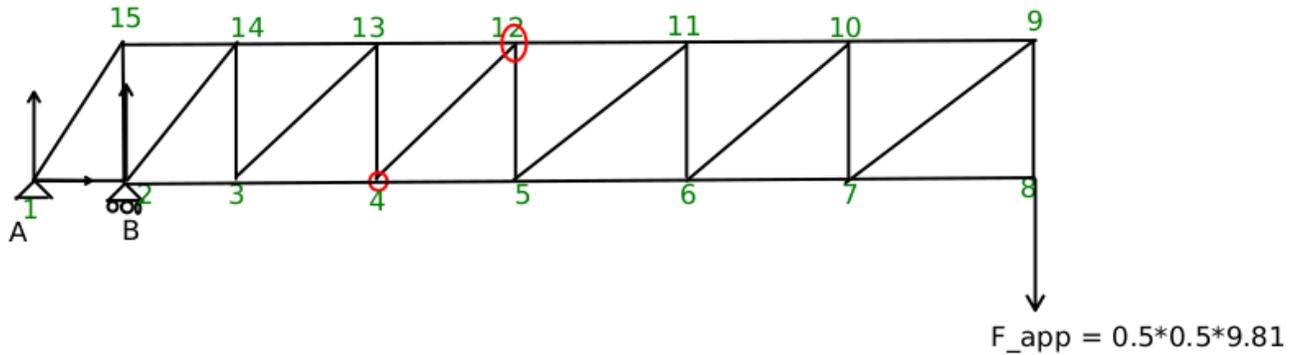
Figure 8: Design 3 - Vertical Deflection FEA



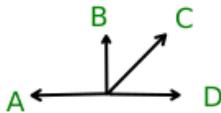
Figure 9: Design 3 - Horizontal Deflection FEA

## 9.4 Truss Analysis: Buckling

Vertical



Bottom Left nodes



Top Right Nodes

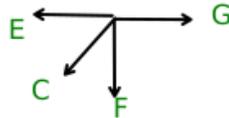


Figure 10: Free Body Diagram and Node/Truss Naming convention for Vertical Force application analysis

$B_y$	33.5909831460674
$A_y$	-31.1384831460674
((W))	22.25
((H))	48.21
Node A	
F1-15	34.294809479998
F1-2	-14.3711107654014
Node 15	
$F_{15} - 14$	14.3711107654014
$F_{15} - 2$	-31.1384831460674
((W))	32.5
((H))	48.21
Node 2	
$F_2 - 14$	-2.70109561391135
$F_2 - 3$	-12.7177971375233

Node 14	
F <sub>14</sub> – 13	12.7177971375233
F <sub>14</sub> – 3	2.4525
W	32.5
H	48.21
2-3-14-15	
Bottom Left	
A	-14.3711107654014
B	-31.1384831460674
C	-2.70109561391135
D	-12.7177971375233
Top Right	
E	14.3711107654014
C	-2.70109561391135
F	2.4525
G	12.7177971375233
3-4-13-14	
W	40
H	48.21
Bottom Left	
A	-12.7177971375233
B	2.4525
C	-3.18674767893517
D	-10.6829495955196
Top Right	
E	12.7177971375233
C	-3.18674767893517
F	2.4525
G	10.6829495955196
4-5-12-13	
W	45
H	48.21
Bottom Left	
A	-10.6829495955196
B	2.4525
C	-3.35487836509921
D	-8.3937461107654
Top Right	
E	10.6829495955196
C	-3.35487836509921

F	2.4525
G	8.3937461107654
5-6-11-12	
W	50
H	48.21
Bottom Left	
A	-8.3937461107654
B	2.4525
C	-3.53333423429597
D	-5.85018668326073
Top Right	
E	8.3937461107654
C	-3.53333423429597
F	2.4525
G	5.85018668326073
6-7-10-11	
W	55
H	48.21
Bottom Left	
A	-5.85018668326073
B	2.4525
C	-3.72062987531815
D	-3.0522713130056
7-8-9-10	
W	60
H	48.21
A	-3.0522713130056
B	2.4525
C	-3.91549695673447
D	0
A	4.42455978812812
B	-3.91549695673447
C	2.4525
Buckling Analysis	
E	102000000000
I	1.57498973252289E-12

$r$	0.00238
$L$ (Use longest rod in buckling calculation)	0.0059242041
$P_{cr}$	45177.0240642098

$F_{Max}$	451.770240642098
-----------	------------------

$F_{max}$  is  $P_{cr}$  with a factor of safety of 100. Since no rod has a force  $> F_{max}$ , we can assume no buckling

### Horizontal

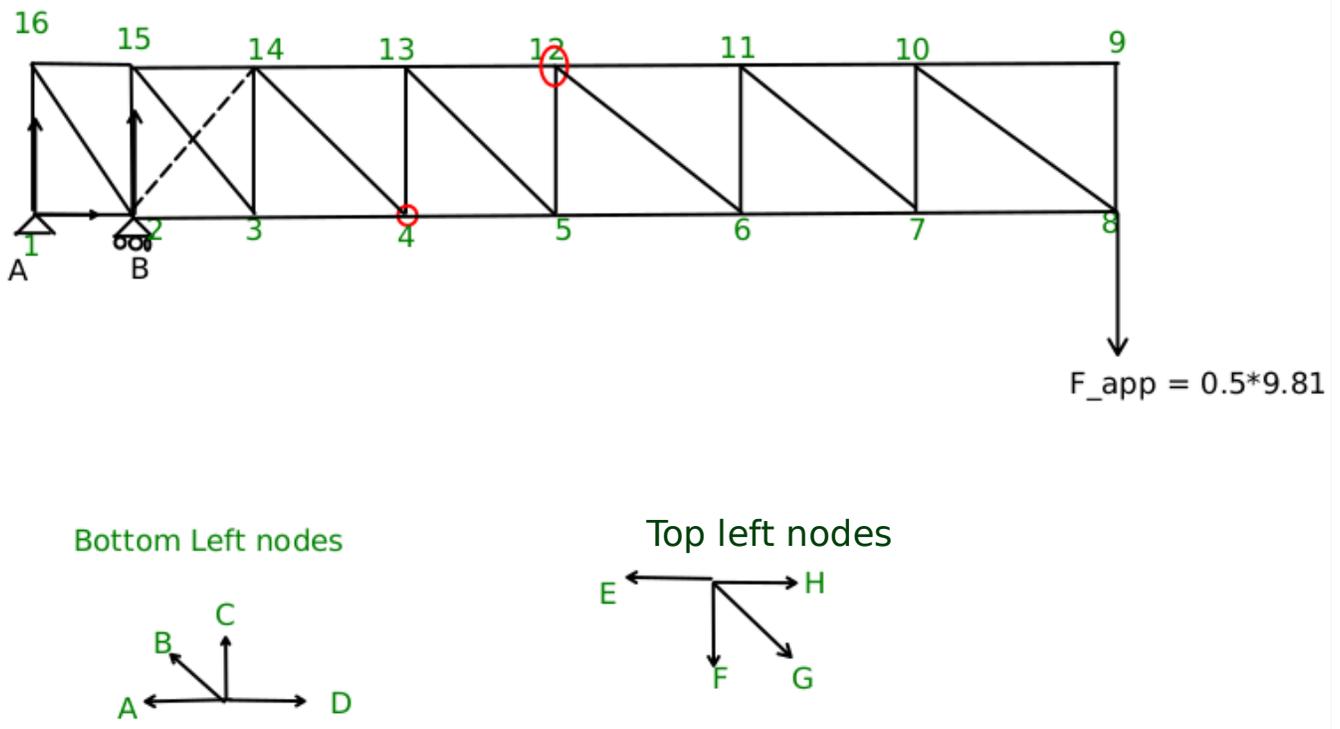


Figure 11: Free body diagram and node/truss naming convention for horizontal force application analysis

Redundant member shown with dashed line

$B_y$	67.1819662921349
$A_y$	-62.2769662921349

((W))	22.25
((H))	44

Node 1

F1-16 62.2769662921349  
F1-2 0

Node 16

F16-15 31.4923295454546  
F16-2 -69.7867276117115

Node 2 Bottom left

A 0  
B -69.7867276117115  
C -4.905  
D -31.4923295454546

((W)) 32.5

((H)) 44

Node 15 Top Left

E 31.4923295454546  
F -4.905  
G 5.49647677649763  
H 27.8693181818182

Node 3 Bottom left

A -31.4923295454546  
B 5.49647677649763  
C -4.905  
D -27.8693181818182

((W)) 40

((H)) 44

Top Left

E 5.49647677649763  
F -4.905  
G 5.49647677649763  
H 1.03738586740672

Node 4 Bottom left

A -27.8693181818182  
B 5.49647677649763  
C -4.905  
D -23.4102272727273

((W)) 45

((H)) 44

Top Left  
E 5.49647677649763  
F -4.905  
G 5.49647677649763  
H 0.479999503770354

Node 5 Bottom left  
A -23.4102272727273  
B 5.49647677649763  
C -4.905  
D -18.39375

((W)) 50  
((H)) 44

Top Left  
E 5.49647677649763  
F -4.905  
G 5.49647677649763  
H -0.07738685986601

Node 6 Bottom left  
A -18.39375  
B 5.49647677649763  
C -4.905  
D -12.8198863636364

((W)) 55  
((H)) 44

Top Left  
E 5.49647677649763  
F -4.905  
G 5.49647677649763  
H -0.634773223502374

Node 7 Bottom left  
A -12.8198863636364  
B 5.49647677649763  
C -4.905  
D -6.68863636363637

((W)) 60  
((H)) 44

Top Left  
E 5.49647677649763

F -4.905  
 G 5.49647677649763  
 H -1.19215958713874

Node 8 Bottom  
 A -6.68863636363637  
 B 5.49647677649763  
 C 0

Member	P1N	L	P-ex	$P_{ex} * P1n * L$	$P_1N * P1n * L$
F <sub>2-14</sub>	1.0000	54.7015	0.0000	0.0000	54.7015
F <sub>2-15</sub>	-0.8044	44.0000	-4.9050	173.5983	28.4682
F <sub>2-3</sub>	-0.5941	32.5000	-31.4923	608.0966	11.4723
F <sub>3-14</sub>	-0.8044	44.0000	-4.9050	173.5983	28.4682
F <sub>14-15</sub>	-0.5941	32.5000	27.8693	-538.1386	11.4723
F <sub>3-15</sub>	1.0000	54.7015	5.4965	300.6653	54.7015
				717.8199	189.2840
	P	$P_{actual}$			
	3.79229	3.79229			
		-7.95539			
		-33.74546			
		-7.95539			
		25.61619			
		9.28877			

Buckling Analysis		
E	102000000000	
I	1.57498973252289E-12	
r	0.00238	
L	0.0059242041	Use longest rod in buckling calculation
$P_{cr}$	45177.0240642098	

$F_{max} = 451N$  is  $P_{cr}$  with a factor of safety of 100. Since no rod has a force  $> F_{max}$ , we can assume no buckling

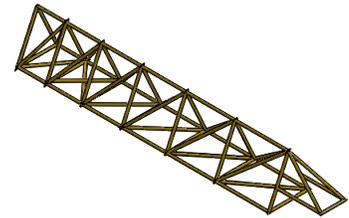
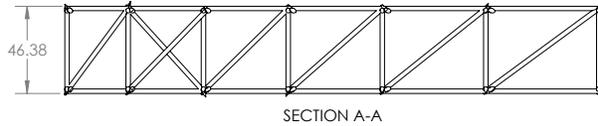
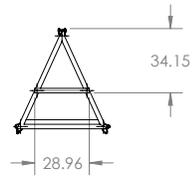
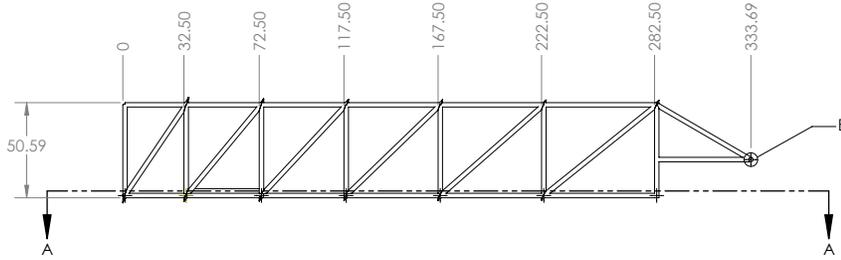
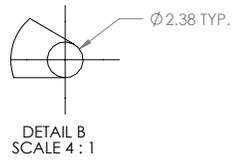
4

3

2

1

REVISIONS					
REV.	DESCRIPTION	DATE	DRAWN BY	CHECKED	APPROVED
1	INITIAL RELEASE	2019-06-06	N.B.	A.M.	B.R.



**PROPRIETARY AND CONFIDENTIAL**  
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF ANGLISREV, BRIDGAN, AND HERKAL. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF ANGLISREV, BRIDGAN, AND HERKAL IS PROHIBITED.

UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MM TOLERANCES: ± 0.5 MM	DRAWN N.B. 2019-06-06	NAME N.B.	DATE 2019-06-06
INTERPRET GEOMETRIC TOLERANCING PER: MATERIAL: <b>BRASS AND SOLDER</b>	CHECKED A.M. 2019-06-06	ENG APPR. B.R. 2019-06-06	
FINISH <b>UNFINISHED</b>	COMMENTS:		
DO NOT SCALE DRAWING			

TITLE: <b>MSE312 TRUSS</b>		
SIZE <b>B</b>	DWG. NO. N/A	REV <b>1</b>
SCALE: 1:2	WEIGHT: 126 grams	SHEET 1 OF 1

4

3

2

1

## 9.5 FEA Final Design Stress and Displacement Plots

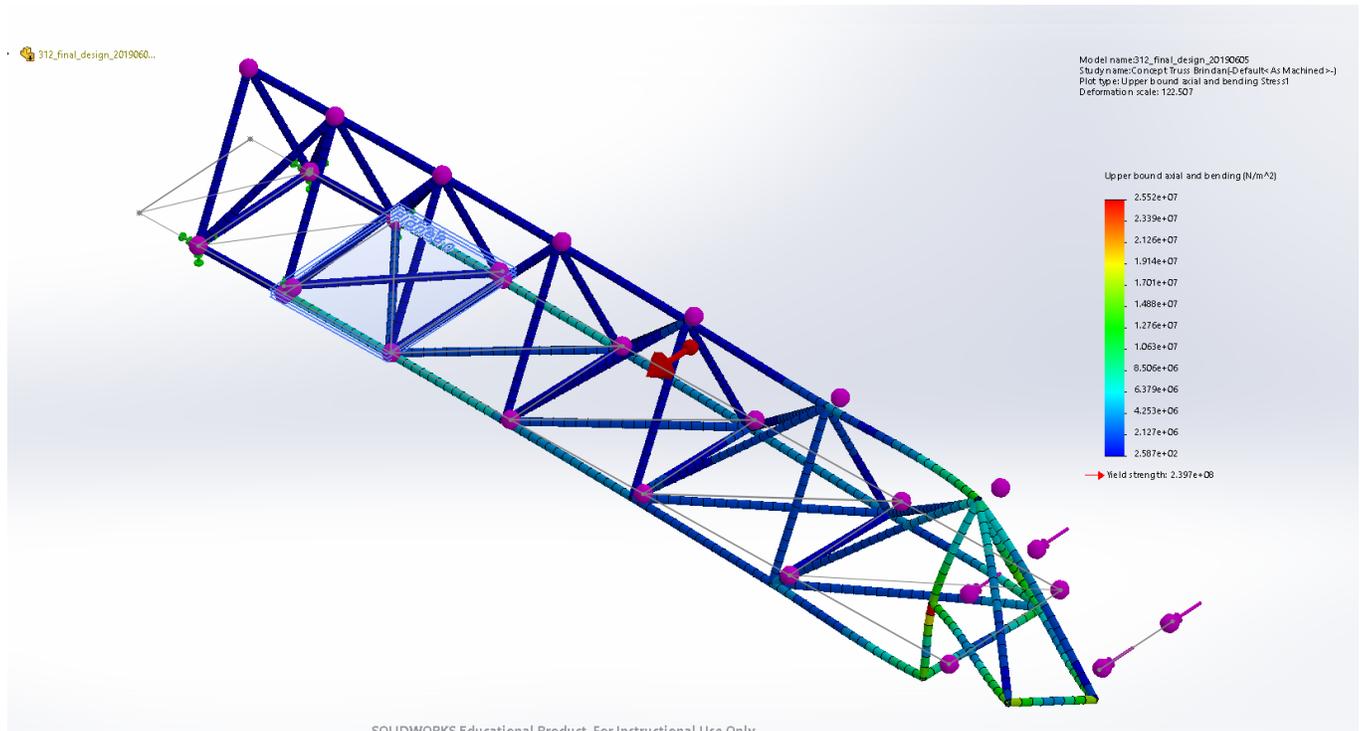


Figure 12: Stress Plot for Final Design ( $F_{App}$  in horizontal Direction)

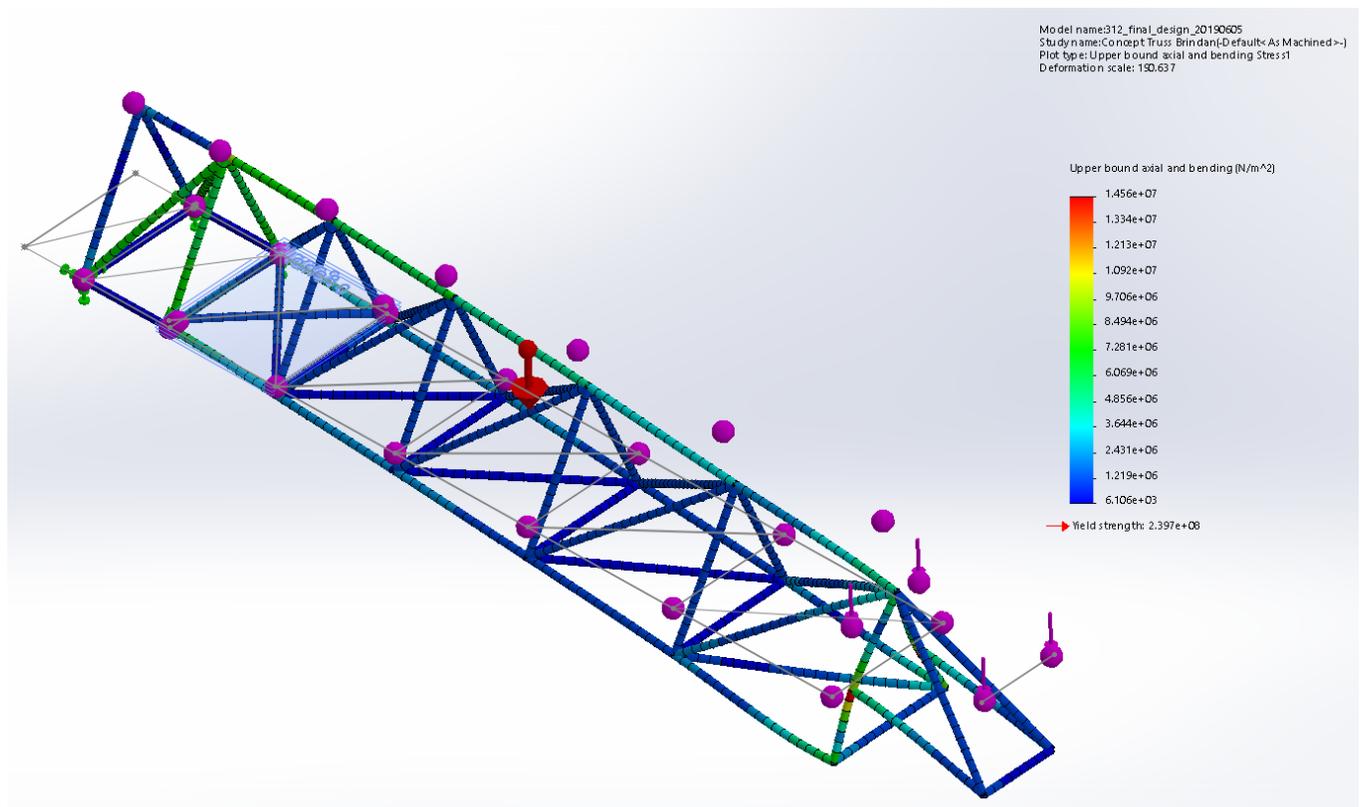


Figure 13: Stress Plot for Final Design ( $F_{App}$  in vertical direction)

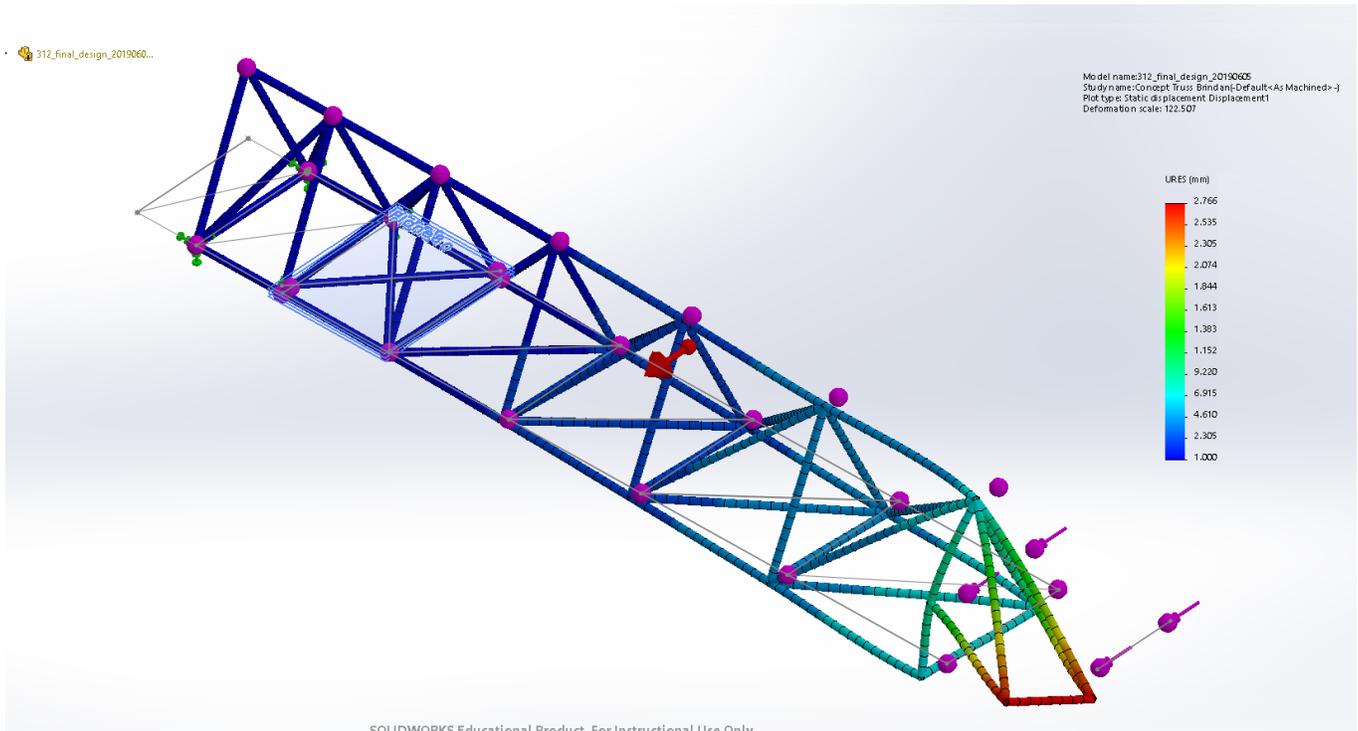


Figure 14: Displacement Plot for Final Design ( $F_{App}$  in horizontal Direction)

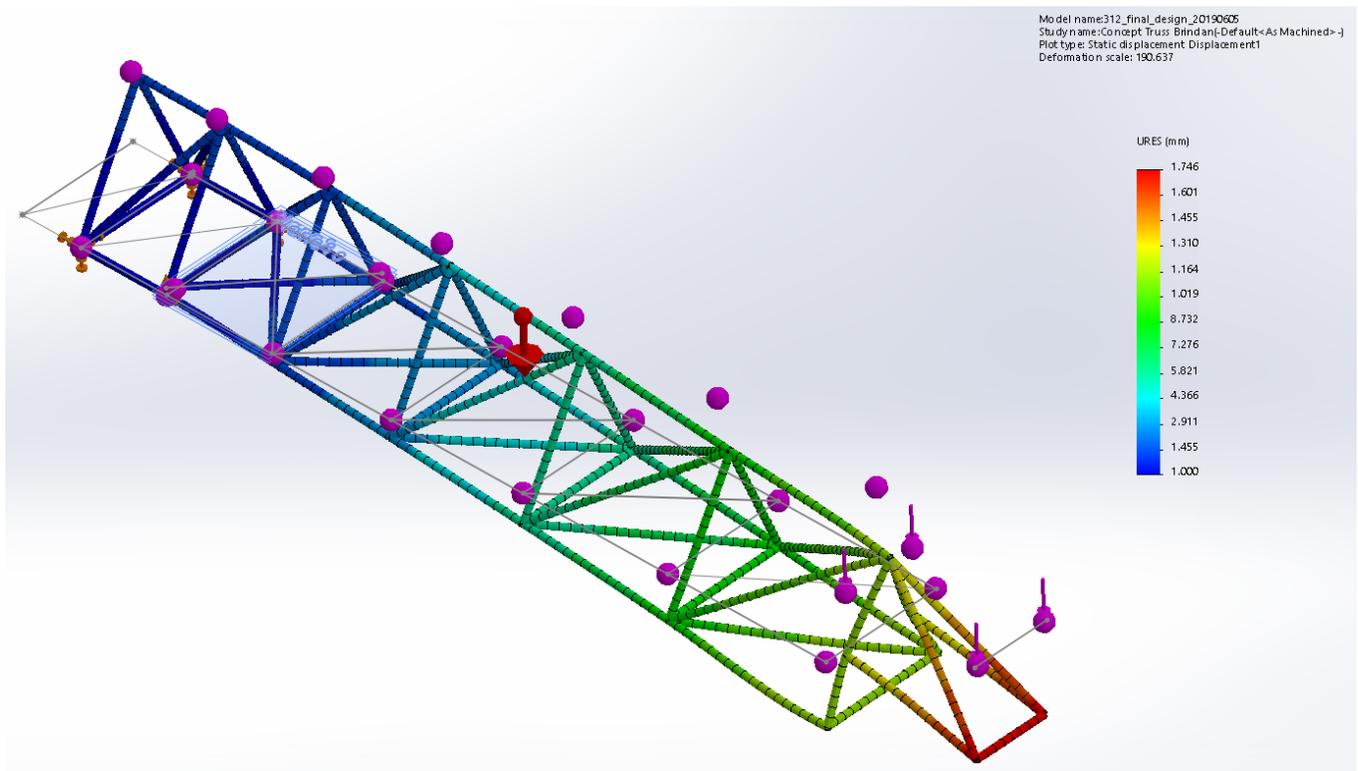


Figure 15: Displacement Plot for Final Design ( $F_{App}$  in vertical direction)

### 9.6 Final Design Drawings and Real Life Pictures

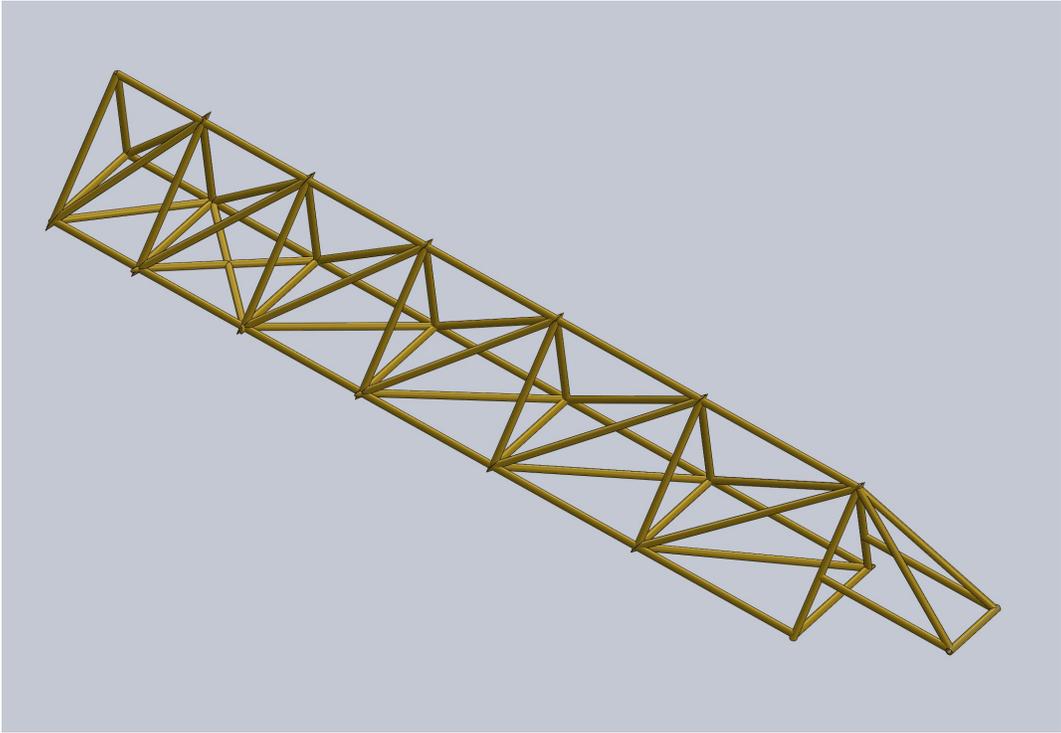


Figure 16: SolidWorks Model Design

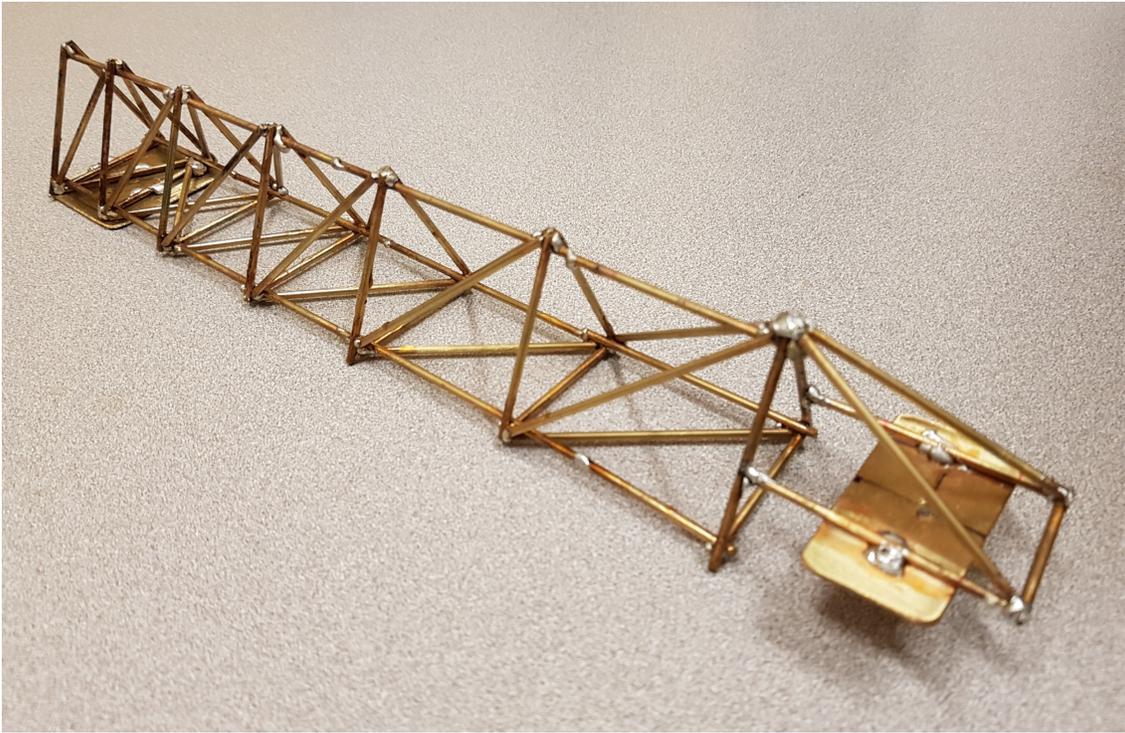


Figure 17: Built truss structure with mounting plates

### 9.7 Mechanical Lab Demonstration Results

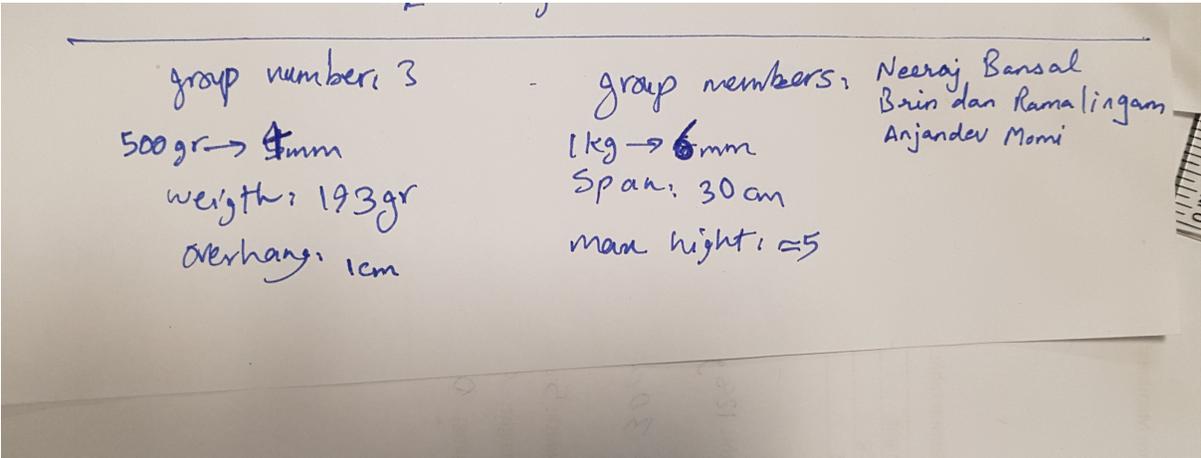


Figure 18: Measured Test results by TA

# 13.2 Electrical Report

MSE 312  
Electrical Report

Anjandev Momi - 301281186  
Brindan Ramalingam - 301271567  
Neeraj Bansal - 301323495

August 2, 2019

# Contents

1	Introduction . . . . .	3
2	Detailed Analysis and Design of Circuits . . . . .	3
2.1	Motor Drive Circuit - Full Bridge DC Motor Drive IC . . . . .	3
2.2	Electromagnet Drive Circuit . . . . .	4
2.3	Power Protection Circuit . . . . .	6
3	Heat Sink Design . . . . .	7
3.1	H-Bridge Power Dissipation . . . . .	7
4	Simulation Results . . . . .	8
4.1	Electromagnet . . . . .	9
4.2	Motor . . . . .	10
4.3	Power stage simulation . . . . .	12
5	Experimental Results . . . . .	14
5.1	DC Motor Characterization . . . . .	15
5.2	Transfer Characteristics of Power Stage . . . . .	17
6	Conclusion . . . . .	18

# List of Figures

1	Motor Drive Circuit . . . . .	3
2	TB6568KQ Full-Bridge DC Motor Driver . . . . .	4
3	Force vs stroke with various duty cycles . . . . .	4
4	Electromagnet Drive Circuit . . . . .	5
5	Power Protection Circuit . . . . .	6
6	Electrical Circuit Simulink Model . . . . .	8
7	Simulink Simulation - PWM input for the electromagnet . . . . .	9
8	Simulink Simulation - Voltage across Electromagnet . . . . .	10
9	Simulink Simulation - 80% PWM input for the motor . . . . .	10
10	Simulink Simulation - Voltage across Motor . . . . .	11
11	Electrical Circuit Board . . . . .	14
12	Electrical Circuit Board connected with relay protection and drive components . . . .	14

# 1 Introduction

MSE 312 term project requires teams to develop a one degree of freedom pick and place arm spanning 30cm to carry a payload from point A to B accurately in the fastest time. This system is electrically powered and controlled. The control will be provided from an Arduino micro-controller which will be provided in the integration section of the project. However, for the electronic portion of the project, we looked to design and build the circuit to control the motor and electromagnet as well as interconnect protection circuitry to react to limit and micro switch activation. Relay protection circuit for power circuit should be operational for on/off conditions and limit switch shut-off. We simulate the micro controller PWM signals for the DC motor and electromagnet using a square wave generator. The system should include heat sinking as required by the full bridge DC motor driver to provide sufficient energy dissipation and components operate under optimal conditions.

## 2 Detailed Analysis and Design of Circuits

### 2.1 Motor Drive Circuit - Full Bridge DC Motor Drive IC

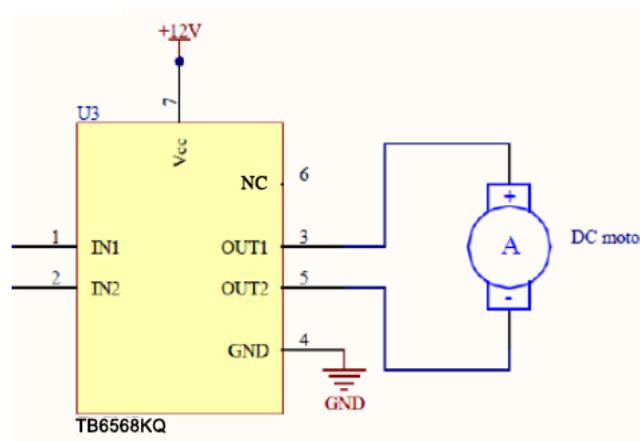


Figure 1: Motor Drive Circuit

The circuit used for driving the motor is shown in figure 1. The TB6568KQ is a full bridge DC motor driver IC manufactured by Toshiba where MOSFET transistors operate the switching control. The low ON-resistance MOS process design of the motor driver, and the PWM signals to input 1 and 2 enables efficient control of the system's motor. In order to connect the leads, we used the data sheet for TB6568KQ full bridge DC motor driver. The output from the motor driver IC to the motor are connections 3 and 5, while pin 4 is ground and 7 is the 12V power feed. In order to turn on the motor, we will input a PWM signal to pin 1 and set pin 2 to low. To turn the motor in the other direction, we send a PWM signal to pin 2 and set pin 1 to low. By changing the duty cycle of the PWM signal, we will change the mean voltages across the motor. As such, the motor will speed up and slow down. In the experimental results section of this report (Section 5.1) we show the motor voltage output when varying 5V duty cycle control signals and switching frequency. The 5V magnitude is used because this is the output which models the output of the Arduino micro-controller.

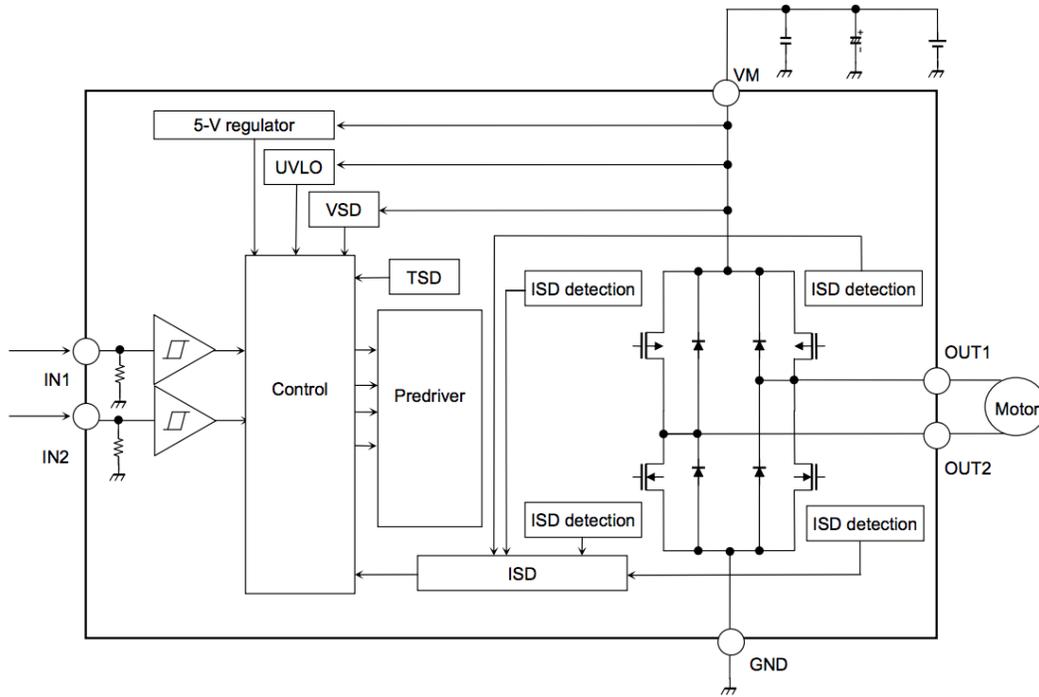


Figure 2: TB6568KQ Full-Bridge DC Motor Driver

## 2.2 Electromagnet Drive Circuit

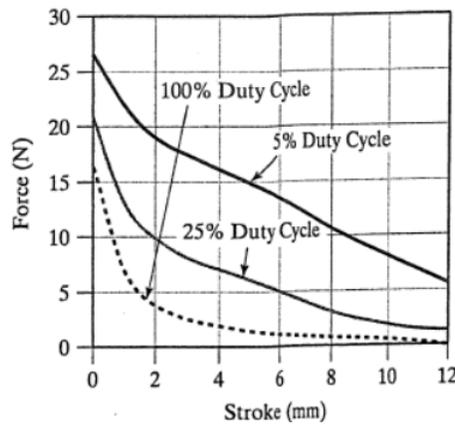


Figure 3: Force vs stroke with various duty cycles

In order to drive the electromagnet, we are using PWM control. Figure 3 shows the justification for this design choice. If we were to use a DC signal to power the electromagnet (related to 100% duty cycle in figure 3), our force generated would be proportional to the current squared. As such, our useful stroke range would only be a few mm before our force drops off significantly. However, if we use a PWM signal (5% duty cycle in figure 3), our useful stroke range increases and we have a larger force over a greater distance.

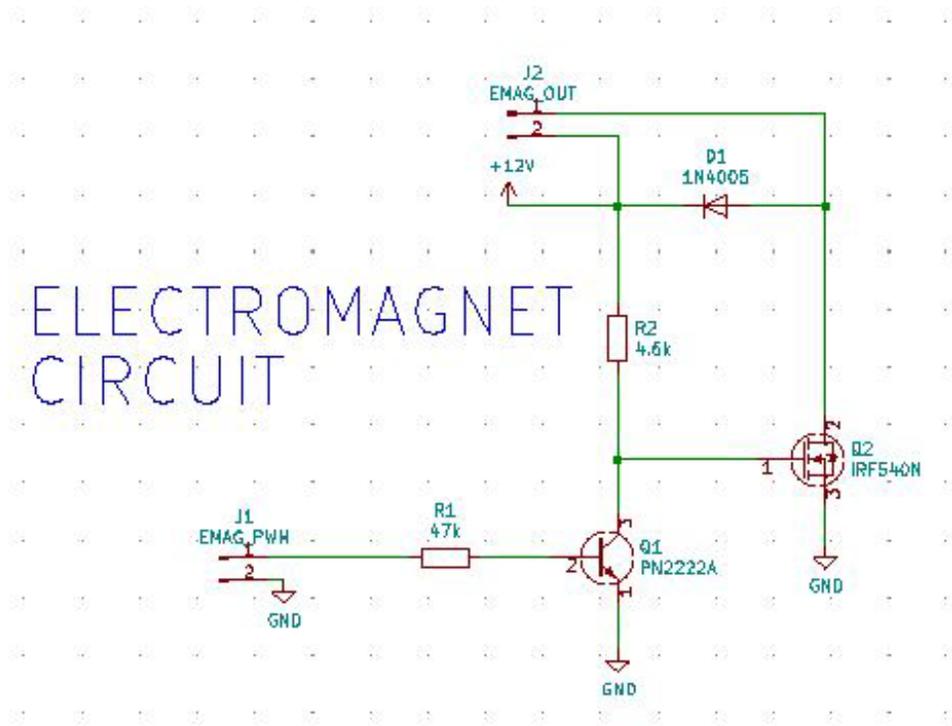


Figure 4: Electromagnet Drive Circuit

Figure 4 shows the circuit used to power the electromagnet. The BJT in the circuit is used to amplify the 5 V PWM signal to a 12 V PWM signal. The diode is placed antiparallel to the electromagnet (shown as an inductor) in order to provide a free wheeling path for the current when the MOSFET turns on and off. Without this diode, we would have a step change in the current through the inductor when the MOSFET turns off causing a voltage spike that may ruin our MOSFET. It is important to note that the circuit is active low. Meaning that when we apply the PWM signal, the electromagnet turns on. However, if we want to turn the electromagnet off, we apply a 5V DC signal to the base of the BJT.

We selected  $R_B$  and  $R_c$  such that when 5 V is applied, transistor Q1 is in saturation. When a BJT is in saturation, the following equation holds true:

$$\beta I_B > \frac{V_{cc} - V_{sat}}{R_c} \quad (1)$$

$$I_B = \frac{5 - 0.7}{R_B} \quad (2)$$

$I_B$  is the current coming from the microcontroller's digital output. We selected  $I_B = 0.1\text{mA}$  from the microcontroller datasheet. Plugging this into equation 2, we find  $R_b = 43 \text{ k}\Omega$ . From the ZN3905 transistor datasheet, we find  $30 < \beta < 100$  and  $V_{CE,sat} = 0.1\text{V}$ . From the circuit diagram,  $V_{cc} = 12\text{V}$ . Plugging this information into equation 2, we get  $R_C > 3.3 \text{ k}\Omega$ . It is important to be careful with the logic inversion here. When the input is LOW, the electromagnet is on.

Speed of the electromagnet increases as the input voltage increases. However, as voltage increases, the heat dissipated in the electromagnet may burn the electromagnet. As such, we calculate the max power rating as such:

$$\frac{((1 - d)V_{cc})^2}{R_e} = P \quad (3)$$

Here,  $R_e$  is the resistance of the electromagnet. We must ensure  $P < P_{max}$ .  $P_{max} = 1W$ . The data sheet for the electromagnet tells us the maximum on time for each duty cycle. Although higher currents with shorter duty cycles can generate larger forces, we need to maintain a duty cycle  $> 10\%$  to avoid magnetic saturation. This information will be important for the controls portion where the speed of the electromagnet matters.

### 2.3 Power Protection Circuit

The power protection circuit is designed for the following purposes:

- Display the power state of the circuit (whether or not the circuit is energized)
- Allow the user to emergency turn off the power
- Enable the limit switches to turn off power to the circuit in case the truss is unstable

There is a cable short circuiting normally open to the relay's switching port to latch the state of the relay. As such, when the user turns on the machine, the machine will remember its state until it is power cycled (ie. turned off from the power supply or user pressed red button).

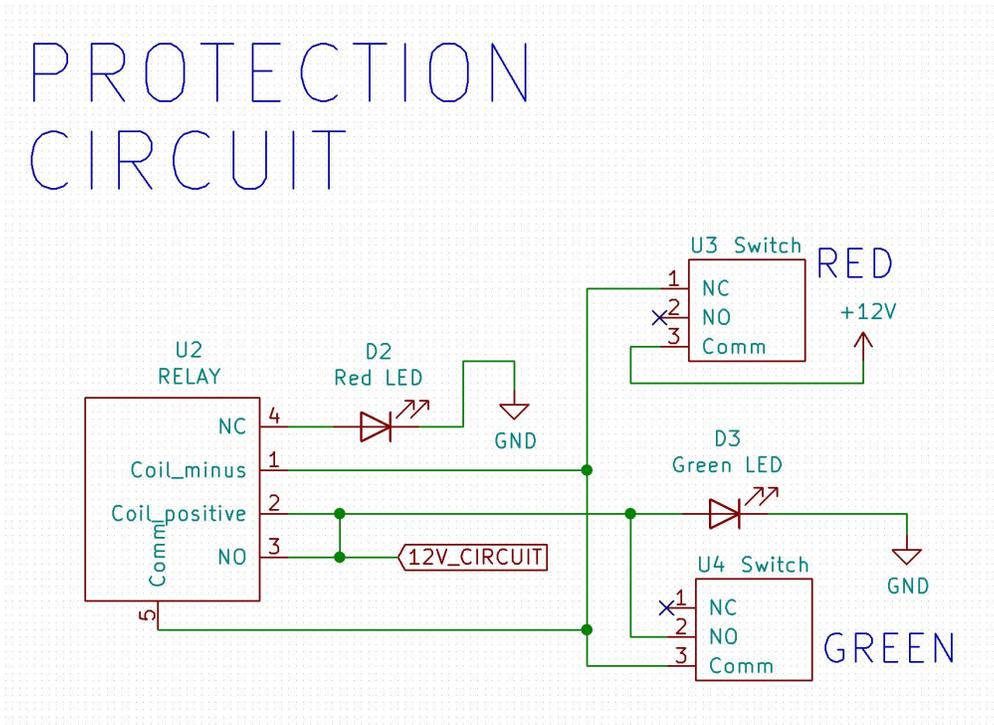


Figure 5: Power Protection Circuit

## 3 Heat Sink Design

### 3.1 H-Bridge Power Dissipation

The heat sink provided by the lab technician was a aluminum metal plate with dimensions: 14.84mm X 22.03mm X 1.59mm

As such the surface area of the heat sink is:

$$14.84 * 22.03 + 2 * (14.84 + 22.03) * 1.59 = 444.1718 \text{ mm}^2 = 4.44 \text{ cm}^2$$

In order to design the heat sink for the TB6568KQ H-Bridge IC, we check the data sheet for the power dissipation. According to the datasheet, the current being sent to the DC motor is the largest. The current being used in the control circuit is in the micro amps range and the power dissipated in the control circuit is very small. As a result, we only consider the current being sent to the DC motor to calculate the size of the heat sink.

From the data sheet:

$$I_o = 3 \text{ A}$$

$$R_{onMax} = 0.9 \Omega$$

In order to calculate total power dissipated:

$$P = I^2 R = 3^2 * 0.9 = 8.1 \text{ W}$$

$$\Theta_{CA} = \frac{T_J - T_A}{P_D} - \Theta_{jc}$$

$$\Theta_{CA} = \frac{150 - 25}{8.1} - 6 = 9.4321$$

$$A = \left(\frac{50}{\Theta}\right)^2 = \left(\frac{50}{9.4321}\right)^2 = 28.1011 \text{ cm}^2$$

We find that a 28.1011 cm<sup>2</sup> heat sink is required to operate the H-bridge motor driver IC at its rated values. The heatsink we received had a surface area smaller than 28.1011 cm<sup>2</sup>. Proper engineering practice is to provide the electronics with the adequate thermal protection in order to operate the motor its rated values. As such, the team will be sourcing another off the shelf heat sink with appropriate heat sink area (as discussed with Dr. Moallem).

# 4 Simulation Results

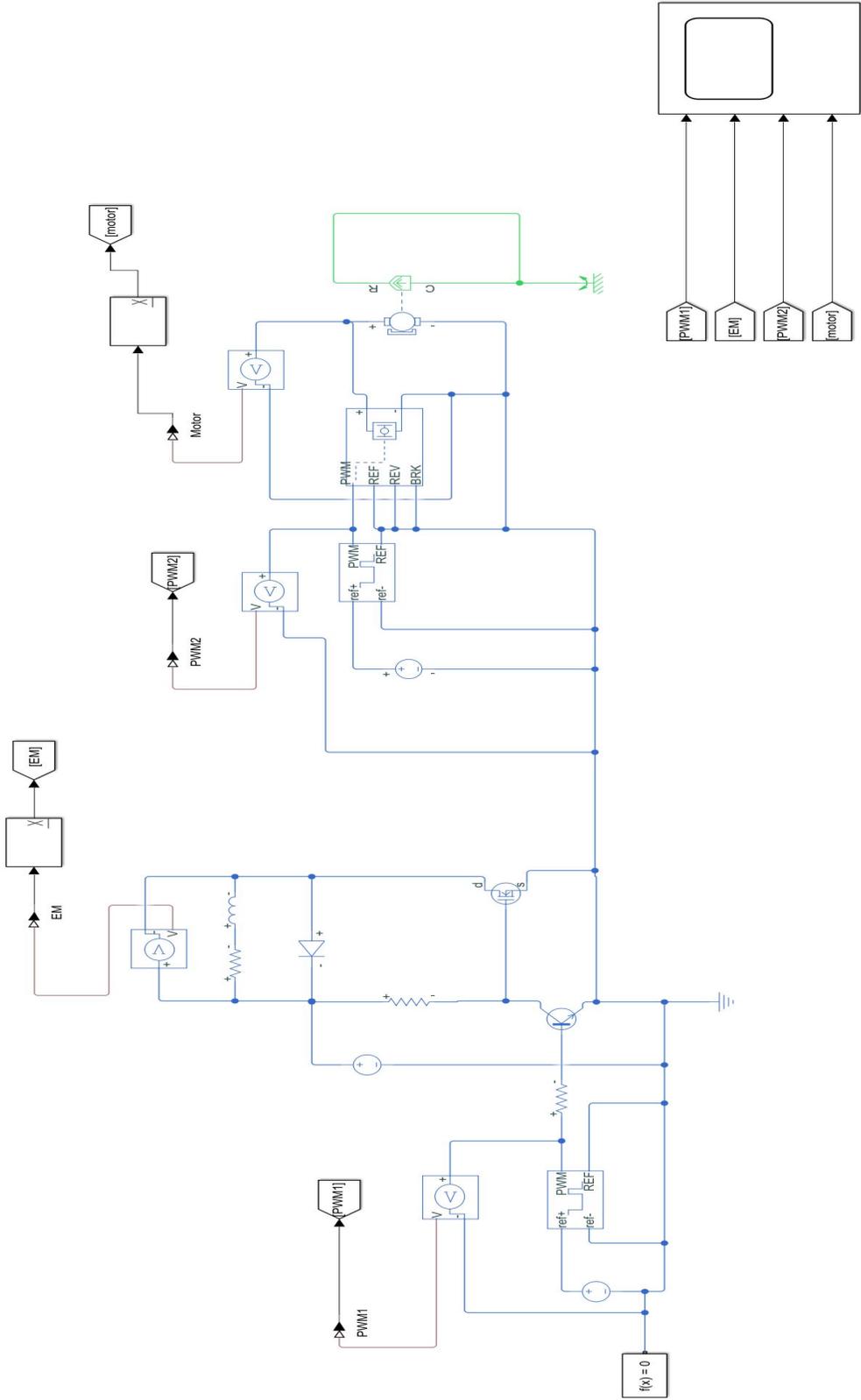


Figure 6: Electrical Circuit Simulink Model

The Simulink model was designed to simulate the electrical system and test the parameters of the circuit in order to compare it with the experimental results. The Simulink model was created by inputting the following details of system components:

- 5V PWM Electromagnet input (varying duty cycles)
- 12V Electromagnet power input
- 5V PWM Motor Input (500 Hz)
- 12V Motor power input controlled by H-Bridge

### 4.1 Electromagnet

The graphs below (figure 7,8) shows the power output of the electromagnet with respect to the PWM input provided. PWM1 signal is for the electromagnet which is 0V (0% duty cycle) giving an output across the electromagnet of 12V. This is the result expected to be observed as this indicates the electromagnet is at full capacity magnetism, and the system would be picking up the load. When applying the 5V PWM signal at 100% duty cycle the voltage across the electromagnet was 0V, thus indicating the circuitry of the electromagnet is operational. It is important to observe that the magnetic force of the electromagnet is not a binary condition of on/off, the simulation evaluated the voltage functionality at minimum and maximum voltage across.



Figure 7: Simulink Simulation - PWM input for the electromagnet



Figure 8: Simulink Simulation - Voltage across Electromagnet

## 4.2 Motor

The graphs below (figure 9,10) shows the Voltage output of the motor with respect to the PWM input provided. The PWM2 signal is the PWM input of the DC motor at a frequency of 500Hz with a duty cycle of 80%. This gives an output of 9.3V across the motor in simulation. The motor is not connected to any load to do the free load analysis.

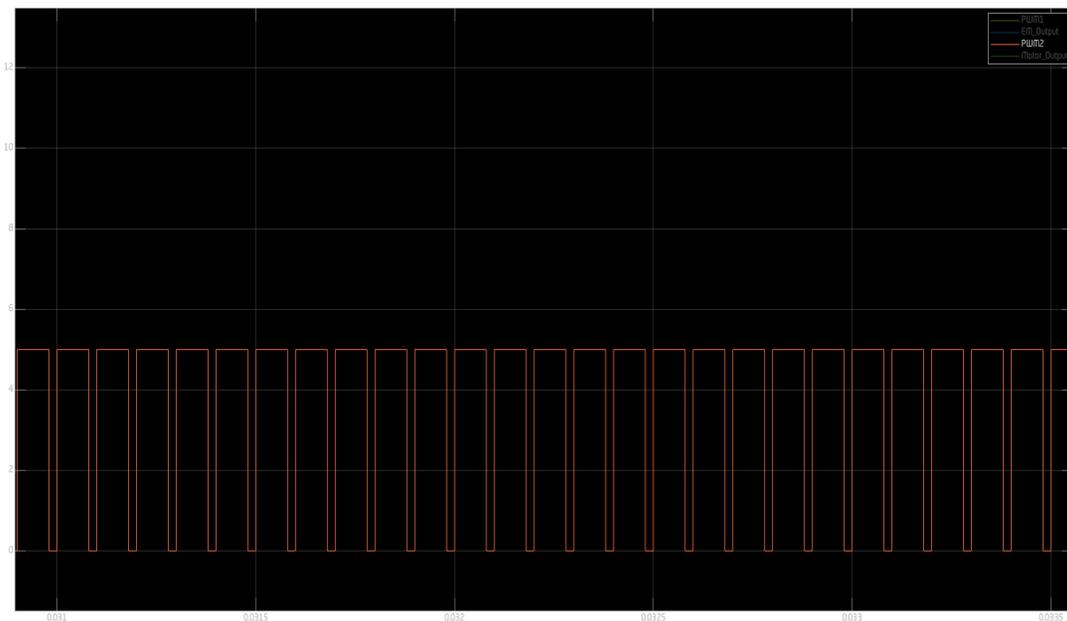


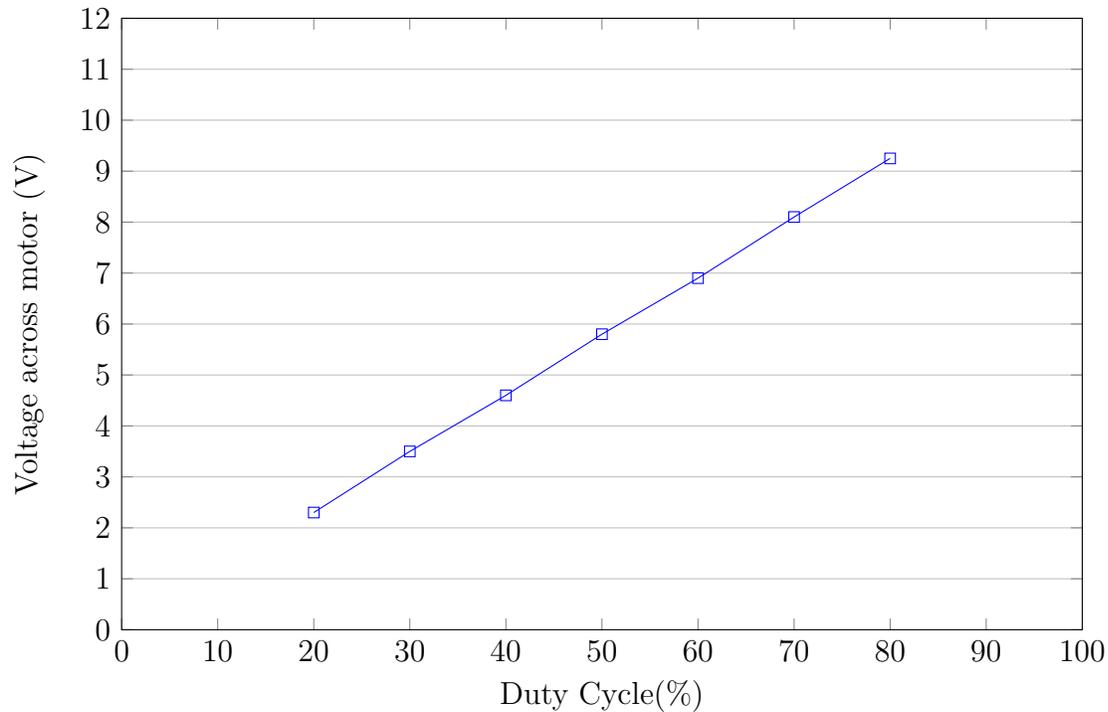
Figure 9: Simulink Simulation - 80% PWM input for the motor



Figure 10: Simulink Simulation - Voltage across Motor

The figure below shows the voltage received by the motor at different duty cycles. In order to compare these results with the experimental values, the data is collected and analyzed from 20% to 80% duty cycle since the measured values were limited using the functional generator. From the simulation, we see that the characteristic curve is linear. This is comprehensible because there is no friction, wear and tear of the gear box etc. that is accounted for in the simulation.

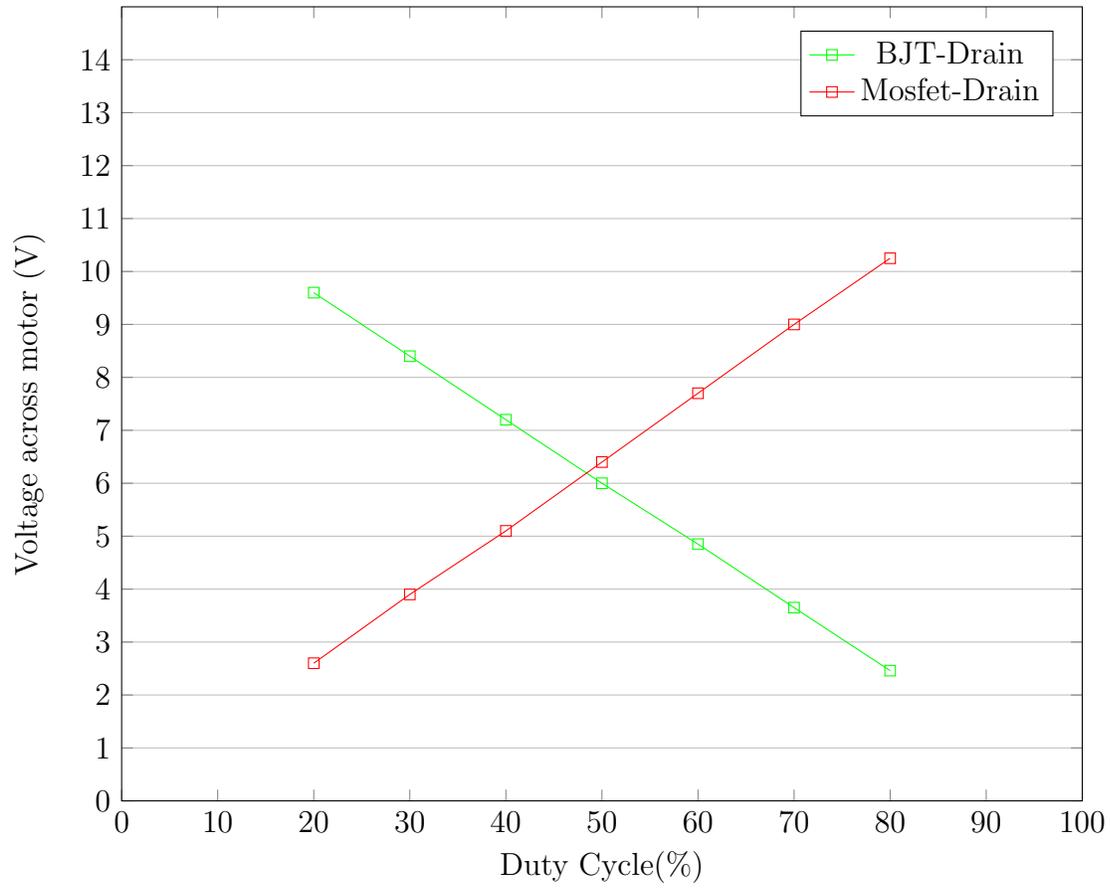
Voltage across motor vs Duty Cycle in simulation



### 4.3 Power stage simulation

The graph below shows voltages at different nodes of the electrical system. We observe that the voltage at the Drain node of the Mosfet is increasing as the duty cycle increases and this is also connected to one end of the electromagnet output. The other end of the electromagnet output is connected to a 12V input. Hence, decreasing the voltage difference across the electromagnet output as the duty cycle increases making it an inverted circuit. The voltage at the Drain node of the BJT is decreasing because most of it is being consumed on the Mosfet in the circuitry design.

Voltage across motor vs Duty Cycle (Simulated)



# 5 Experimental Results

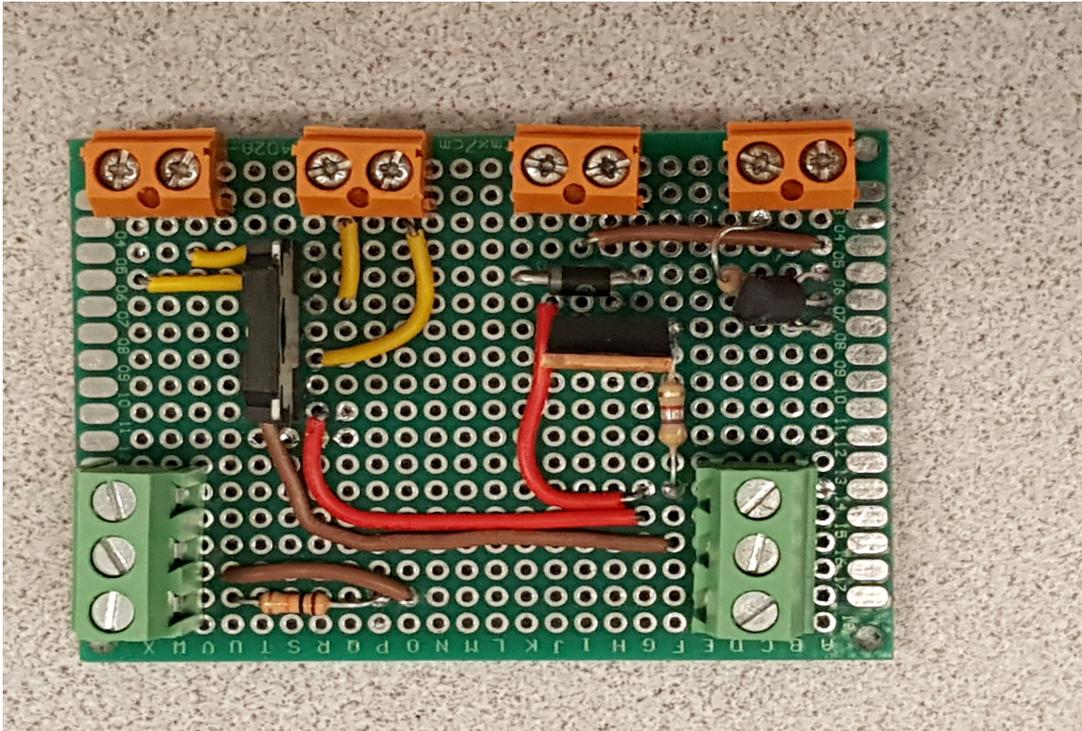


Figure 11: Electrical Circuit Board

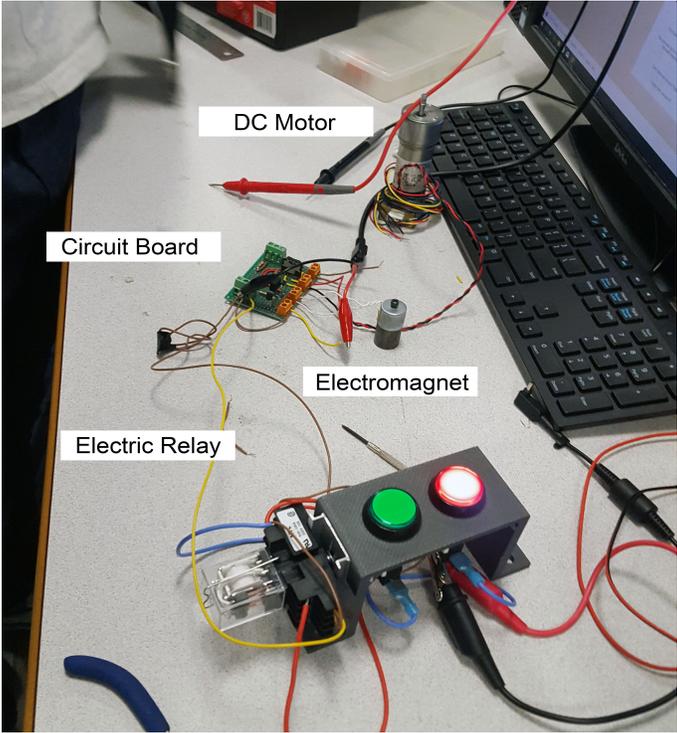
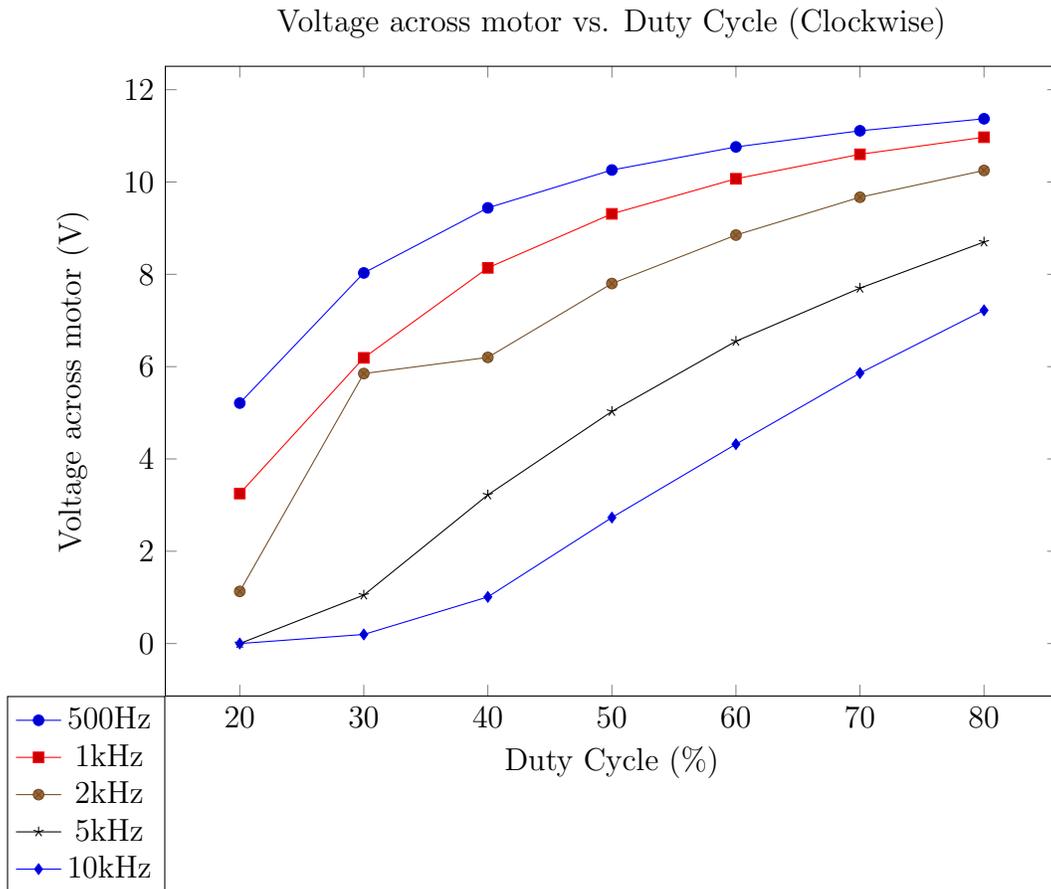


Figure 12: Electrical Circuit Board connected with relay protection and drive components

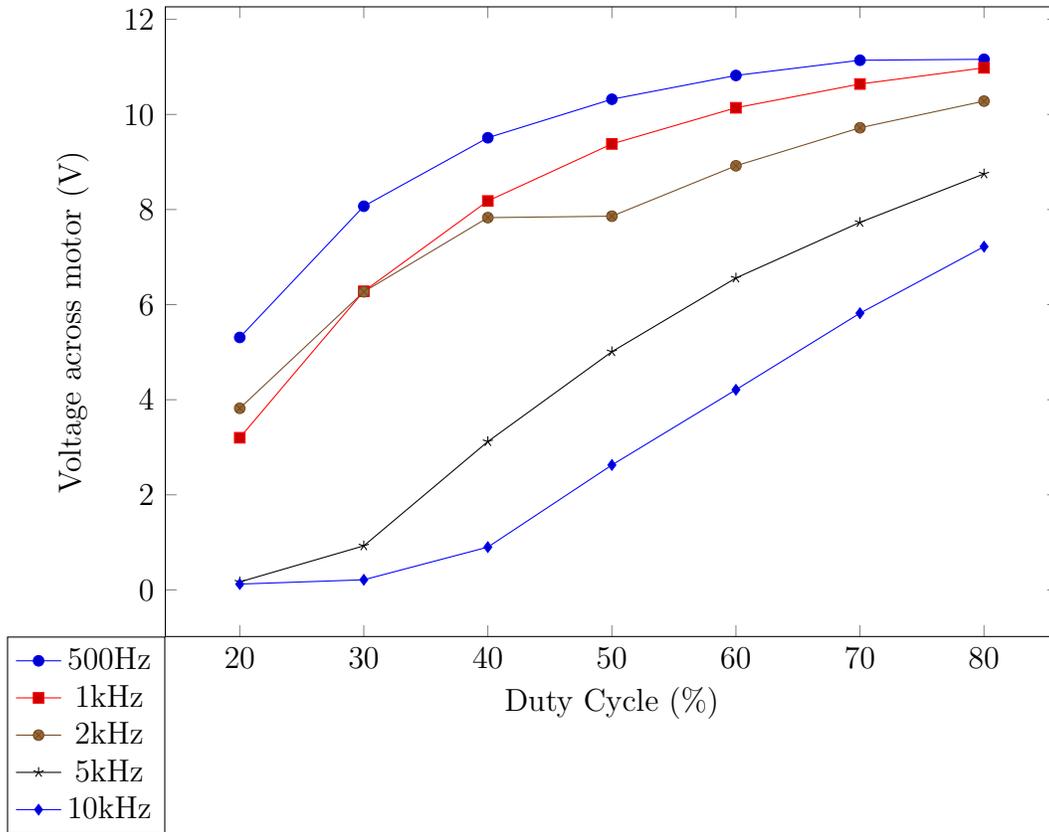
## 5.1 DC Motor Characterization

In order to observe the specific operational motor characteristics, we measure the output voltage across the motor terminals when the duty cycle is varied using the lab provided signal generator. Since the motor PWM signal will actually be provided with an Arduino micro-controller, we will be able to have 0 to 100% duty cycle available however, for this characterization testing using the lab signal generator we are limited to varying the duty signal of the control waveform between 20 - 80%. We vary the frequency of the signal to determine optimal operational frequency for the motor (500Hz, 1kHz, 2kHz, 5kHz, 10kHz).

Observed in the plots depicting the voltage across the motor vs. duty cycle (for both CW and CCW rotation), we observed that the highest voltage output is observed when the frequency was set at 500Hz.



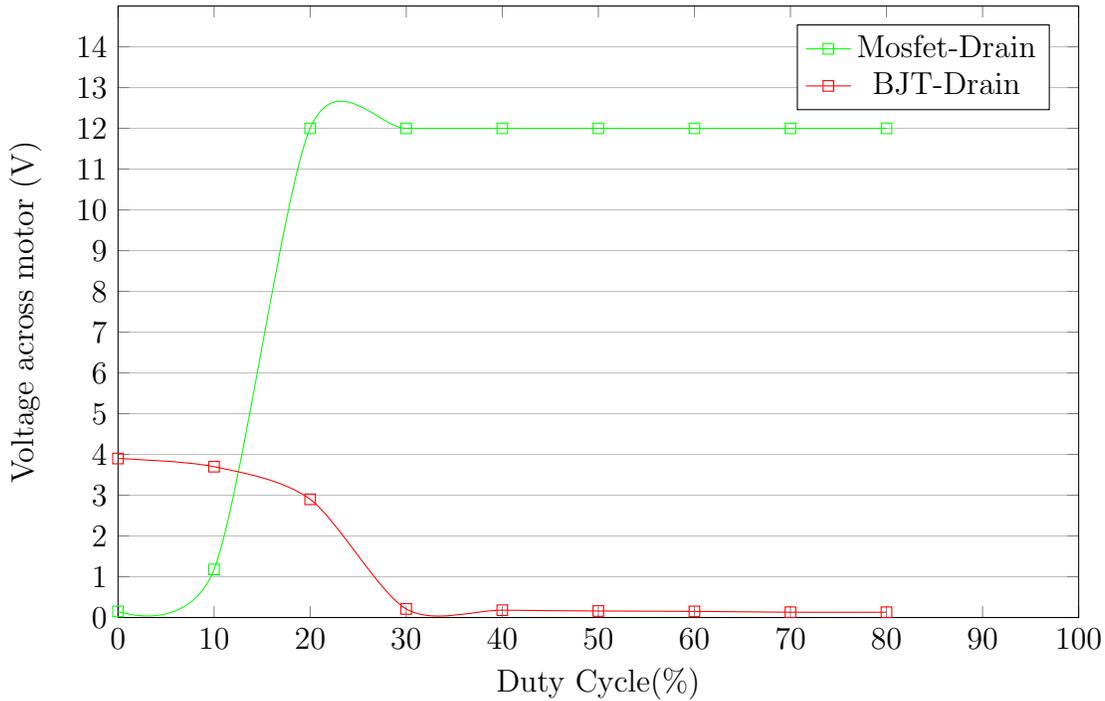
Voltage across motor vs. Duty Cycle (Counterclockwise)



In comparison to the simulated data, the experimental values are non-linear which is due to friction, gear backlash, electrical component non-linearities, etc. Also, the setup in the Simulink might not represent the actual limitations as the simulation assumes an ideal behaviour. Furthermore, we assumed values from the datasheet to be exact. However, there may be variances in the manufacturing of the various components.

## 5.2 Transfer Characteristics of Power Stage

Voltage across motor vs Duty Cycle (Experimental)



As we can see from the graph observing the BJT drain plot, 20% to 30% is the minimum duty cycle required to stop the electromagnet from having magnetic pull. At 30% duty cycle, the node at BJT drain goes to 0V as it gets deactivated. This makes the voltage drop to go from the mosfet transistor and makes the voltage at the other end 12V therefore there is 0V across the electromagnet output. The plot also indicates that significant voltage drops across the electromagnet between 10% to 20% duty cycle which resonates with the characteristics of electromagnet indicated on the data sheet (figure 3).

## 6 Conclusion

In conclusion, using applicable electrical drive components and theoretical knowledge we are able to utilize PWM control signals to operate and control the voltage output provided to the electromagnet and motor separately. This was a requirement in order to use micro-controller (Arduino) output to integrate the pick and place arm system. After modelling the system on the breadboard we were able to integrate the electrical circuitry onto a board, and implement circuit protection via relay and NO/NC push buttons which will also integrate the limit and micro switches of the arm platform.

Through simulation and experimentation, we see that we get obtain a acceptable response for electromagnet and DC motor. The electromagnet response curve matched the data sheet provided figure (3), and the characteristic operation of the DC motor at varying frequencies and duty cycles was observed.

With the observed characteristics of the circuit components and built electric drive of the system, the project can now progress into characterizing the transfer function of the system with the component characteristics and identify control parameters for the ideal system response in order to meet the project target: moving the load from position A to B in the fastest time possible.

A video of the system operating is found at this link:

<https://www.youtube.com/watch?v=iQn-vked-jEfeature=youtu.be>

### 13.3 Control Report

MSE 312  
Controls Report

Anjandev Momi - 301281186  
Brindan Ramalingam - 301271567  
Neeraj Bansal - 301323495

August 2, 2019

# Contents

1	Introduction . . . . .	3
2	Detailed Analysis and Design of the System Involved . . . . .	4
	2.1 Development of Transfer Function . . . . .	4
	2.2 Control design considerations . . . . .	7
3	Test procedure and simulation results . . . . .	7
	3.1 P Controller . . . . .	7
	3.2 PI Controller . . . . .	11
	3.3 PD Controller . . . . .	12
	3.4 PID Controller . . . . .	18
4	Simulation results including system parameters and non-linearities . . . . .	18
	4.1 System Parameter Selection . . . . .	20
5	Performance summary of designed system, meeting design requirements, and resolutions to problems encountered . . . . .	21
6	Conclusions . . . . .	22
7	Code and simulation files as an attachment . . . . .	22

# List of Figures

1	Circuit Schematic of motor drive circuit . . . . .	4
2	Signal flow diagram with disturbance . . . . .	6
3	Closed loop signal flow diagram . . . . .	6
4	Signal diagram of P controller . . . . .	7
5	Root locus of the P controller . . . . .	8
6	Root locus of the P controller zoomed into imaginary axis . . . . .	9
7	Matlab PID Tuner plot for P controller at $K_p$ value of 98.319. . . . .	10
8	Matlab PID Tuner plot for P controller at $K_p$ value of 0.2748 . . . . .	10
9	Matlab PID Tuner plot for P controller at $K_p$ value of 0.0043291 . . . . .	11
10	Root locus of the PD controller when $\frac{K_P}{K_D} > 235$ . . . . .	13
11	Root locus of the PD controller when $\frac{K_P}{K_D} = 50$ . Search region for the system solution is on the real axis shown in yellow . . . . .	13
12	Root locus of the PD controller when $\frac{K_P}{K_D} = 200$ . Search region is on the real shown in yellow . . . . .	14
13	Zoomed in root locus of the PD controller when the control values are $K_D$ is 4.2648 and $K_P$ is 1.1391 ( $4.2648(s + 0.2671)$ ) . . . . .	15
14	Zoomed out root locus of the PD controller when the control values are $K_D$ is 4.2648 and $K_P$ is 1.1391 ( $4.2648(s + 0.2671)$ ) . . . . .	16
15	Step response of the PD controller when the control values are $K_D$ is 4.2648 and $K_P$ is 1.1391 - ( $4.2648(s + 0.2671)$ ) . . . . .	17
16	Simulink Model of PID controller including non-linearities . . . . .	19
17	Simulink PD controller step response . . . . .	19
18	Simulink PID controller step response . . . . .	20

# 1 Introduction

MSE 312 term project requires teams to develop a one degree of freedom pick and place arm spanning 30cm to carry a payload from point A to B accurately in the fastest time. This system is electrically powered and controlled. The control will be provided from an Arduino micro-controller which will be provided in the integration section of the project. However, for the controls portion of the project, we look to analyze the system involved and develop a transfer function. With the transfer function, we initially assume a completely linear system and looked to pick the gain constants such that:

- Settling time is minimized
- Percent overshoot is within the maximum overshoot (5 cm maximum by project description)
- Rise time is minimized
- Phase margin and gain margin (stability) is maximized

In our linear analysis, we ignore nonlinearities such as: gear backlash, different motor characteristics when driving motor clockwise and driving motor counterclockwise, noise, etc. As such, after developing an appropriate control scheme using the linear system, we will use Simulink to simulate these nonlinearities and further improve our controller's design.

## 2 Detailed Analysis and Design of the System Involved

### 2.1 Development of Transfer Function

#### Electric Circuit

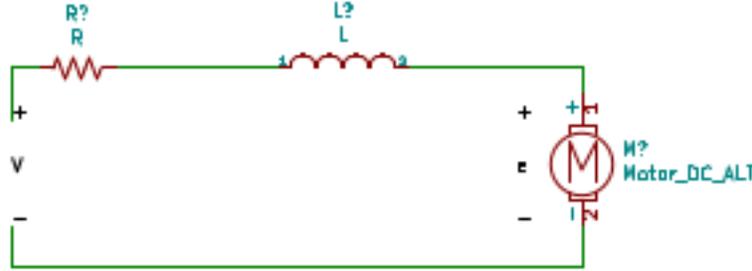


Figure 1: Circuit Schematic of motor drive circuit

The circuit powering the DC motor is shown in figure 1. The  $R$  is the resistance of the turned on mosfets and resistance of the inductance coil of the DC motor (from motor datasheet). The  $L$  is the inductance of the motor (from motor datasheet). In order to determine the transfer function of the electric circuit, we use KVL:

$$V = iR + L \frac{di}{dt} + e \quad (1)$$

The back-emf of the motor ( $e$  in the circuit diagram figure 1) can be determined by the following equation:

$$k_m n \omega = k_m n \dot{\Theta} = e = k_m \omega_m \quad (2)$$

$\omega_m$ : motor speed

$k_m$ : motor constant (from data sheet)

Using equations 1 and 2, we determine the electrical domain equation:

$$V = iR + L \frac{di}{dt} + k_m n \dot{\Theta} \quad (3)$$

#### Gear Reduction

$$\dot{\Theta} = \frac{\dot{\Theta}_m}{n} \quad (4)$$

$\dot{\Theta}$ : load side's speed

$\dot{\Theta}_m$ : motor's speed

$n$ : gear ratio

## Mechanical Transfer Function

In order to develop the mechanical portion of the transfer function, we apply newton's second law to the system:

$$\Sigma\tau = J\alpha = \tau - \tau_f = J\ddot{\Theta} \quad (5)$$

Friction ( $\tau_f$ ) in equation 5 can be expressed as the following equation:

$$\tau_f = B\dot{\Theta} + \tau_s \quad (6)$$

$B\dot{\Theta}$ : viscous damping

$\tau_s$ : static friction

As such, equation 5 and 6 become:

$$\tau - \tau_s - B\dot{\Theta} = J\ddot{\Theta} \quad (7)$$

In order to relate the mechanical domain and the electrical domain, we use equation 8.

$$\tau = \eta k_m i n \quad (8)$$

$\tau$ : torque out of gear box

$\eta$ : efficiency of gearing

As such, our final mechanical equation becomes:

$$\eta n k_m i - \tau_s - B\dot{\Theta} = J\ddot{\Theta} \quad (9)$$

## Laplace and Signal Flow

Taking the Laplace transform of the electric domain equation (equation 3), we get:

$$V(s) = LsI(s) + RI(s) + k_m n s \Theta(s) \quad (10)$$

Isolating  $I(s)$ :

$$I(s) = \frac{V(s) - k_m n s \Theta(s)}{Ls + R} \quad (11)$$

Taking the Laplace transform of the mechanical domain equation (equation 9), we get:

$$\eta n k_m I(s) - \tau_s(s) - Bs\Theta(s) = Js^2\Theta(s) \quad (12)$$

Isolating  $\Theta(s)$ :

$$\Theta(s) = \frac{\eta n k_m I(s) - \tau_s(s)}{s(Js + B)} \quad (13)$$

Our signal flow diagram is shown in figure 2.

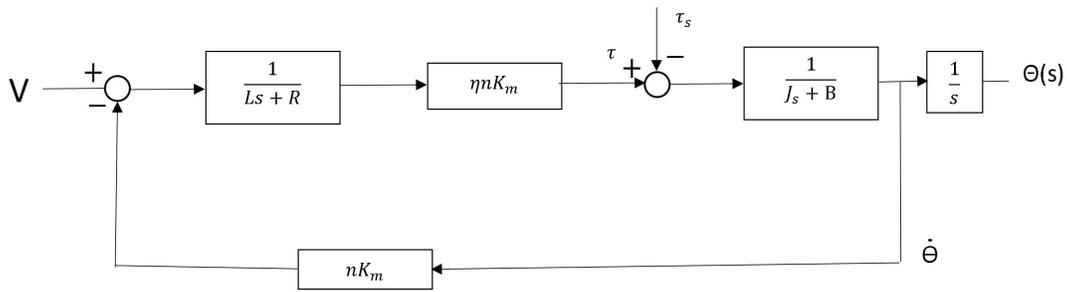


Figure 2: Signal flow diagram with disturbance

If we assume no disturbance ( $\tau_s = 0$ ), we can get a simple input/output transfer signal flow diagram that we can easily design a controller for. The signal flow diagram for the closed loop system is shown in figure 3.

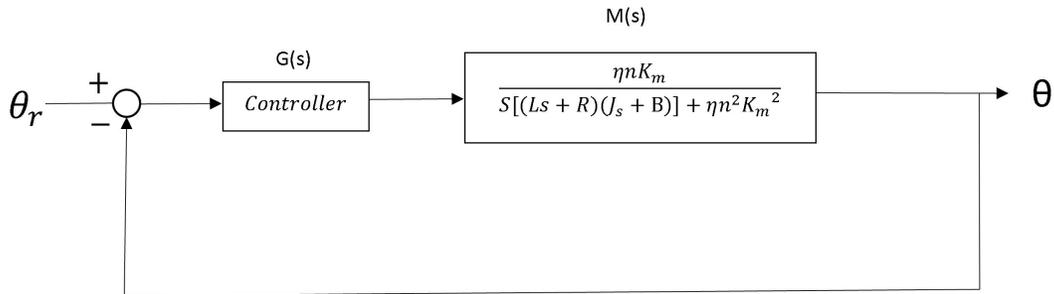


Figure 3: Closed loop signal flow diagram

$M(s)$  in figure 3 is the plant we will be designing a controller for in section 3. This will allow us to generate the initial design values for the PID controller that we will fine tune in the simulation portion of the controls report. In the simulation portion of the report (section 4), We will be using the signal flow diagram shown in figure 2 as the plant to design the system.

## 2.2 Control design considerations

To develop an appropriate controller for the developed system model, we define targets of the controller:

- Overshoot < 5cm.  $\theta = \arccos\left(\frac{(a)^2+(b)^2-(c)^2}{2*a*b}\right) = \arccos\left(\frac{(30)^2+(30)^2-(5)^2}{2*30*30}\right) = 0.16686$  rad
- Moving truss to 180 degrees:  
Percent Overshoot =  $\frac{0.16686}{\pi} = 5.31\%$
- Moving truss to 90 degrees:  
Percent Overshoot =  $\frac{0.16686}{\frac{\pi}{2}} = 10.62\%$
- Considering the inertia of the system changes when the system picks up the load (45g metal mass) we will have to design PID control for the clockwise and counter-clockwise directions to simulate and operate the system to be optimal.
- limit  $K_d$  to limit stress on actuator

## 3 Test procedure and simulation results

In order to design an appropriate controller to operate the pick and place arm, we will evaluate the system considering P, PI, PD and PID controller schemes to evaluate optimal speed of the system while also being stable and robust in nature.

### 3.1 P Controller

$$\frac{\eta * n * km(K_P)}{(L * J) * s^3 + (B * L + R * J) * s^2 + (nu * n^2 * km^2 + R * B) * s} \quad (14)$$

Substituting the selected system parameters into the found transfer function of the system, we get a numeric transfer function as follows:

$$H(s) = \frac{0.06284}{1.627 * 10^{-5}s^3 + 0.03393s^2 + 0.004164s} \quad (15)$$

We plot the root locus of the characteristic equation of the following system:

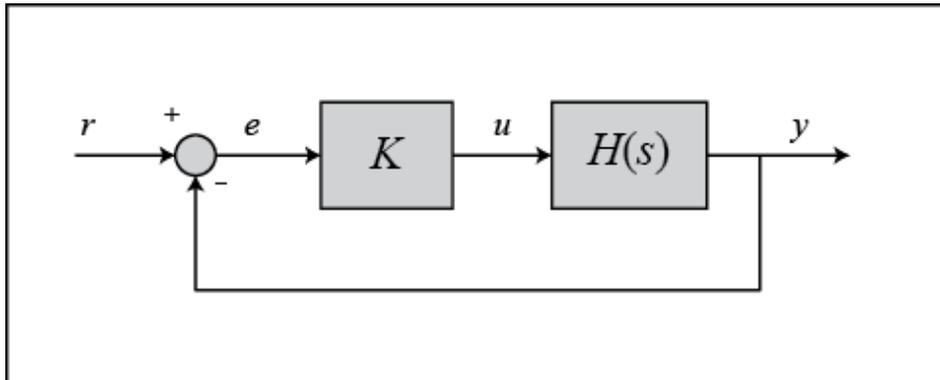


Figure 4: Signal diagram of P controller

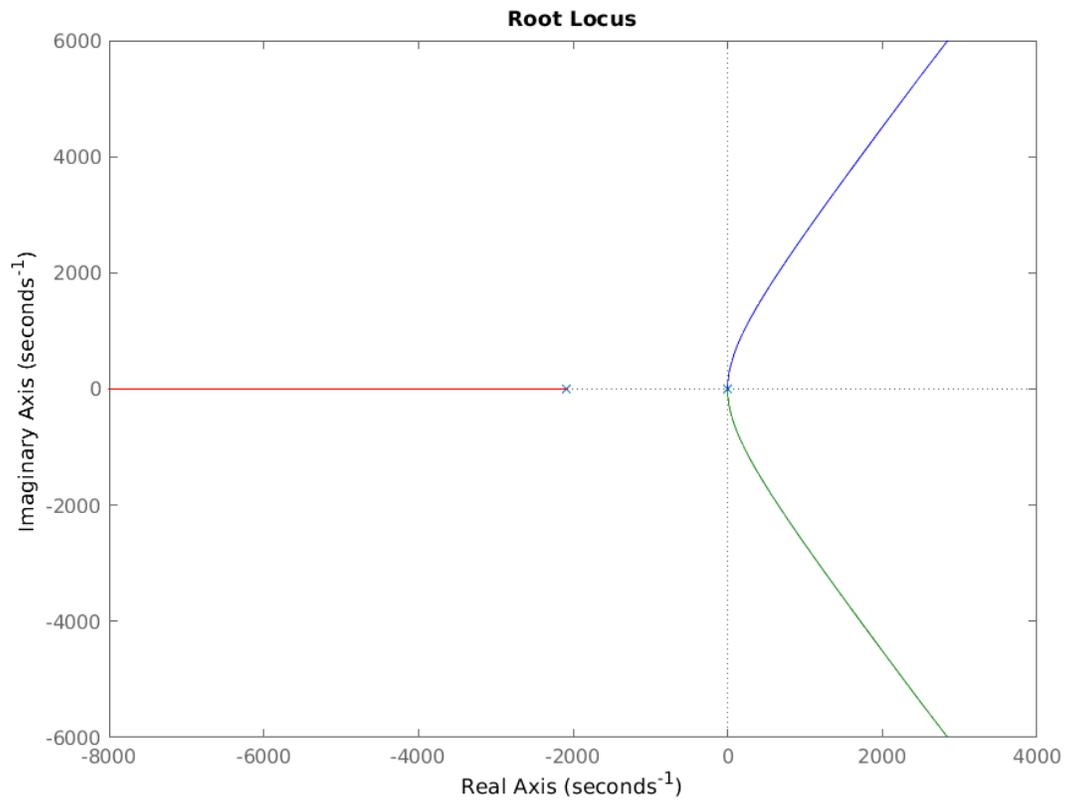


Figure 5: Root locus of the P controller

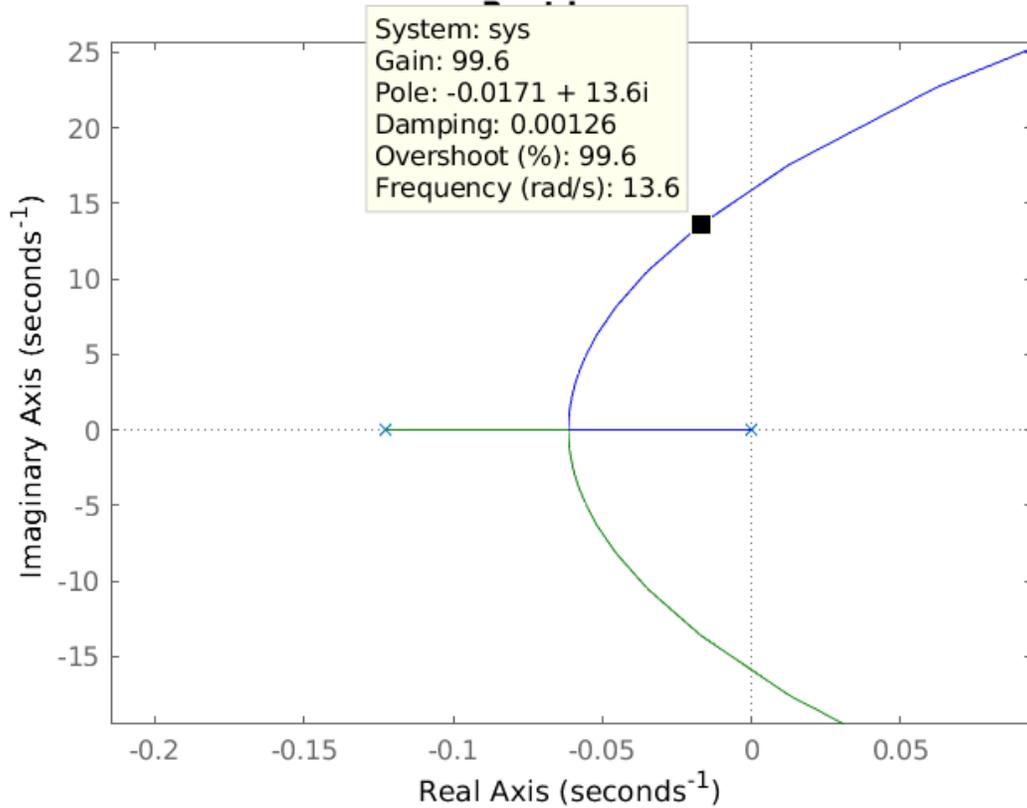


Figure 6: Root locus of the P controller zoomed into imaginary axis

As shown in figure 6, any value of  $K_p > 99$  places the two roots of the characteristic equation to the right of the imaginary axis. As such, any  $K_p > 99$  makes the entire system unstable. Therefore, our search region for the P controller is  $0 < K_p < 99$ .

### Matlab PID Tuner Plots

With the defined region of  $K_p$ , we can now use the PID tuner function of Matlab to evaluate the systems rising time, settling time and overshoot characteristics to choose an appropriate  $K_p$  value:

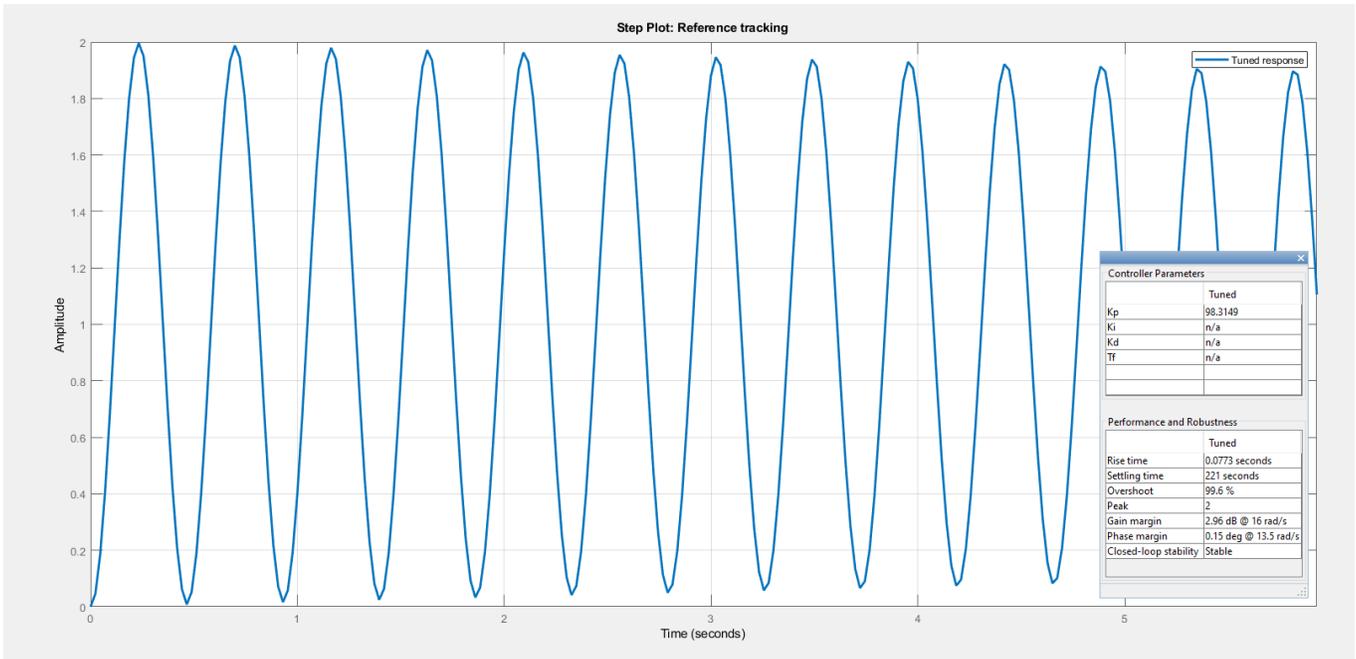


Figure 7: Matlab PID Tuner plot for P controller at  $K_p$  value of 98.319.

Observable characteristics of the system response shows the settling time is 221 seconds, very quick rise time of 0.0773, gain margin of 2.96 dB, phase margin of 0.15 deg. The  $K_p$  value of 98.319 is very close to the upper bound value of 99. This response of the system has significant back and forth fluctuation in the system almost at the brink of instability. The overshoot of the system (99.6%) is beyond the threshold of the system, therefore this is not a valid  $K_p$  for the system.

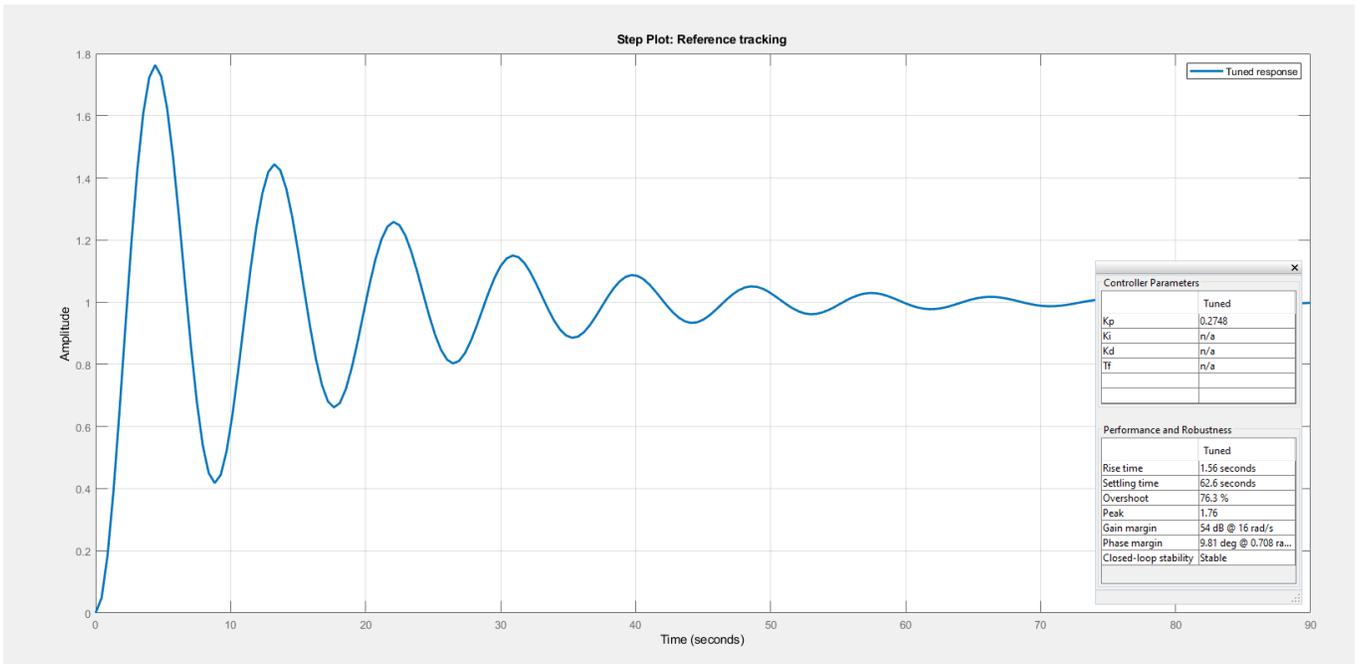


Figure 8: Matlab PID Tuner plot for P controller at  $K_p$  value of 0.2748

Observable characteristics of the system response shows the settling time is 62.6 seconds, Rise time of 1.56 seconds, gain margin of 54 dB @ 16 rad/s, phase margin of 9.81 deg. This response of

the system has a long settling time, and high overshoot passed the bounds calculated in the allowable limit (5.31%)

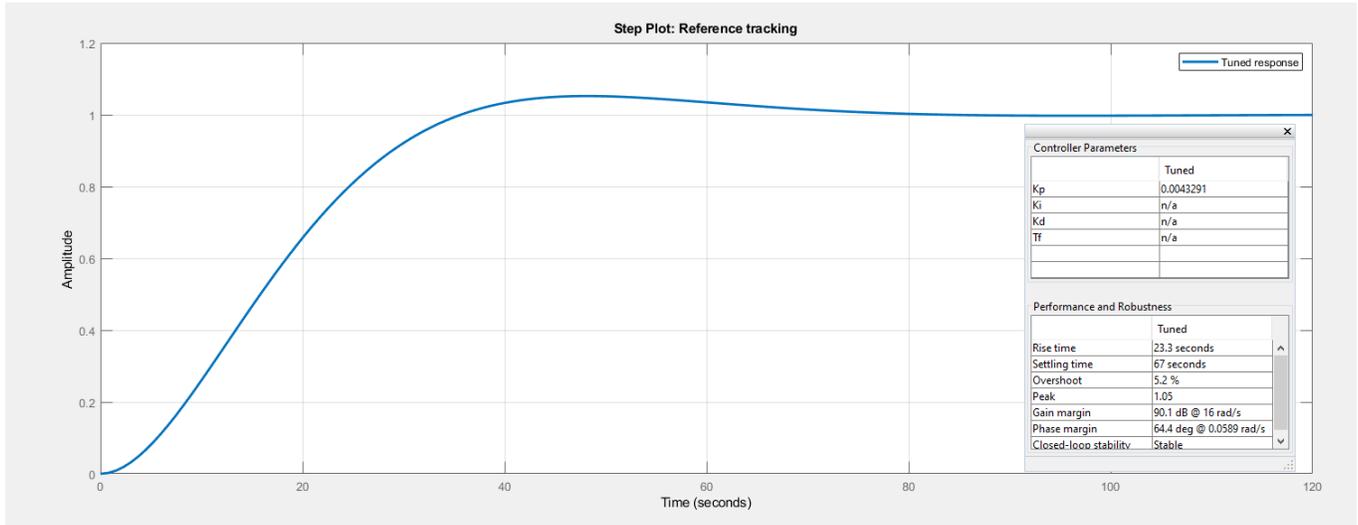


Figure 9: Matlab PID Tuner plot for P controller at  $K_p$  value of 0.0043291

This system response is valid which has a overshoot value of 5.2% which is very close to the calculated margin limit of 5.31%. The system response shows the settling time is 67 seconds, Rise time of 23.3 seconds, gain margin of 90.1 dB @ 16 rad/s, phase margin of 64.4 deg @ 0.0589 rad/s. This system has a very slow rise time and since we are trying to move the system to pick up the load in fast as possible manner, these system characteristics are not optimal.

Notable with proportional control, as we move the  $K_p$  value up the system rise time decreases however stability of the system decreases which coincides with the plotted root locus (figure 6). We observe with figure 9 adding proportional gain to the system does not have the appropriate system response to meet the requirements of the project targets. Since we observed that the with a  $K_p$  values close to the upper margin of the root locus had quicker rising time, but settling time was a significant issue we require damping, therefore we proceed further to evaluate the system using other forms of control.

### 3.2 PI Controller

The PI controller was not considered because as it is shown in P controller design, the main issue with our system is the lack of damping (when we looked at system responses with quick rise time). This lack of damping causes the the  $\omega_n$  to increase quickly as a result of increasing the gain. The rapid  $\omega_n$  increase causes the overshoot of our system increase significantly. The differential controller can be seen as the damper and it is advisable to use PD instead of the PI controller.

Another perspective is in the root locus for the P controller (figure 5), when  $k > 99$  the system is unstable. The vast majority of possible designs with the proportional controller lie on the right side of the imaginary axis. In order to increase number of designs it is advisable that we place a zero left of the imaginary axis such that when k increases, some of the poles will approach our zero rather than crossing the imaginary plane. In order to create a zero, we need a PD controller.

It is true that the integral component will compensate for the disturbance  $\tau_s$ . However, this is not our main issue currently. We will first ensure the transience of the system is adequate and then work to ensure the disturbance does not cause error.

### 3.3 PD Controller

In order to increase the number of designs we could use for our design, we look to employ a PD controller. A PD controller increases the number of possible designs because we are introducing a zero into our system. The greater possibilities and potential for the system comes from the poles crossing the imaginary axis will approach the newly added zero instead and as a result we will have more possible designs. In the solution for the system we cannot use the value of  $K_P$  found in the P controller section (section 3.1) because both  $K_P$  and  $K_D$  affect the transient response of the system. As such, we will develop a separate design procedure for the PD controller.

If we choose  $G_c = K_D s + K_P$ , the forward gain becomes:

$$\frac{\eta * n * km(K_D s + K_P)}{(L * J) * s^3 + (B * L + R * J) * s^2 + (nu * n^2 * km^2 + R * B) * s} \quad (16)$$

$$\frac{\eta * n * km(s + \frac{K_P}{K_D})K_D}{(L * J) * s^3 + (B * L + R * J) * s^2 + (nu * n^2 * km^2 + R * B) * s} \quad (17)$$

We can rewrite equation 16 as equation 17 where the value of  $\frac{K_P}{K_D}$  will determine the location of the zero.  $\frac{K_P}{K_D}$  will be selected and then we will use the root locus to determine the value of  $K_D$ .

When searching for a system solution on the root locus with the added zero, we search on the real axis in order to ensure overshoot is 0%.

#### Location of Zero

We want the zero to be on the left side of the imaginary axis such that the poles near the imaginary axis go to the zero we create. As a result we have a larger number of designs to choose from. As you increase  $\frac{K_P}{K_D}$ , you move the zero further to the left and the circle becomes larger.

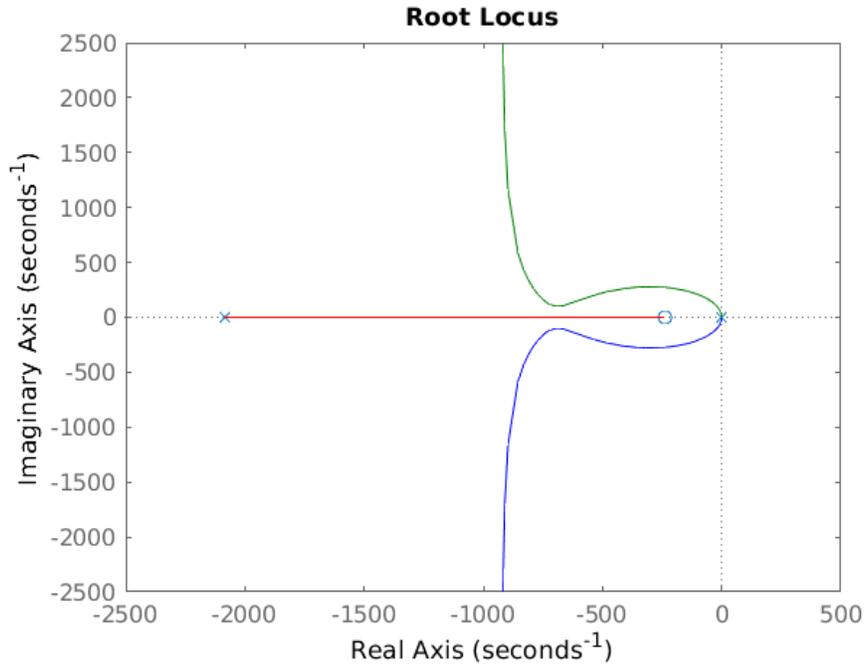


Figure 10: Root locus of the PD controller when  $\frac{K_P}{K_D} > 235$

When  $\frac{K_P}{K_D} > 235$ , the root locus looks like figure 10. When  $\frac{K_P}{K_D} = 235$ , we need  $K_D = 233$  to get fit the overshoot constraints we have. This gain is too large and our search region for  $0 < \frac{K_P}{K_D} < 235$ .

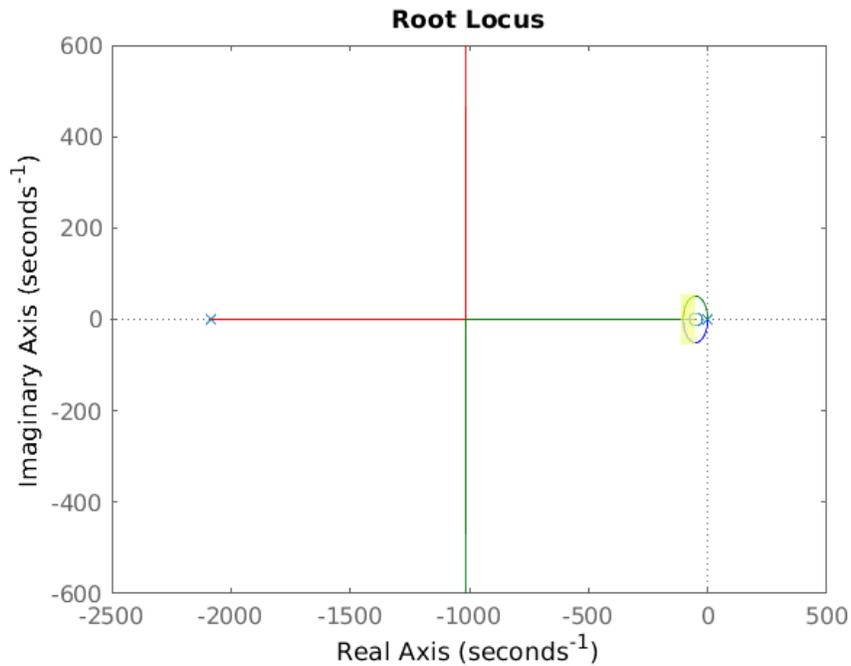


Figure 11: Root locus of the PD controller when  $\frac{K_P}{K_D} = 50$ . Search region for the system solution is on the real axis shown in yellow

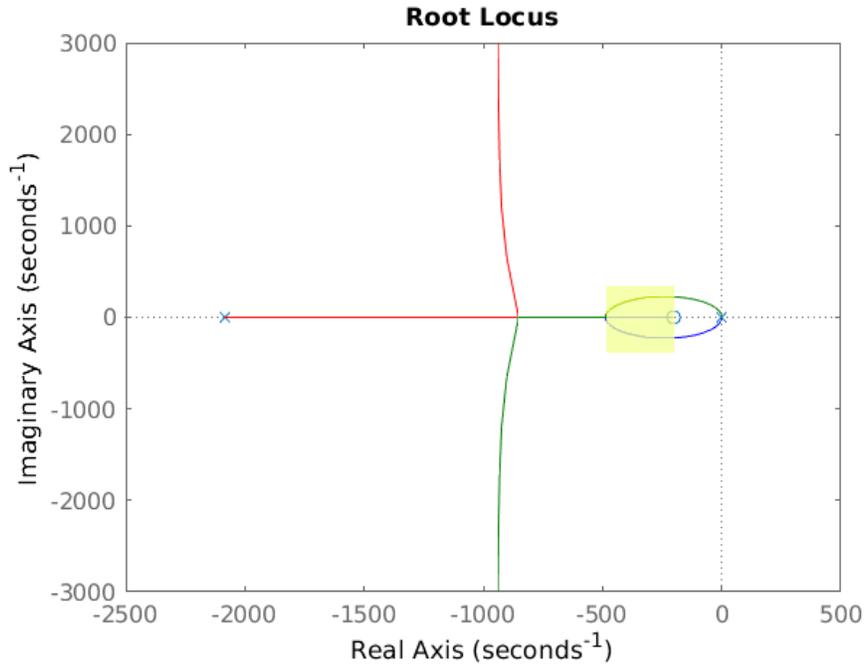


Figure 12: Root locus of the PD controller when  $\frac{K_P}{K_D} = 200$ . Search region is on the real shown in yellow

When we did a sweep from  $235 > \frac{K_P}{K_D} > 0$  on varying root locus models, any values on the circle gave us excessive overshoot. As the circle becomes larger (by increasing  $\frac{K_P}{K_D}$ ), we observe the following changes to our design:

- the gains (both  $K_d$  and  $K_P$  increase)
- our search region is further away from the imaginary axis. As such, our designs are much more stable
- our search region increases (more possible good designs)
- rise time is slower
- settling time is slower

When we design our PD controller, we try to make the overshoot lower than 5.31% which is because when proceed to add a integrative component to the controller in the PID section, we know the integrative component will increase the overshoot of the system. Using Matlab's Control System Designer tool, we add the derivative zero to the system, and adjust the location of the zero such that we find the appropriate metric response in the system. An appropriate solution is shown below:

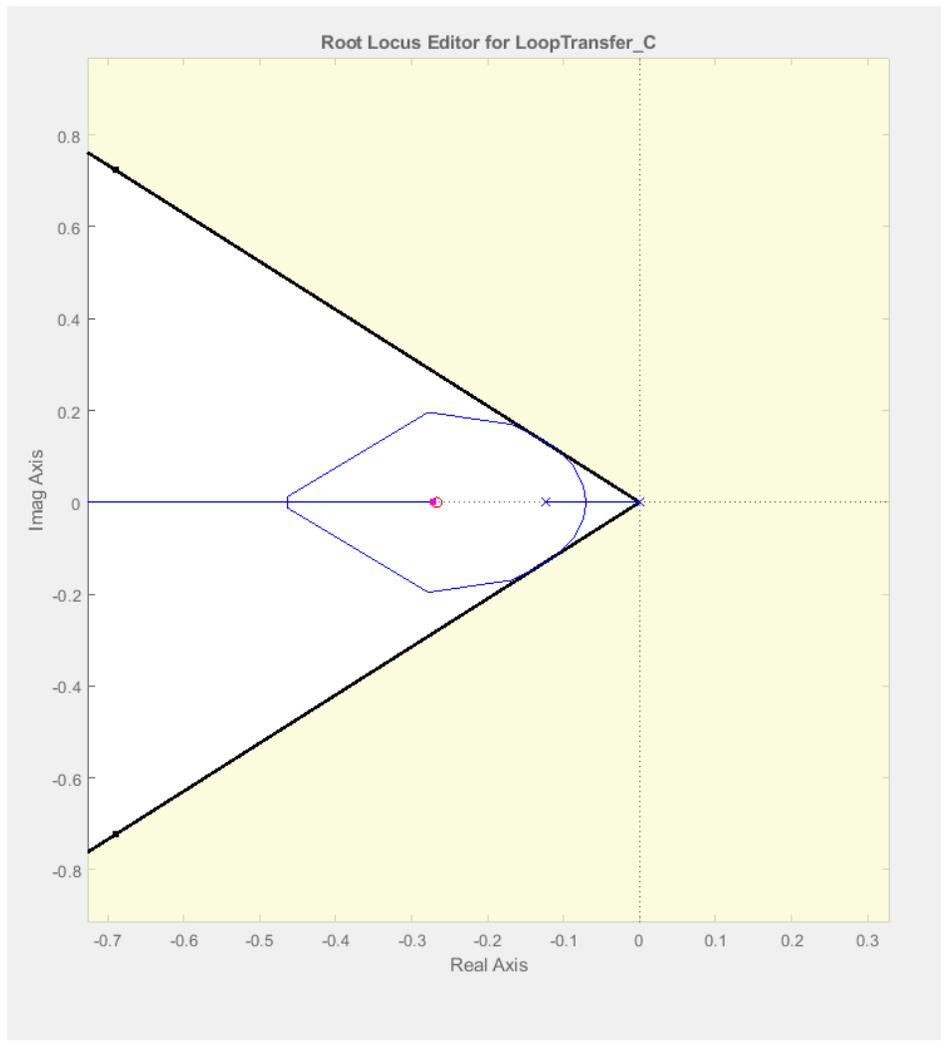


Figure 13: Zoomed in root locus of the PD controller when the control values are  $K_D$  is 4.2648 and  $K_P$  is 1.1391 ( $4.2648(s + 0.2671)$ )

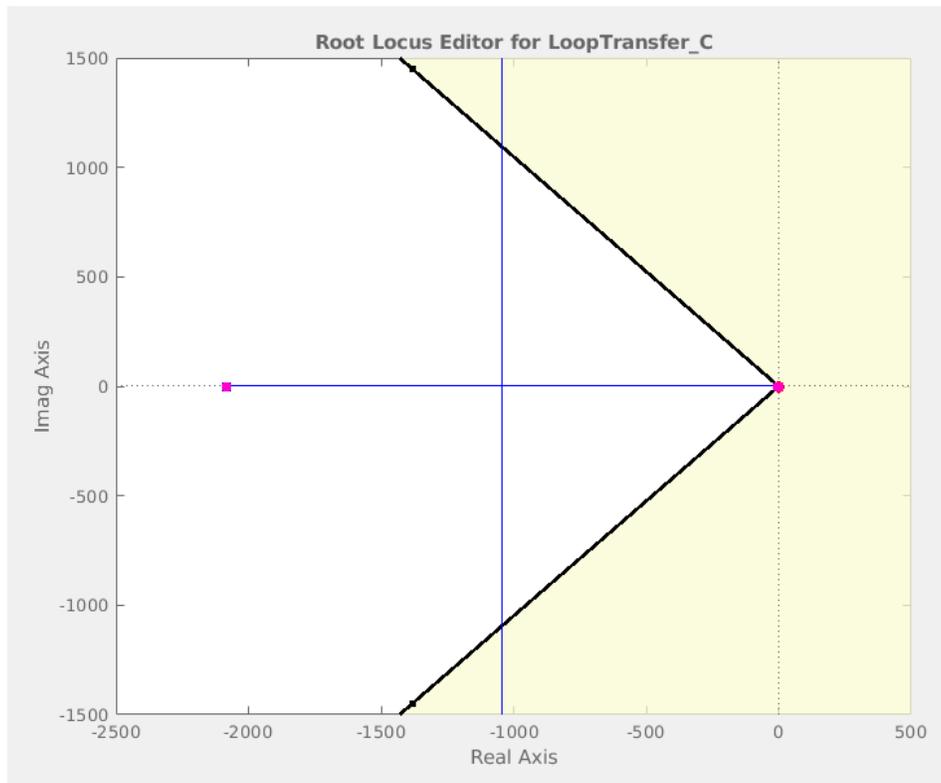


Figure 14: Zoomed out root locus of the PD controller when the control values are  $K_D$  is 4.2648 and  $K_P$  is 1.1391 ( $4.2648(s + 0.2671)$ )

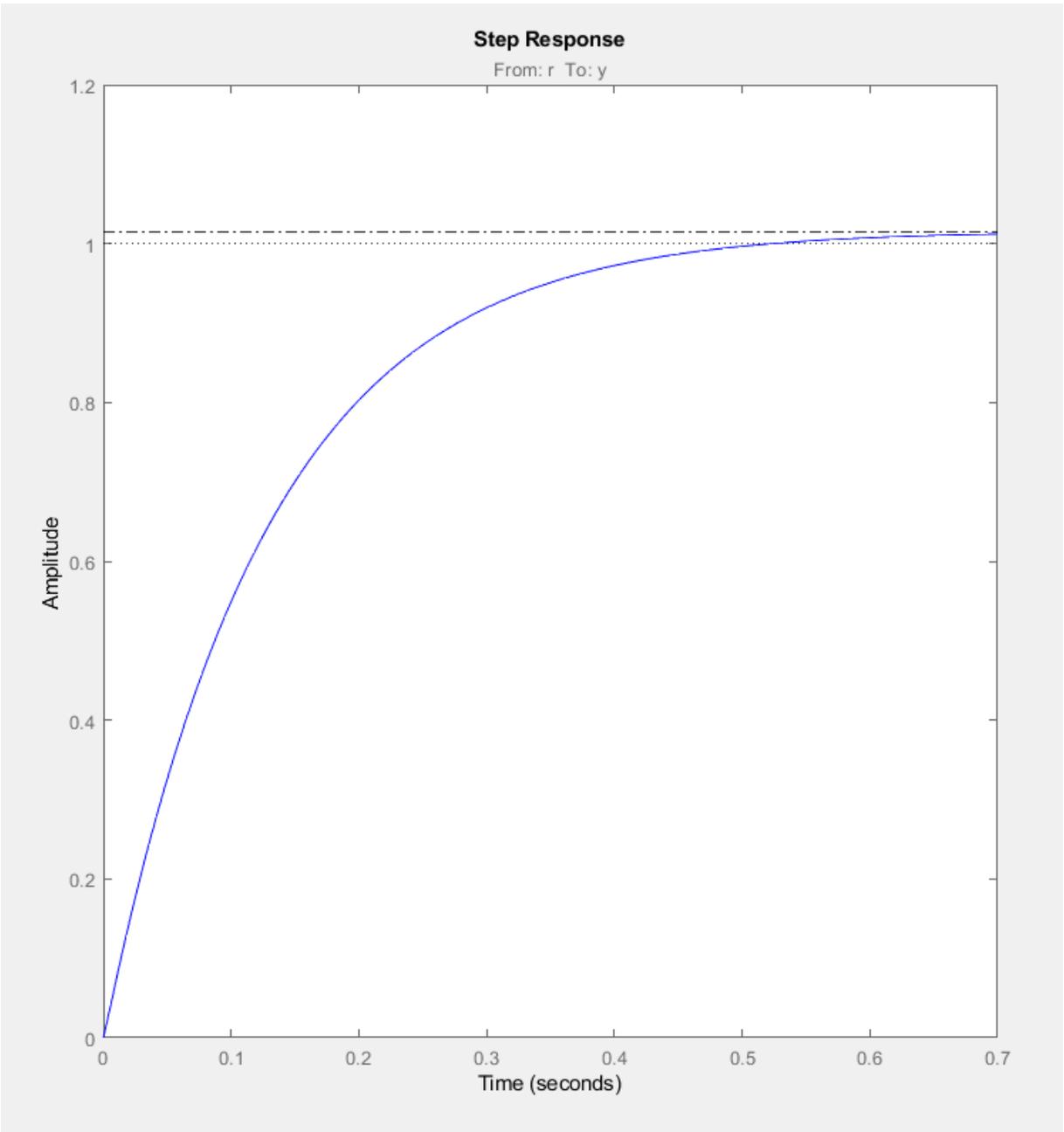


Figure 15: Step response of the PD controller when the control values are  $K_D$  is 4.2648 and  $K_P$  is 1.1391 -  $(4.2648(s + 0.2671))$

Through using a PD controller, we were able to develop a system controller which has appropriate linear response attributes desired for the system. We observe the overshoot is 0.37%, rise time is 0.217 seconds, settling time is 0.372 seconds. This system is very close to a critically damped system and has a reasonable slope curve to max amplitude. To proceed with the designer control we will evaluate the PID controller.

### 3.4 PID Controller

In finding a acceptable response to our system using the PD controller we still consider the PID controller because we know current simulations has only been completed analyzing a linear system in ideal conditions. The I of the PID controller is to control steady state error or compensate for disturbances. Our  $e_{ss}$  is already zero. Even though the disturbance  $\tau_s$  (static friction) does not show up in our linear transfer function, we would still like to introduce the  $K_i$  to compensate for static friction as it expected that  $\tau_s$  will cause significant error for our controller if left unchecked. The integral portion of the controller will also compensate for non-linearities such as gear backlash and the dead zone of the duty cycle signal to the motor. A drawback of the integral portion of the controller is that it increases the overshoot. As such, we want to tune  $K_I$  such that it does not affect the transience of the system.

This tuning is done through a trial and error method outlined in section 4 which will allow us to find a suitable PID controller including non-linearities in the system for more practical and realistic control.

## 4 Simulation results including system parameters and non-linearities

In order to evaluate the non-linearities of the system, we model the system using Simulink as shown below in figure 16.

To quantifiably observe the Simulink system with the PD controller values ( $K_D$  and  $K_P$ ) found after developing a PD controller, we simulate the Simulink model using only the PD controller values found in section 3.3. Next starting at  $K_I = 0$  and increasing  $K_I$  until our settling time is sufficiently short, we simulate the system to find a desired response in the system forming the PID controller. While settling time is a parameter of interest, we also want to ensure that our overshoot is  $< 5\%$ .

The non linearity parameters are as follows:

- Friction for motor:  $\tau_s = 2.5 * 10^{-3}$  Nm (Motor data sheet)
- Voltage deadband:  $-1.5 < V < 1.5$  V (Empirical testing with our circuitit. The motor does not move when the voltage across the motor is within this range)
- Gear deadband width: 0.5 (Empirical testing)

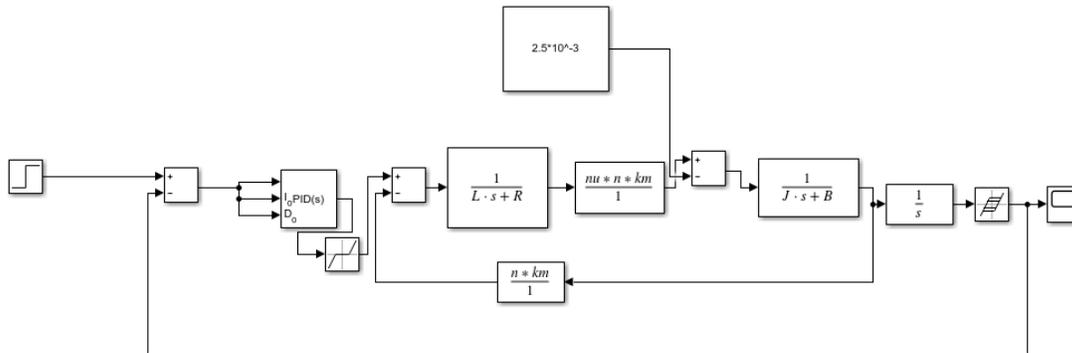


Figure 16: Simulink Model of PID controller including non-linearities

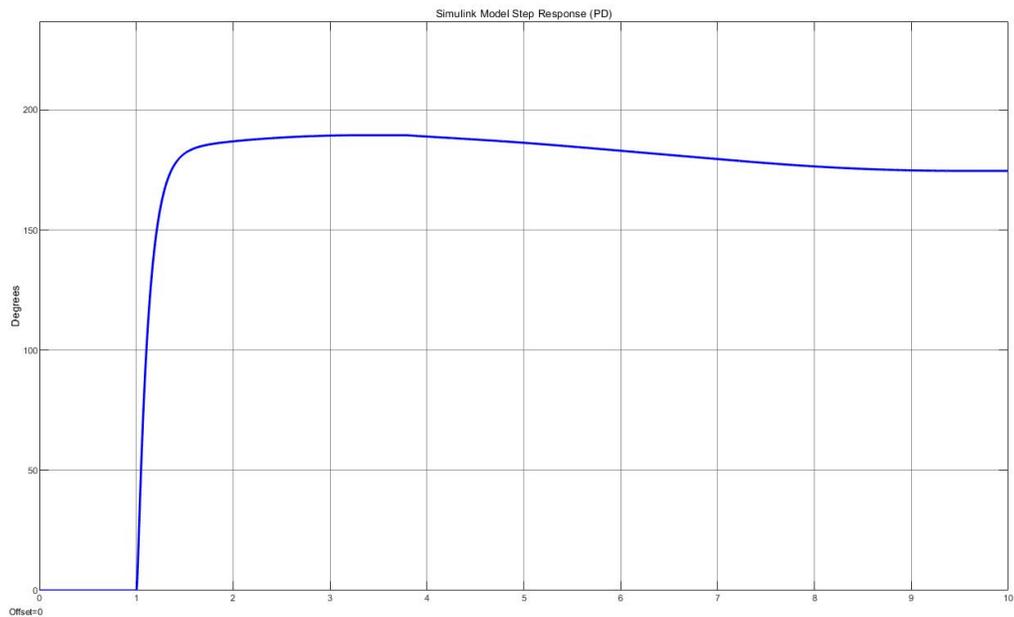


Figure 17: Simulink PD controller step response

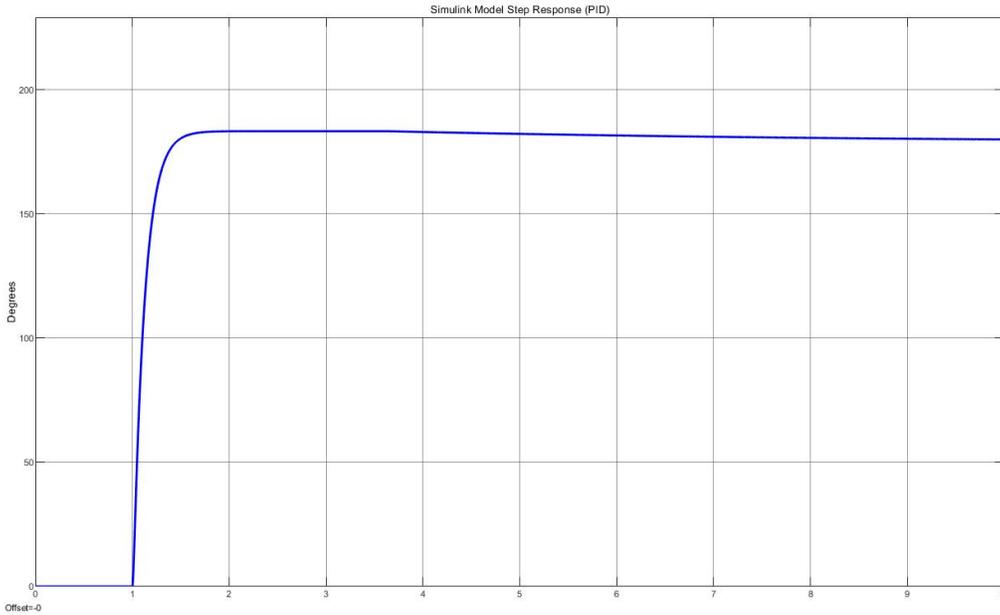


Figure 18: Simulink PID controller step response

Observing figure 17 which is the step response of the non linear system, we observe that there is a observable increase in transient error and overshoot due to the nonlinearities affecting the performance of the system. This also affects the settling time of the system. Although the rise time is still quick, the error of the system can be improved by adding  $K_I$  to the system which will form the PID controller.

Observing figure 18, simulating the model introducing  $K_I$  values into the system to help the system compensate the effects of the non-linearities of the system, we found that there was a threshold for the  $K_I$  value to aid the system before the  $K_I$  itself was contributing to the overshoot which we were limiting to meet the performance requirements. We can observe comparatively between figure 17 18 that with the PID control we have less error in the system, stability and less overshooting which will make timing the electromagnet operation and pick and place operation more accurate.

The final values of the PID controller are as follows:

- $K_P = 1.1391$
- $K_I = 0.1415$
- $K_D = 4.2648$

## 4.1 System Parameter Selection

In evaluating the  $J_{total}$  (inertia of the system), the  $J$  was summed to use the  $J$  of the mechanical arm, the brass mounting plates (arm mount, electromagnet mount), electromagnet, and aluminum cylindrical mounting attachment. Since the travel is  $180^\circ$  in order to pick up the pay load, the travel of the arm tip (radially) has a tighter limit of overshoot calculated, we modelled the system around the  $J_{total}$  to not include the electromagnet of the system.

Efficiency of the gearing, gear ratio, motor inductance, motor constant(Km) and viscous damping are taken from the datasheet of the motor. The resistance is taken from the motor and the

H-bridge combined as shown in the table. The other measurement were taken using a vernier caliper to get good accuracy on the length and diameter of different components of the system.

Parameter	Value
Efficiency of gearing	0.95 [unitless]
Gear Ratio	6.3 [unitless]
Mass of Electromagnet	23 [grams]
Radius of the magnet	19.1 [mm]
Mass of the Mount	49 [grams]
Radius of the Mount	18.5 [mm]
Motor constant (Km)	$1.05 * 10^{-2}$ [Nm/ $\sqrt{(W)}$ ]
Inductance	$2.34 * 10^{-3}$ [H]
Resistance	$4.33 + 0.55 = 4.88$ [Ohm]
Motor's viscous damping	$1.4 * 10^{-6}$ [Nms]
Inertia of the Structure	$5.11 * 10^{-3}$ [Kg m <sup>2</sup> ]
Inertia of the Electromagnet	$1.8 * 10^{-3}$ [Kg m <sup>2</sup> ]
Inertia of the Mount	$8.4 * 10^{-6}$ [Kg m <sup>2</sup> ]
Total Inertia	$7.0 * 10^{-3}$ [Kg m <sup>2</sup> ]

## 5 Performance summary of designed system, meeting design requirements, and resolutions to problems encountered

- Rise time = 0.263 s
- Settling time = 0.421 s
- Overshoot = 1.77%
- Gain margin = -74.4 dB @ 0.0201 rad/s
- Phase margin = 88.7 deg @ 7.9 rad/s

The design requirements of the system were met through implementation of the PID controller. The performance of the final system are indicated above and meets the requirements defined in the introduction of this report.

A problem which was encountered was simulating the system to develop a PD model to have high level of stability. We found that when moving the zero further away from the imaginary axis, the system response time was decreases (as desired) however stability in the system was also decreasing. In order to solve this problem, we developed a root locus model where the zero introduced was closer to the imaginary axis and adjusted to chosen value to find a solution with the appropriate system values. We chose this zero to be close to the imaginary axis and not further along the real axis because we know that the immediate system response is more affected by the zeros closer to the imaginary axis than those further away, and since we want to control the system, we should place the zero closer. Another problem which was encountered during non-linear system modelling using PD values, the system response had greater overshoot and error in the system then observed in the

linear model (this was a expected result). In order to compensate for the non-linearity  $K_I$  value was introduced until the system has an appropriate response and reduce the steady state error which was found. This resulted in the final implemented PID controller for the system.

## 6 Conclusions

In this report we analyzed and evaluated different controller designs to control the mechanical pick and place arm. During linear system analysis, evaluating a P controller we note that the system characteristics did not meet the speed of response desired by the system and no practical solution existed. System's with a fast response had a large over shoot and fluctuation which needed dampening therefore we explored PD controllers determining that PI controllers would not aid with the dampening of the system. Evaluating PD controllers in a linear simulation, we found a suitable response which meets the criteria of the system, and moved to analyzing this system with non-linearities applied to the system. In evaluating the system we found steady state error which was minimized by introducing a integrative component to the system therefore developing a full PID controller. We observed the performance of the system to meet the considerations outlined and will use the developed PID controller and root locus to adjust the system design in actual integration portion of this project.

## 7 Code and simulation files as an attachment

Refer to 'MSE312controlscode.zip' attached for reference simulation files and codes. README file is attached for details.