MSE312

ELECTRONIC & CONTROL DESIGN REPORT

Student Name	***Student ID***
Abdullah Tariq	301297687
Junaid Jawed Khan	301308300
Syed Imad Azeem Rizvi	301297227

Report due date: 25th, JULY 2021

Prepared for: Dr. Mehrdad Moallem

MSE-312 / MECHATRONIC DESIGN - II

MECHATRONIC SYSTEMS ENGINEERING

SCHOOL OF APPLIED SCIENCE

SIMON FRASER UNIVERSITY

Table of Contents

List of Figures

List of Tables

1. Introduction

The primary goal of this project was to develop a system for throwing a ball a predetermined distance. We presented three different mechanical designs in the previous report, and after analyzing their structures, we decided that the throwing ball bot would be our main design.

We will discuss three major aspects of the project in this report: electronic design, control design, and their integration with our chosen mechanical design. As we are building a power circuitry for driving a DC motor carrying an inertial load, the first section of the report will provide a detailed analysis of its electronics. MOSFETs, H-bridges, and other necessary components are included in the design, as well as an evaluation of their performance under various thermal conditions.

As we will be using a feedback controller to control the motor speed and position, the second part of the project will involve implementing the control algorithm on our system. In this section, we will first analyze our electronic system by determining its stability as well as the position of its poles and zeros. Then, in accordance with that, we will construct a feedback loop comprised of either P, PI, or a PID controller. Once the model for speed and position of motor control is created, we will simulate our system and note different parameter values with different plots to obtain an adequate value for the selected controller gains.

2. Electronic Design

This section of the report discusses the project's electronic design. To achieve the required projectile motion for the ball, a DC motor is required to drive the throwing arm at a desired angular velocity. The primary goal of this section is to design and simulate a power electronic drive circuitry for a DC motor, which drives the arm used to throw the ball. We have included the following components for the operation of the DC motor: An H-Bridge Circuit, a current sensor, a PWM controlled voltage source, and a DC voltage source.

2.1. Power Circuitry Design

2.1.1. H-Bridge Circuit (TB6568KQ)

An H-bridge circuit is used to change the direction of a motor; in other words, it is an electronic circuit that changes the polarity of a voltage-applied across the motor.

Figure 1 depicts a transistor-based H-bridge circuit. As seen in the diagram, two types of transistors are used: PNP and NPN. We know from transistor electronics that when current is applied at its base, the transistor will conduct. We can also note that the PN junction diodes are used as freewheeling diodes, releasing electrical energy even when the motor is turned off.

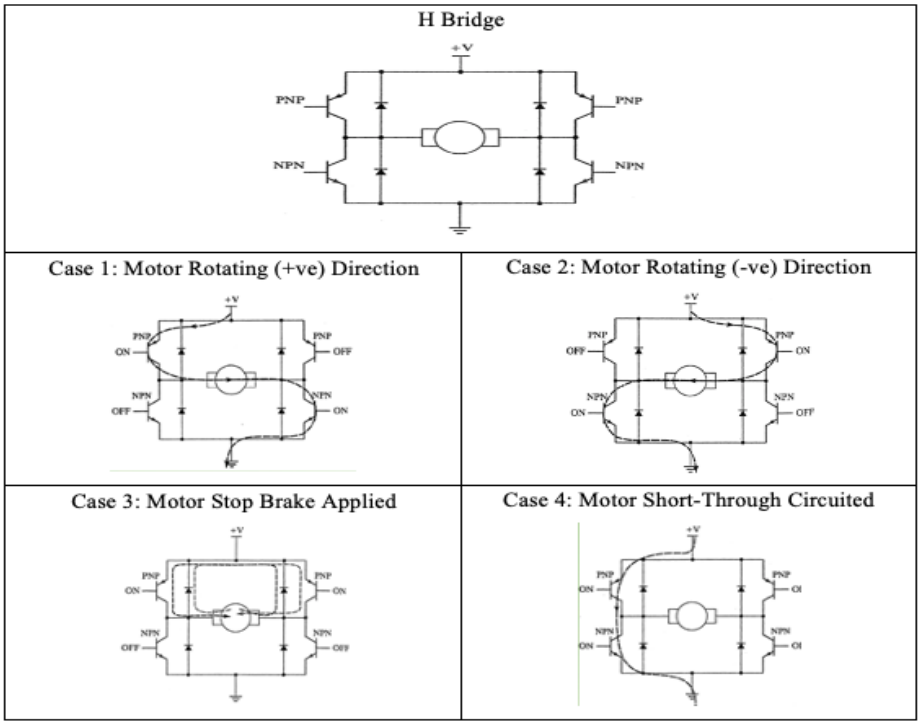


Figure 1: H-Bridge Operation Scenarios

Table 1: H-Bridge Cases Evaluation

SW1	SW2	SW3	SW4	MODE
OPEN	OPEN	OPEN	OPEN	OFF
CLOSED	OPEN	OPEN	CLOSED	CW
OPEN	CLOSED	CLOSED	OPEN	CCW
CLOSED	CLOSED	CLOSED	CLOSED	INVALID

With reference to Figure 1 and Table 1, if PNP (left) and NPN (right) are both turned on, the motor will rotate clockwise. If both PNP (right) and NPN (left) are turned on, the motor will rotate counterclockwise. If both left transistors are turned on and both right transistors are turned off, a short circuit is formed. Dynamic or regenerative braking can be performed by turning on both upper transistors while turning off both lower transistors. The duty cycle can be changed to control the motor's speed.

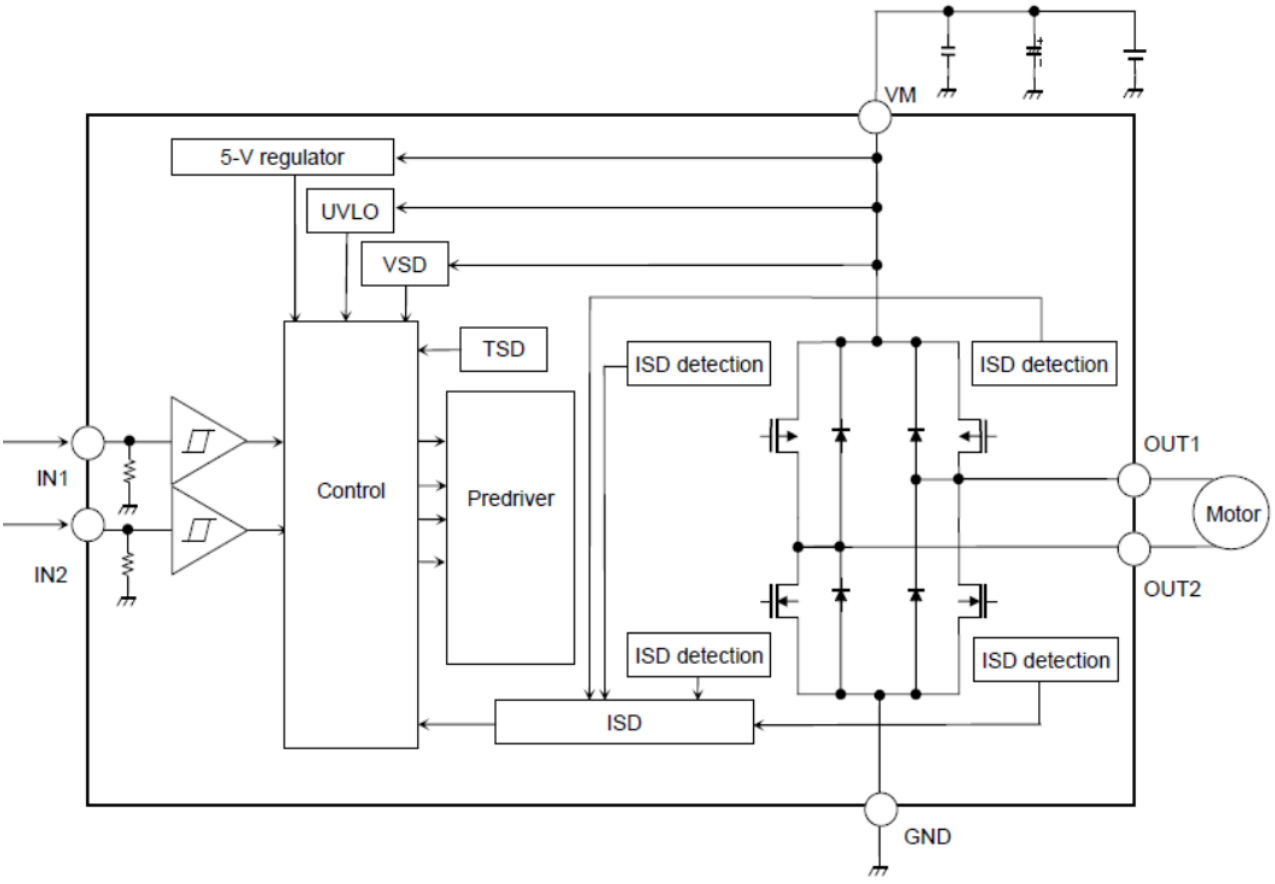


Figure 2: H-Bridge TB6568KQ

The motor driver we used for our motor control is the TB6568KQ. The functional block diagram of the driver is shown in Figure 2. The TB6568KQ is a full bridge DC motor driver IC that uses the MOS process for output power transistors. Aside from providing bidirectional motor control, the IC includes several protection circuits that protect the motor from overcurrent, overvoltage, thermal effects, undervoltage, and shoot through current.

Figure 3 is the data sheet of TB6568KQ H-bridge driver. This datasheet will be referenced in the Heat Sink design section of this report.

Absolute Maximum Ratings (Note) (Ta = 25°C)

Characteristics	Symbol	Rating	Unit
Power supply voltage	VM	50	V
Output voltage	V_O	50	V
Output current	I_O (peak)	3	A
Input voltage	V_{IN}	−0.3 to 5.5	V
Power dissipation	P_D	1.25 (Note 1)	W
Operating temperature	T_{opr}	−40 to 85	°C
Storage temperature	T_{stg}	−55 to 150	°C

Electrical Characteristics (unless otherwise specified, Ta = 25°C, VM = 24 V)

Characteristics		Symbol	Test Condition	Min	Typ.	Max	Unit
Power supply current		I_{CC1}	Stop mode	—	2.5	8	mA
		I_{CC2}	CW/CCW mode	—	2.5	8	
		I_{CC3}	Short Brake mode	—	2.5	8	
Control circuit IN1 pin, IN2 pin	Input voltage	V_{INH}		2	—	5.5	V
		V_{INL}		0	—	0.8	
	Hysteresis voltage	$V_{IN\ (HYS)}$		—	0.4	—	
	Input current	I_{INH}	V_{IN} = 5 V	—	50	75	μA
		I_{INL}	V_{IN} = 0 V	—	—	5	
PWM frequency		f_{PWM}	Duty: 50 %	—	100	—	kHz
PWM minimum pulse width		$f_{PWM\ (TW)}$	(value given as a guide)	1	—	—	μs
Output ON-resistance		$R_{ON\ (U+L)}$	I_O = 3 A	—	0.55	0.9	Ω
Output leakage current		$I_{L\ (U)}$	VM = 50 V, V_{OUT} = 0 V	−2	—	—	μA
		$I_{L\ (L)}$	VM = V_{OUT} = 50 V	—	—	2	
Diode forward voltage		$V_{F\ (U)}$	I_O = 3 A	—	1.3	1.7	V
		$V_{F\ (L)}$	I_O = −3 A	—	1.3	1.7	

Figure 3: H-Bridge (TB6568KQ) Datasheet

2.1.2. Current Sensor

In our design, we will use a current sensor, which is an electronic device that detects current and then generates a signal proportional to that value.

The basic principle for current sensing devices is that current flowing through a conducting wire generates a magnetic field around it and uses two approaches: direct and indirect.

Direct sensing is based on Ohm's law, whereas indirect sensing is based on Faraday's and Ampere's laws. As shown in Figure 4, we will use a built-in MATLAB Simulink block for current sensing in our simulation.

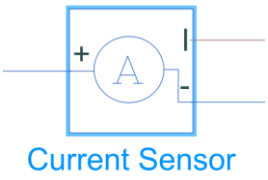


Figure 4: Current Sensor Block

The use of a current sensor will be critical in the control portion of our project because we will need to measure the armature current to implement feedback control of the motor current.

2.1.3. Heat Sink Design

Heat sinks are one of the most effective components for thermal management in electronics. Its primary function is to direct heat away from the hot device. While operating, electronic components can generate a significant amount of heat, and overheating can lead to component failure. As a result, designing a heat manager for them is critical. The magnitude of heat generation is directly related to power dissipation or the surface area of the electronic case from which power is dissipated, the greater the power dissipation or the smaller the surface area, the greater the generated heat. In other words, by increasing the surface area of our electronic case, we can reduce the risk of overheating.

In our project, the major source of heat generation is the H-bridge driver (TB6568KQ). Figure 5 shows the thermal characteristics of this driver.

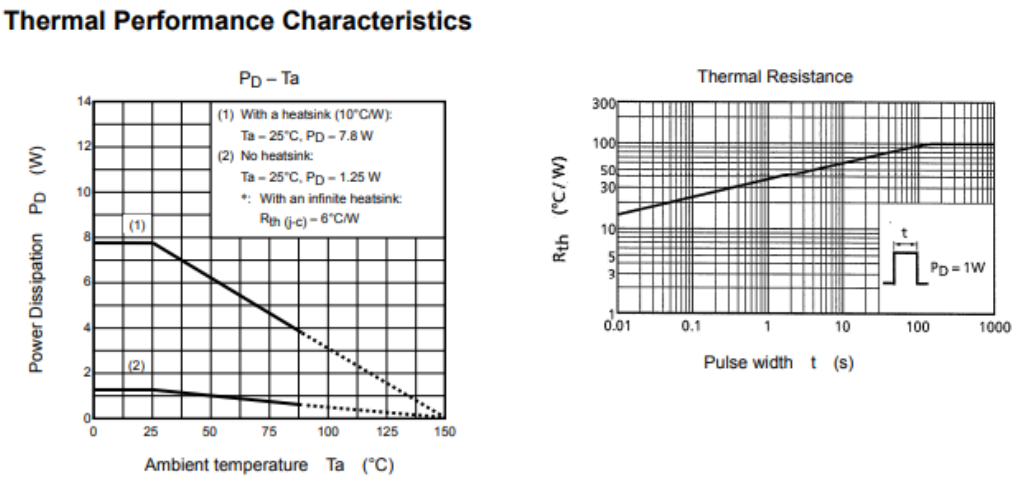


Figure 5: Thermal characteristics of H-Bridge.

The key point from the above figure is that if we operate our motor in the region covered by graph 2, we don't need a heatsink. However, if we want to run the motor anywhere between graphs 1 and 2, we'll need a heatsink to keep the H-bridge from overheating.

The thermal calculation for this component is as follows:

Based on the data sheet of TB6568KQ shown in Figure 5, the ambient temperature and maximum junction temperature are defined as $25^{\circ}C$ and $150^{\circ}C$.

The maximum power dissipated by the H-bridge can be calculated as in Equation 1.

$$P_D = I_0^2 * R_{H-bridge}$$
$$P_D = (3) * (0.9)$$
$$P_D = 8.1\, Watts$$

Equation 1

Thermal resistance between the junction is defined as shown in Equation 2.

$$\theta_{jc} = 6°c/w$$

Equation 2

The thermal resistance between junction and ambient can be calculated as shown in Equation 3.

$$\theta_{jc} = \frac{T_{junction,max} - T_{amb}}{P_D}$$
$$\theta_{jc} = \frac{150° - 25°}{8.1}$$

Equation 3

$$\theta_{jc} = 15.43\ c/w$$

The thermal resistance between the case and the ambient can be calculated as shown in Equation 4.

$$\theta_{JA} = \theta_{CA} + \theta_{JC}$$
$$\theta_{JA} = \theta_{CA} - \theta_{JC}$$
$$\theta_{JA} = 9.432°c/w$$
$$\theta_{JA} = 6°c/w$$

Equation 4

Finally, the case area required to dissipate the generated heat can be calculated as shown in Equation 5.

$$A_{req} = \left(\frac{50}{\theta_{JA}}\right)^2\ cm^2$$
$$A_{req} = 28.101\ cm^2$$

Equation 5

2.2. Motor Selection

We were given the option of choosing our own motor for this project. To complete the project's deliverables, we selected the three motors listed below based on their respective features and specifications that met our project requirements.

- Motor 1: GM8224S009
- Motor 2: C23-L33 W10
- Motor 3: PC280LG-011

Our goal is to simulate the operations of the three motors to evaluate their performance when subjected to a desired inertial load.

Table 2 shows the important features of the three shortlisted motors. This table was populated using the datasheets of the respective motor.

Table 2: Motor Comparison for RPM and Continuous Torque

Motor	Rated RPM	Rated Continuous Torque (Nm)
GM8224S009	720	0.1
C23-L33 W10	4700	0.117
PC280LG-011	8200	0.0255

Appendix A contains datasheets for the three motors that were shortlisted. In the following section, we will analyze the behavior of the short-listed motors to choose our final motor for the controller design.

2.2.1. Motors's comparison:

In this section, the motors shortlisted in Section 2.2 are compared with each other. The Simulink model in Figure 9 will be simulated with a duty cycle of 75% and the DC motor parameters will be changed according to the shortlisted motors

Figures 6, 7 and 8 are the resultant currents, speed, and torque of each motor for a duty cycle of 75%.

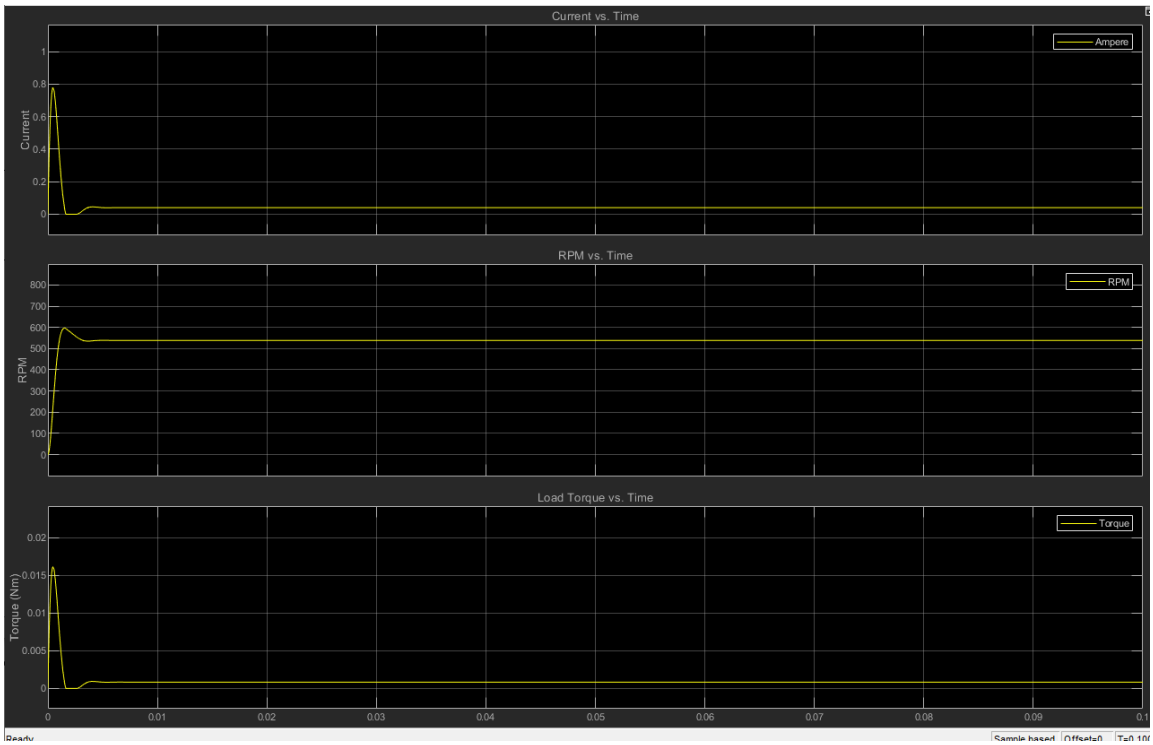


Figure 6: Motor 1 Resultant Current, RPM, and Torque for a 75% duty cycle.

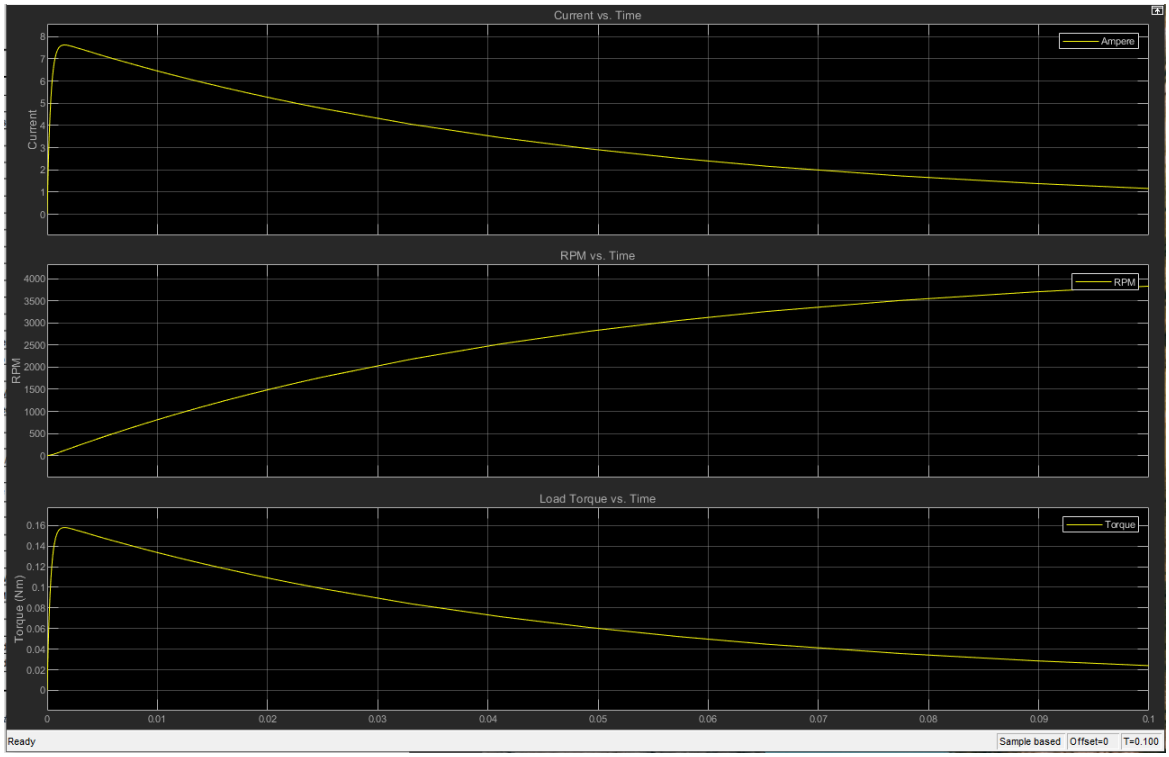


Figure 7: Motor 2 Resultant Current, RPM, and Torque for a 75% duty cycle.

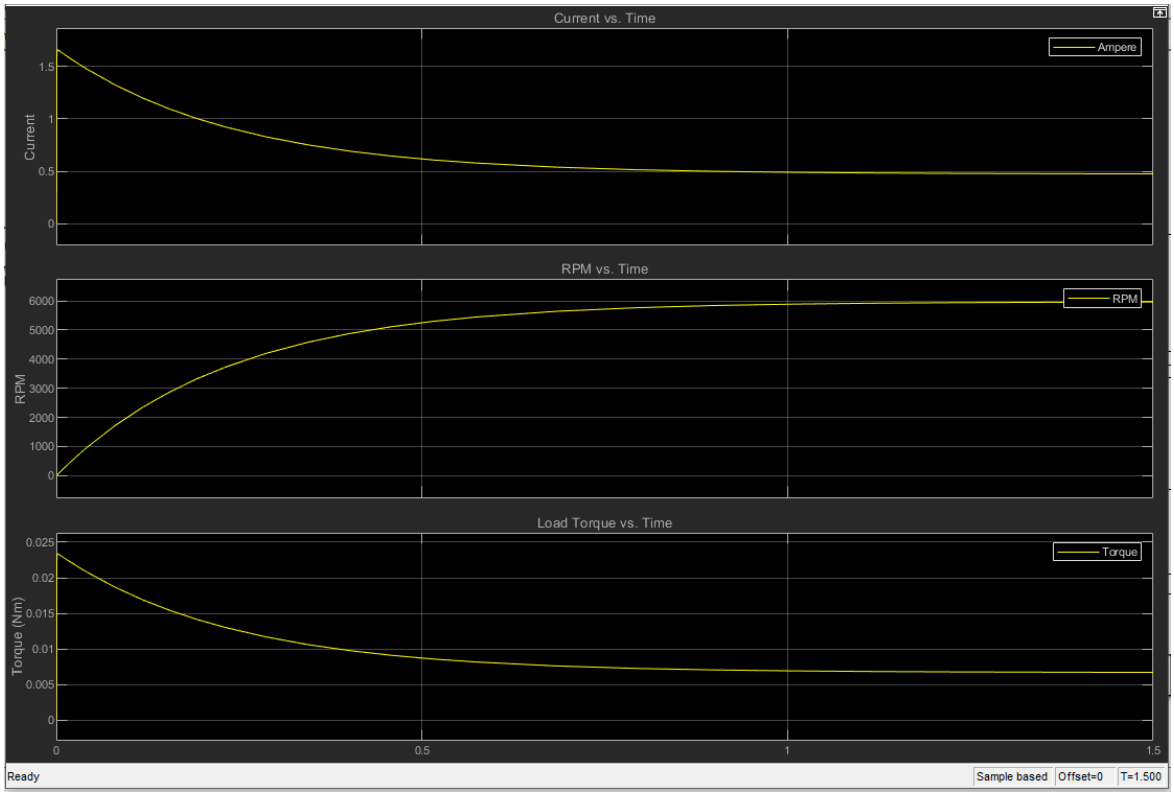


Figure 8: Motor 3 Resultant Current, RPM, and Torque for a 75% duty cycle.

Table 3 below illustrates the all the important parameters of each motor's simulation results that will be used for selecting the optimal motor for our project.

Table 3: Important results of each shortlisted motor.

Duty Cycle 75%	Motor 1	Motor 2	Motor 3
RPM	539 rpm	4370 rpm	5970 rpm
Current	40.62 mA	233.575 mA	474.797 mA
Torque	8.41e-4 Nm	4.461 e-3 Nm	6.695e-3 Nm
Time to reach Steady State	0.472 s	0.353 s	1.558s

The motor speed and torque data in the Table 3 were expected based on the motor data sheets in Appendix A. Motor 3 has the highest no-load speed, whereas Motor 1 has the slowest. The system's torque was greatest for Motor 3 to achieve its no-load speed. As higher torque is required to achieve high speed, Motor 1 had the lowest torque due to its lower no-load speed.

Another important consideration when choosing a motor is the time it takes for the motor to reach a steady state. According to Table 3, Motor 2 took the shortest time to reach steady state, while Motor 3 took the longest.

Based on the preceding discussion, Motor 1 requires the least amount of electromagnetic torque to generate enough rpm to drive the arm. As a result, we decided to use Motor 1 for our throwing arm project. However, there is a trade-off between Motors 1 and 2 in terms of the time required to reach no-load speed, but the time difference is not significant enough to affect our controller design criteria.

2.3. Simulation of a DC Motor

In this section, we will first look at the simulation results of a DC motor using MATLAB Simulink in three different scenarios:

1) PWM and Average Mode Simulation
2) With and Without Gear Reduction Simulation
3) With and Without Gear Reduction Simulation

The output behavior of motor parameters will be discussed and compared in each case, followed by a comparison of the three shortlisted motors by varying the duty cycle of their PWM signal.

Figure 9 shows the designed Simulink model of a DC motor employed for analysis in this section of the report.

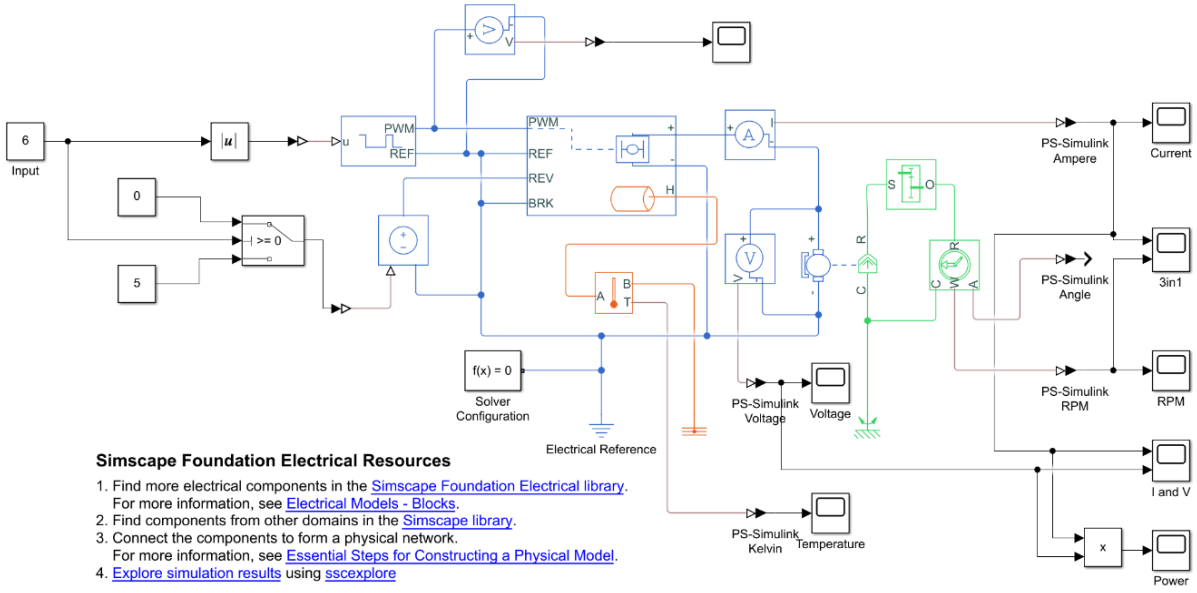


Figure 9: Simulink Model of a DC Motor

The parameters of the PWM, H-bridge and motor block used in the above Simulink model are shown in Table 4.

Table 4: Simulink DC motor model parameters.

Parameters	Values
Controlled PWM Voltage	
PWM Frequency	4000Hz
Input value 0% DC	0
Input value 100%DC	12
Output Voltage	12
H-Bridge	
Total bridge on resistance	0.55 ohms
Freewheeling diode on resistance	1.3/3 ohms
Measurement temperature	298.15 K
Freewheeling diode off-state conductance	1e-6 S
Total bridge on resistance at second measurement temperature:	0.9 ohms
Freewheeling diode on resistance at second measurement temperature	1.7/3
Second measurement temperature	303.15 K
DC Motor	
Armature inductance:	2.34 mH
Stall torque:	3.01e-1 N*m
No-load speed:	720 rpm
Rated DC supply voltage:	12 V
Rotor inertia:	2000 g*cm^2
Rotor damping:	1e-04 N*m/(rad/s)

2.3.1. Motor (GM8224S009) No-Load Simulation:

2.3.1.1. PWM and Average Mode Simulation

Figures 10 and 11 show the resulting motor current and speed when the motor is simulated in PWM mode versus Average mode.

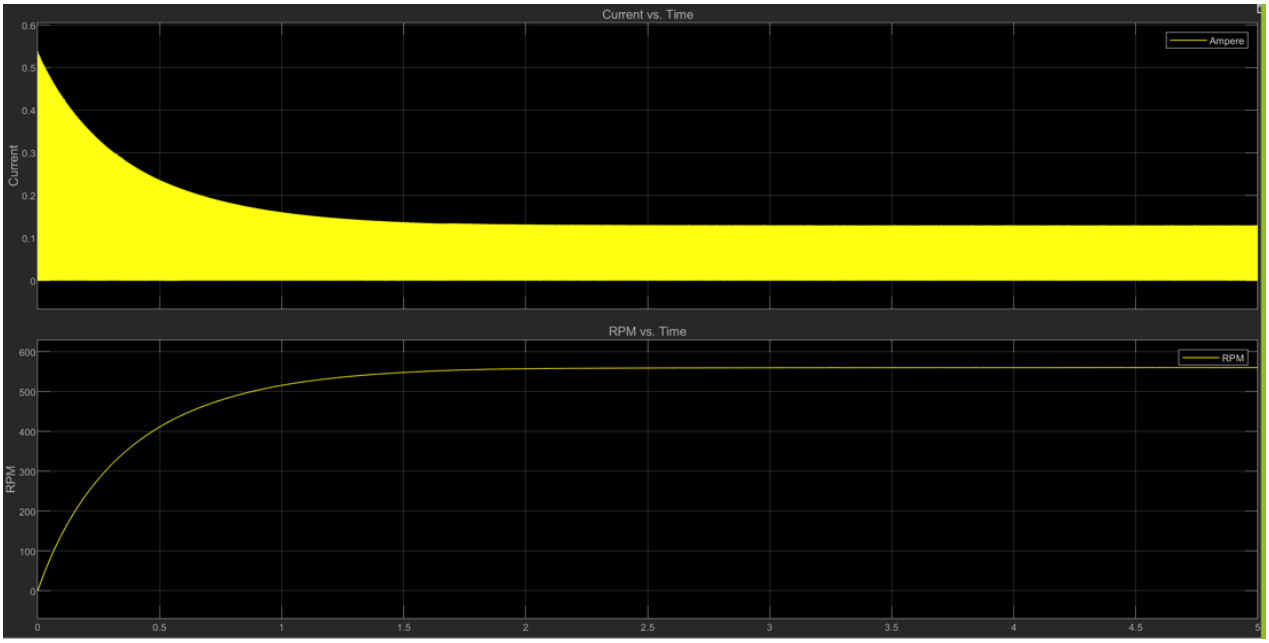


Figure 10: Motor current and RPM in PWM mode

Figure 11: Motor current and RPM in Average mode

In this case, two simulation modes, PWM and average mode, are compared. The simulation duty cycle is set to 50%. First, we ran our model in PWM mode, and Figure 10 shows the results. The plot shows that in PWM simulation mode, current output switches between high and low, while the duration for a pulse to be high or low is the same due to the applied 50% duty cycle. In average mode, the current output does not switch between high and low because the average of the PWM pulse is applied to the motor.

When we compare the RPM plots in both cases, we can see that the average mode RPM rise time is shorter than the PWM mode. In average mode, RPM reaches steady state in less than 0.25 seconds, whereas it takes 2 seconds in PWM mode.

2.3.1.2. With and Without Heatsink Simulation

Figures 12 and 13 show the resulting temperature when the motor is simulated without a heatsink versus with heatsink for the H-bridge.

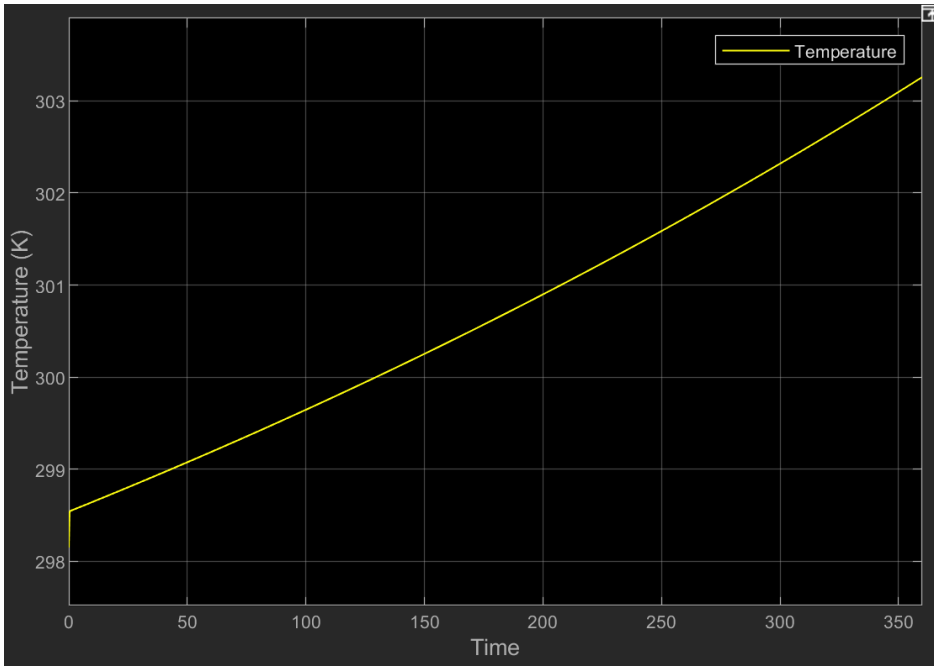


Figure 12: H-bridge temperature without heatsink

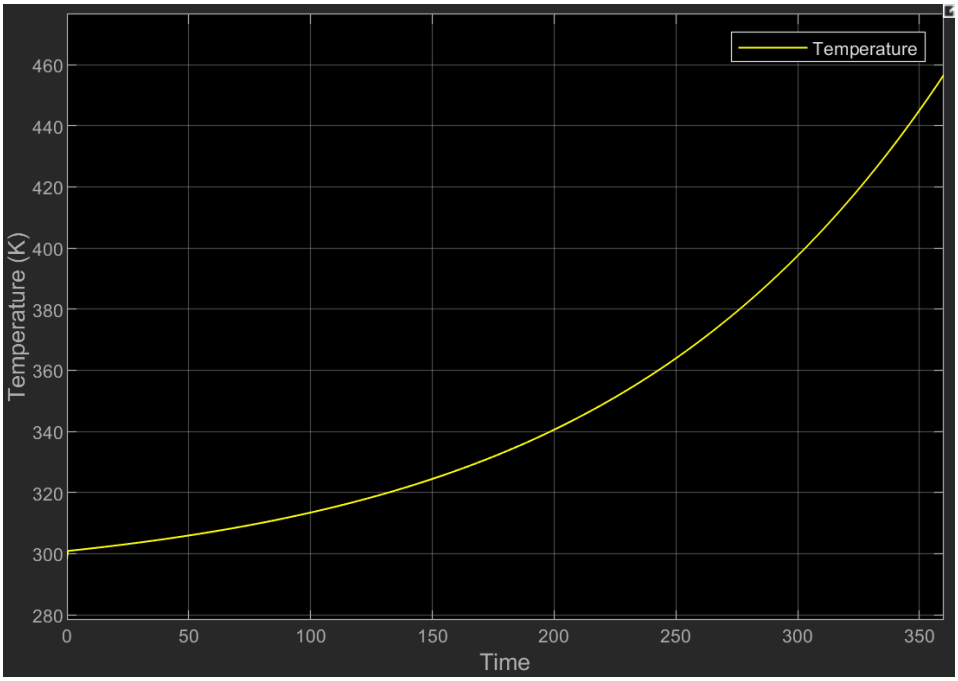


Figure 13: H-bridge temperature with heatsink

While running, electronic components generate heat; it is critical to consider the effect of heat sinks on the system, as excessive heat generation can cause component failure.

Figure 12 depicts the temperature rise over time when no heat sink is used; it is important to note that after 200 seconds, the temperature rises at an increasing rate. Now, if we look at Figure 13, we can see how efficiently the heat sink affects the system, as the temperature with the heat sink is around 303K at time 350 seconds, but it is at 450K without the heat sink. The purpose of a heat sink is to increase the surface area of the cover so that heat generated during operation has more surface area to escape from the system.

2.3.1.3. With and Without Gear Reduction Simulation

Figures 14 and 15 show the resulting rpm when the motor is simulated with a gear box versus without a gearbox at the output.

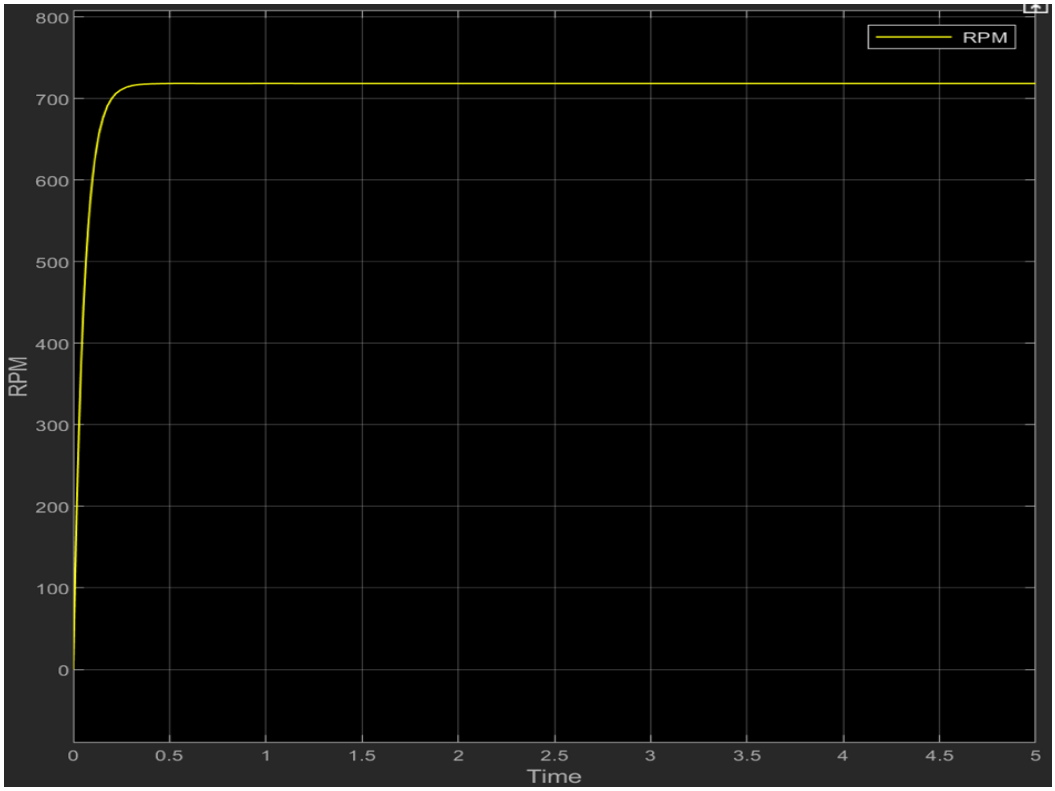


Figure 14: Output speed with gears.

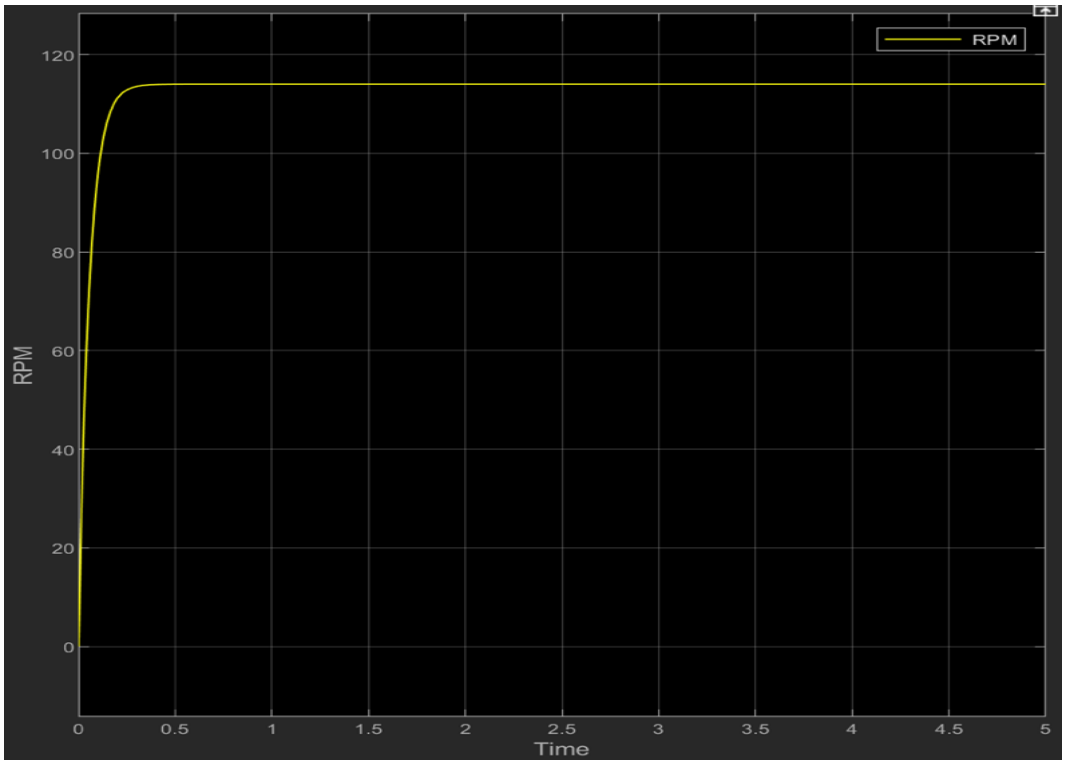


Figure 15: Output speed with gear reduction

Concept of gear reduction is quite straight forward; Gear reduction is as shown in Equation 6.

$$\text{New RPM} = \frac{\text{No load speed}}{\text{gear ratio}}$$

$$New\ RPM = \frac{720}{6.3} = 114.2$$

Equation 6

Figure 14 depicts a situation in which there is no gear reduction; the plot shows that RPM is equal to the motor's no-load speed, as shown in Table 3. In Figure 15, gear reduction is used because the gear ratio is 6.3. The final RPM obtained is 114, which is equal to the calculated equation 6.

3. Controller Design

In most control systems, a simple unity feedback is not sufficient for achieving the desired output. Therefore, a controller is typically added to the system to minimize the error measured and desired output. There are several ways to design a feedback controller for your plant. However, the most common design approach for a motor-controlled application is the cascaded design approach.

The cascaded design approach allows for the motor control to be decoupled into three feedback loops, with the innermost loop being the fastest and the outermost loop being the slowest. The cascaded design approach enables the fastest inner loop to deal with disturbances while ensuring that the controlled variable does not deviate significantly from the desired set point.

As mentioned in the previous section, we were given the option in this project of choosing our own DC motor for our project. We chose Motor 1 because it has a decent rated speed, continuous torque, and is affordable based on our budget.

The selected DC motor is GM8724S009, and its parameters are shown in Table 5

Table 5: GM8724S009 important parameters

Motor Parameter	Symbol	Value
Rated DC Voltage	V	12
Armature Resistance	R	4.33
Armature Inductance	L	$2.34e^{-3}$
Damping Constant	B	$1.1e^{-4}$

Torque/Back-emf Constant	K_m	$2.18e^{-2}$
Rotor Inertia	J	$2.3e^{-4}$
Gear Ratio	n	6.3
Gear efficiency	η	0.95
Total Inertia	J_{total}	$8.3e^{-3}$

3.1. Design Criteria

There are no fixed requirements for settling-time, overshoot, or steady-state error in this project because we are free to choose our own motor and tune it to meet our needs.

To launch a ball to the specified target, the rotating arm will go through three stages. The arm will accelerate to the desired angular velocity in the first stage, then continue at the desired angular velocity until it reaches the desired launch time in the second stage. At the time of launch, the rotating arm is in stage 3 and will return to its initial position after the ball has been launched. In summary, the first two stages control the motor's speed, while the final stage controls the motor's position.

Table 6, and Figures 16 and 17, lists the design specifications for the rotating arm’s current, and speed.

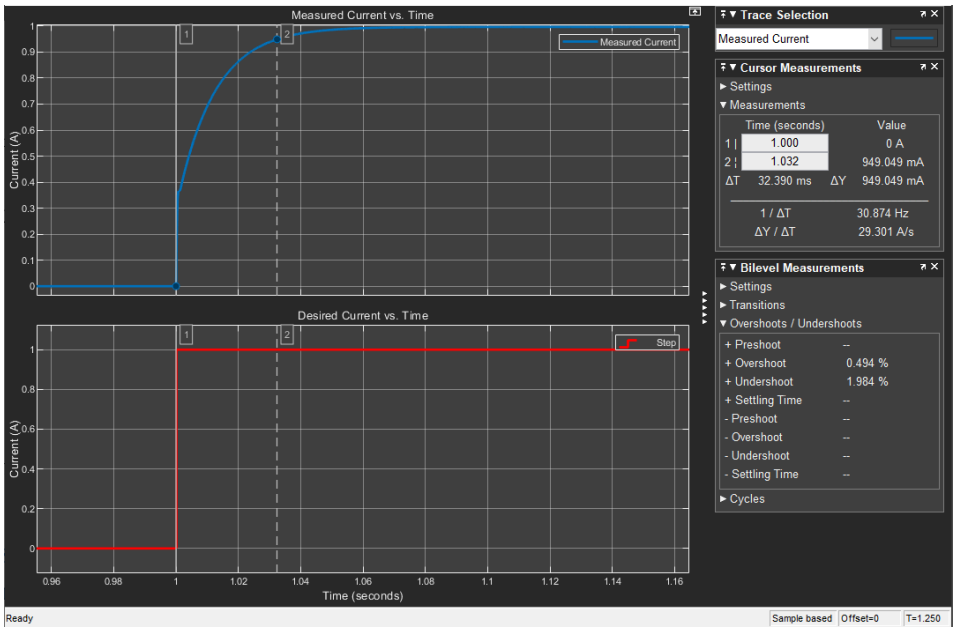


Figure 16: Step Response for Current Controller

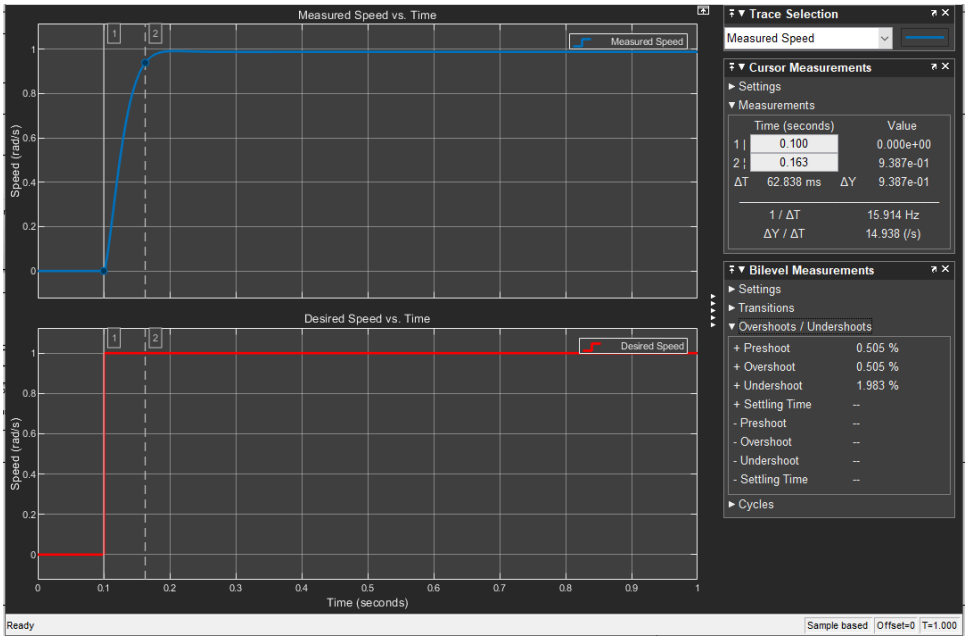


Figure 17: Step Response for Speed Controller

Table 6: Current, Speed and Position controller desired characteristics.

Control Block	Percentage Overshoot	Settling Time
Current Control	< 0.5%	< 0.05 sec
Speed Control	< 2%	< 0.12 sec

Based on the step-response behavior of the current and speed controller blocks, the tabulated value in table 6 establishes the boundary conditions for our controller design criteria.

We want the current and speed controllers to be as robust as possible; because the current controller is the fastest, we want it to settle quickly with minimal overshoot so it can handle disturbances. Once the current controller is tuned properly, the speed, and position controllers can handle a little delay.

3.2. System Modelling

3.2.1. Motor Equation

The motor receives voltage as input, and it produces torque as an output. This is illustrated graphically, as shown in figure 18.

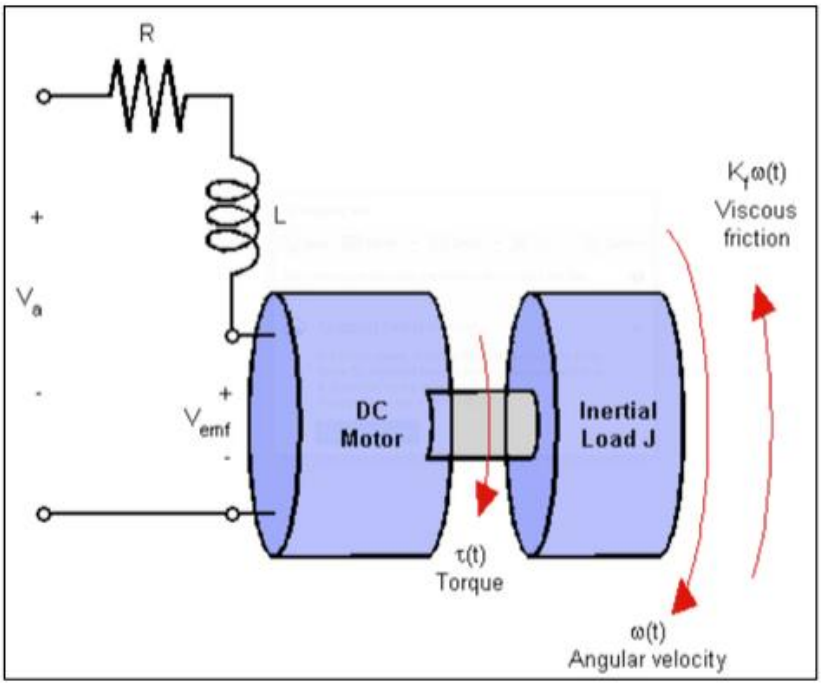


Figure 18: Graphical model of motor and connected load.

Based on Figure 18, the input voltage rotates the shaft of the motor, which is connected to an inertial load, the rotation of the motor shaft rotates the inertial load. Some assumptions that are made in this model are that the rotor and shaft of the motor are rigid, and friction is proportional to the speed of the motor.

The electrical energy from the motor's circuitry is converted to mechanical energy using a torque transducer.

The electromechanical equation relating the electrical system and the mechanical system is shown in equation 7

$$\mathrm{V} = \mathrm{L}\frac{\mathrm{di}}{\mathrm{dt}} + Ri + e \qquad \text{Equation 7}$$

Where the generated back-emf is relative to the angular velocity, gear ratio(n) and motor constant (K_m) as shown in Equation 8.

$$e = K_m n\dot{\theta} \qquad \text{Equation 8}$$

The net torque that is applied to the connected inertial load is as shown in Equation 9

$$\tau - \tau_{friction} = J\ddot{\theta} \qquad \text{Equation 9}$$

Where, the motor torque is related to the armature current by the motor constant (K_m), gear efficiency and gear ration, as shown in equation 10.

$$\tau = \eta K_m n i \qquad \text{Equation 10}$$

The friction torque comprises of load resistance and frictional torque of the motor, resulting in the net torque produced by the motor as shown in equation 11.

$$\tau - \mathrm{B}\dot{\theta} - \tau_{load} = J\ddot{\theta} \qquad \text{Equation 11}$$

3.2.2. Transfer Functions

Appling Laplace transformation on equations 7 and 11, we get the results shown as equations 12 and 13.

$$\mathrm{V(s)} = \mathrm{LsI(s)} + RI(s) + K_m n s\theta \qquad \text{Equation 12}$$

$$\eta K_m n I(s) - \mathrm{B}s\theta(s) - \tau_{load} = Js^2\theta(s) \quad \text{Equation 13}$$

Assuming zero initial conditions, we can control the position and angular velocity of the inertial load by controlling armature current of the motor.

Using equations 12 and 13, it can be shown that the transfer function relating angular velocity of the inertial load and the voltage applied at the motor terminals is as shown in equation 14.

$$\frac{\dot{\theta}(s)}{V(s)} = \frac{\eta K_m n}{(Ls + R)(Js + B) + \eta K_m^2 n^2} \quad \text{Equation 14}$$

Taking the integral of the angular velocity shown in equation 14, the transfer function relating angular position and voltage applied across the motor are related as shown in equation 15.

$$\frac{\theta(s)}{V(s)} = \frac{\eta K_m n}{s[(Ls + R)(Js + B) + \eta K_m^2 n^2]} \quad \text{Equation 15}$$

The motor parameters in Table 5 are plugged into equation 15. The result is shown in Equation 16.

$$\frac{\theta(s)}{V(s)} = \frac{3324.1}{s(s + 1850)(s + 0.843)} \quad \text{Equation 16}$$

Figure 19 shows the root locus of the transfer function shown in Equation 16.

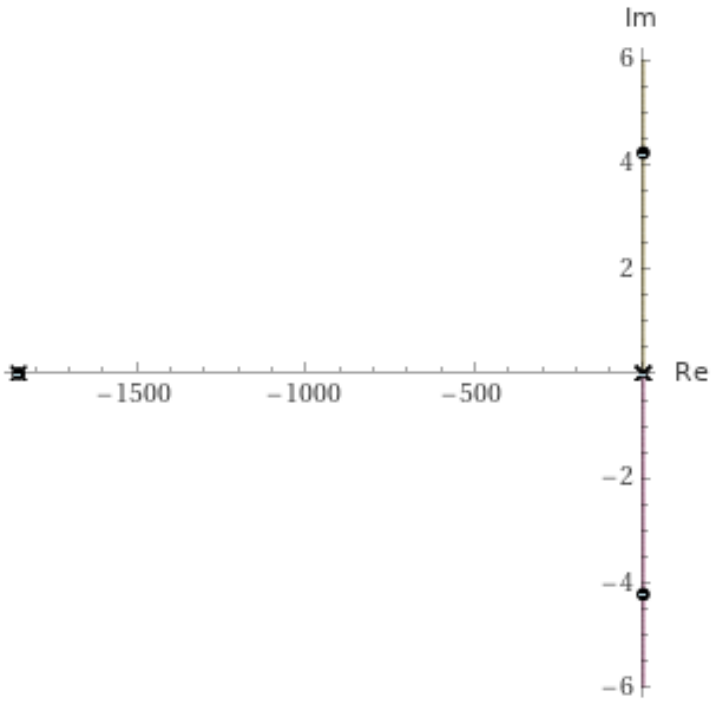


Figure 19: Root-locus of motor transfer function.

Based on Figure 19, the pole at the far left is at 1850, which corresponds to the inverse of the electrical time constant (R/L). Because this pole is relatively far away from the imaginary axis, it has a negligible effect on overall system performance. The poles and zeros closest to the imaginary axis have the most influence.

Based on these pole placements, we can conclude that the mechanical time constant is much larger than the electrical time constant, and this conclusion leads to Cascade control, a popular method of motor control in the industry.

3.3. Cascade Control Scheme

A cascade control strategy consists of inner and outer control loops, with the inner loop being the fastest and the outer loop being the slowest. The fastest loop in motor control applications is the current control, and the slowest is the angular position control.

In this section, we look at the transfer function of the inner current loop, followed by cascaded outer speed loop with inner current loop, and finally the outer position loop cascaded with inner speed and current loop.

3.3.1. Current Control

Figure 20 shows the block diagram of a typical DC motor corresponding to the transfer function represented in Equation 15.

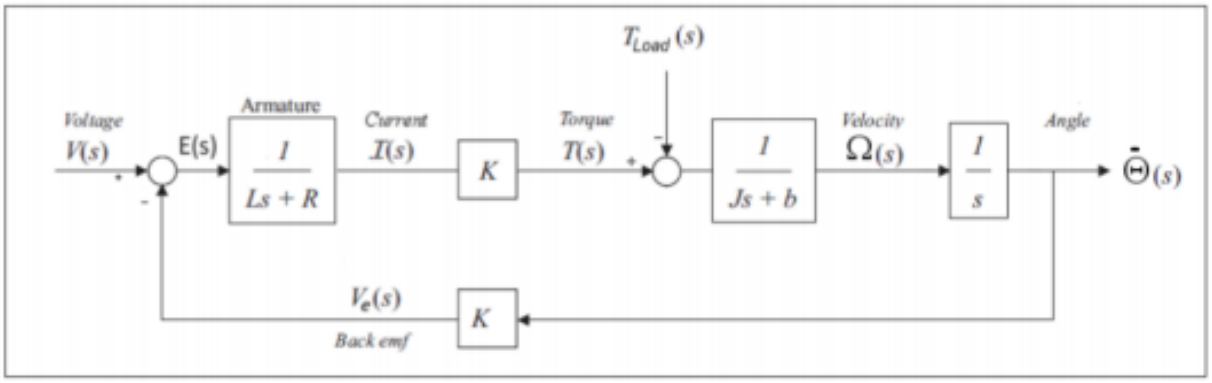


Figure 20: Block diagram of DC motor control.

Rearranging the block diagram for a transfer function of armature current as output we get the transfer function shown in equation 17.

$$\frac{I(s)}{V(s)} = \frac{J(s + \frac{B}{J})}{[(Ls + R)(Js + B) + \eta K_m^2 n^2]}$$ Equation 17

In Equation 17, we can note that the characteristic equation is the same shown in Equation 17 with a zero at $-\frac{B}{J}$.

This new zero is close to the pole represented by the inverse of your mechanical time constant. The poles are attracted to zeros by using a feedback loop with gain. As a result, the pole represented by the inverse mechanical time constant can be cancelled with the new zero, allowing us to approximate the second order transfer function shown in Equation 17 with a simple purely electrical first order transfer function shown in Equation 18.

$$\frac{I(s)}{V(s)} = \frac{\frac{1}{L}}{[s + \frac{R}{L}]}$$ Equation 18

The approximation shown in equation 18 allows you to decouple the electrical and mechanical part of your controller, i.e., the torque and the trajectory generation. The following is the control loop for current control, where the motor transfer function is represented by equation 18.

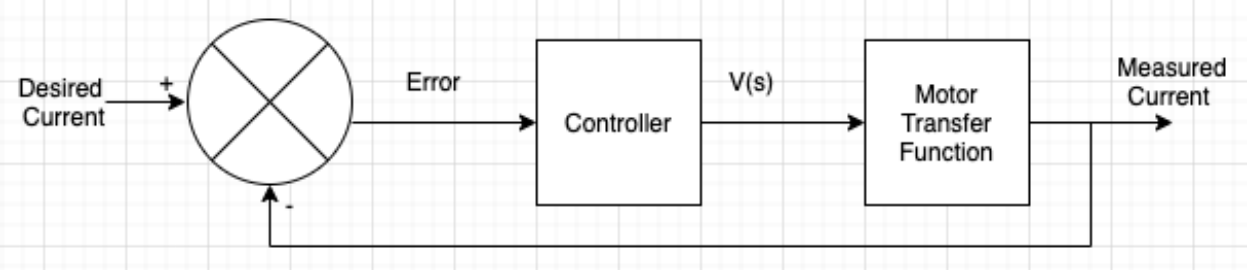


Figure 21: Block diagram of current loop.

The controller block in figure 21 could be a simple P, PI, PD, or PID. Each combination has different effects on the system.

3.3.1.1.1. Controller Choices

The P controller is simply an amplifier that provides a proportional constant based on the error. A simple P controller will have a slow/sluggish response if the gain is small, and a too aggressive response if the gain is high, resulting in an overshoot and large steady state error.

A PI controller will aid in reducing the steady-state error introduced by a simple P controller. A PI controller, on the other hand, slows response time and decreases system stability. Furthermore, with this controller, the output rise times cannot be reduced.

A PD controller improves system stability by analyzing error changes and adjusting gain to minimize overshoot. A PD will reduce overshoot while increasing steady-state error.

A PID controller provides us with optimal control for fast response, minimal steady-state error, and increased stability.

We chose a PI controller to design the current loop because the zero generated by the PI controller is used to cancel the pole at R/L shown in Equation 18, allowing us to control the current by changing the gains of the PI control block.

The closed-loop transfer function of the current loop with a PI controller to cancel motor pole at R/L is as shown in equation 19.

$$\frac{I(s)_{measured}}{I(s)_{desired}} = \frac{\frac{\mathrm{Kp}}{\mathrm{Ls}}}{1 + \frac{Kp}{Ls}} = \frac{1}{1 + \frac{s}{(\frac{Kp}{L})}}$$

Equation 19

Where, $\frac{\mathrm{Kp}}{\mathrm{Ls}}$ is the open loop gain of the system shown in figure 19.

Moreover, the value $\frac{Kp}{L}$ is the bandwidth of the closed loop system, which should be orders of magnitude small in comparison to the sampling frequency of the closed loop system for a practical implementation of this controller. More on the sampling frequency and tunning of this inner loop is presented in the Integration section.

3.3.2. Speed Control

The angular speed of the motor is controlled by speed controller cascaded to the current control loop presented in the previous section as shown in Figure 21.

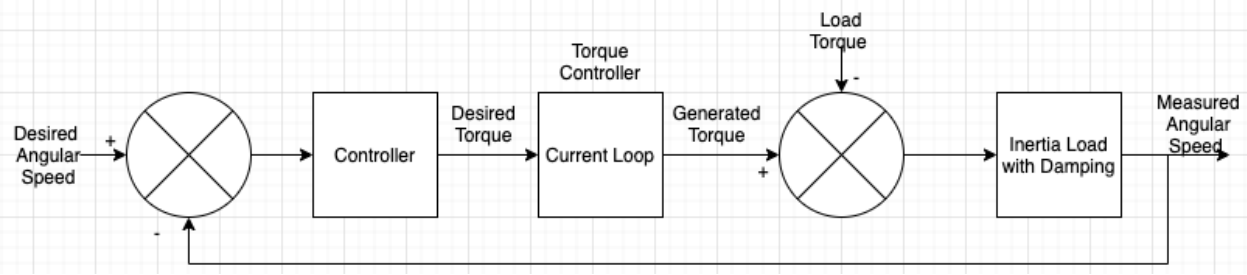


Figure 22: Block diagram of cascaded speed loop.

The inner current loop shown in figure 22 will convert the desired torque into current equivalent using the torque constant of the motor as shown in equation 20.

$$\tau_{desired} = K_m n I_{desired}$$

Equation 20

For a practical implementation of a cascaded speed controller, the bandwidth of the closed loop system shown in Figure 22 should be orders of magnitude lower than the bandwidth of the Current Control loop shown in Figure 21.

The available controller choices of the speed controller are the same as presented in the previous Current Control section. The details on the selected controller and tunning are presented in the Integration section of this report.

3.3.3. Position Control

The angular position of the motor is controlled by a position controller cascaded to the speed control loop presented in the previous section, as shown in Figure 23.

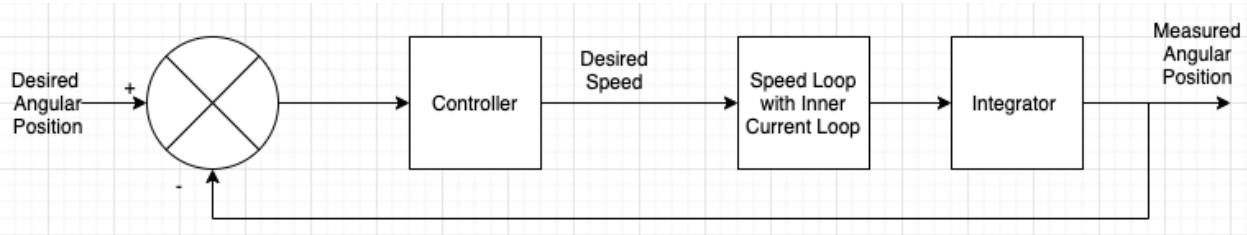


Figure 23: Block diagram of cascaded position control.

For a practical implementation of a cascaded position controller, the bandwidth of the closed loop system shown in Figure 23 should be orders of magnitude lower than the bandwidth of the Speed Control loop shown in Figure 22.

The available controller choices of the position controller are the same as presented in the previous Current Control section. The details on the selected controller and tunning are presented in the Integration section of this report.

3.3.4. Trajectory Tracking

Figure 23 in the preceding subsection depicts the final cascaded control of a DC motor. After launching the ball, the DC motor is expected to return to its starting position in our project. Trajectory Control is used to achieve this.

For each launch of a ball to a target location, a DC motor goes through three stages. The DC motor ramps up to a cruise speed in stage one, maintains the cruise speed in stage two, and ejects the ball at the predetermined target angle in stage three. In stage three, after launching the ball, the motor should return to its original resting position in preparation for the next target location.

Trajectory tracking is used to return the motor to its initial state. The trajectory controller is only activated at the end of the motion. The goal here is to define a speed profile that will be used to toggle between speed and position control. Only the speed and current loops are active in stages one and two, and only the position and current loops are active in stage three, i.e., torque commands in stages one and two are based on speed error, and torque commands in stage three are based on position error. Figure 24 depicts a typical speed profile of DC motor used for launching a target.

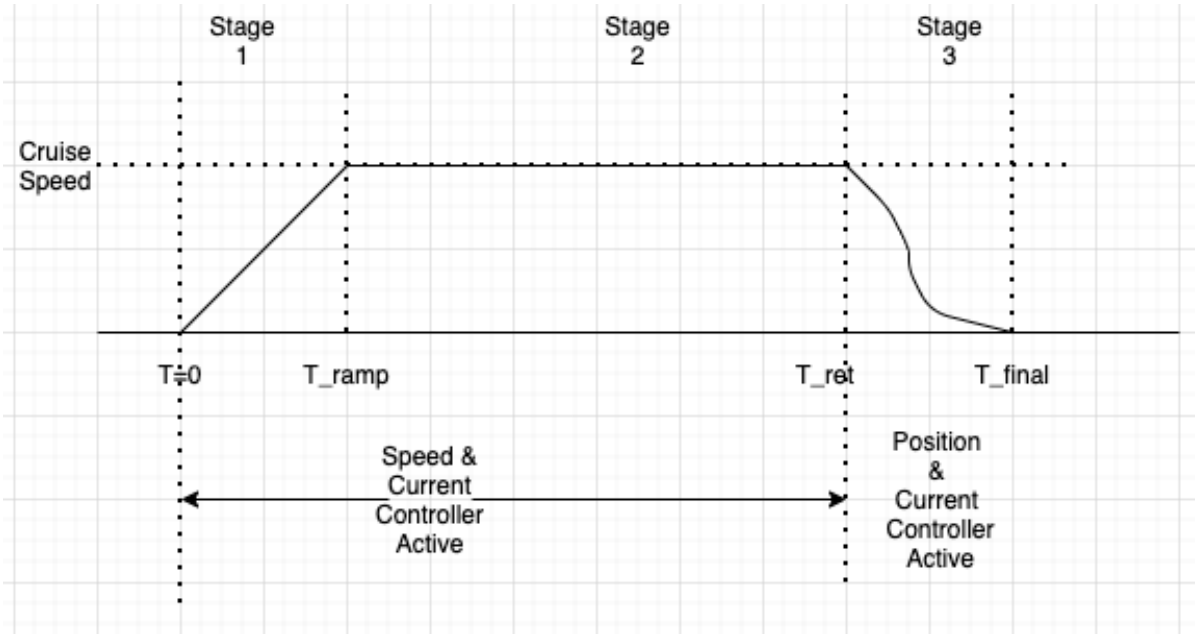


Figure 24: Three stages of a cascaded control of a DC motor.

According to Figure 24, after launching the ball at a predetermined retraction time, the motor returns to its initial state at stage three. Because we have six constraints: the angular position, velocity, and acceleration of the motor at the start of stage three and the desired angular position, velocity, and acceleration to return the motor to its initial state, the profile shown in stage three is approximated using a fifth-degree polynomial.

The six constraints can be satisfied using a quintic trajectory profile. This profile is formulated as shown in Equation 21.

$$\mathrm{q}(\Delta\mathrm{t}) = q_0 + q_1\Delta\mathrm{t} + q_2\Delta\mathrm{t}^2 + q_3\Delta\mathrm{t}^3 + q_4\Delta\mathrm{t}^4 + q_5\Delta\mathrm{t}^5 \quad \text{Equation 21}$$

Where $0 \leq \Delta\mathrm{t} \leq \mathrm{T_{final}} - T_{retraction}$

$q_0, q_1, q_2, q_3, q_4, q_5$ can be determined using the information we have from observing the speed profile in stage three of Figure 24.

When $\Delta\mathrm{t} = 0$, q_0, q_1 *and* q_2 can be determined as shown in Equation 22.

$$q = q_{retraction} = q_0, \quad \text{Equation 22}$$
$$\dot{q}(\Delta\mathrm{t} = 0) = q_1 = \omega_{cruise},\ \ddot{q}(\Delta\mathrm{t} = 0) = 2q_2 = \ddot{q}_{desired}\ at\ T_{retraction}$$

When $\Delta\mathrm{t} = \mathrm{T_{final}} - T_{retraction}$:

By using the results shown in Equation 22 and solving the three equations $q(\Delta t) = 0, \dot{q}(\Delta t) = 0, \ddot{q}(\Delta t) = 0$ simultaneously, we can calculate $q_3, q_4, \& \, q_5$ and generate a trajectory profile that has a speed profile as shown in Figure 24

To control the torque of the motor such that the angular position, velocity, and acceleration follow the desired profile represented by $q, \dot{q}, and \, \ddot{q}$, we use a PID controller. The motion planner function is provided in the Appendix B for your reference.

4. Integration of Electronics and Control

In this section, we will simulate the theory discussed in sections 2 and 3 and analyze the results. This section integrates the electrical and control of the motor with specified load values.

Before fine-tuning our control parameters, we developed a desired kinematic profile for our system to follow. Figure 25 depicts the kinematic profile that will be used in our control system.

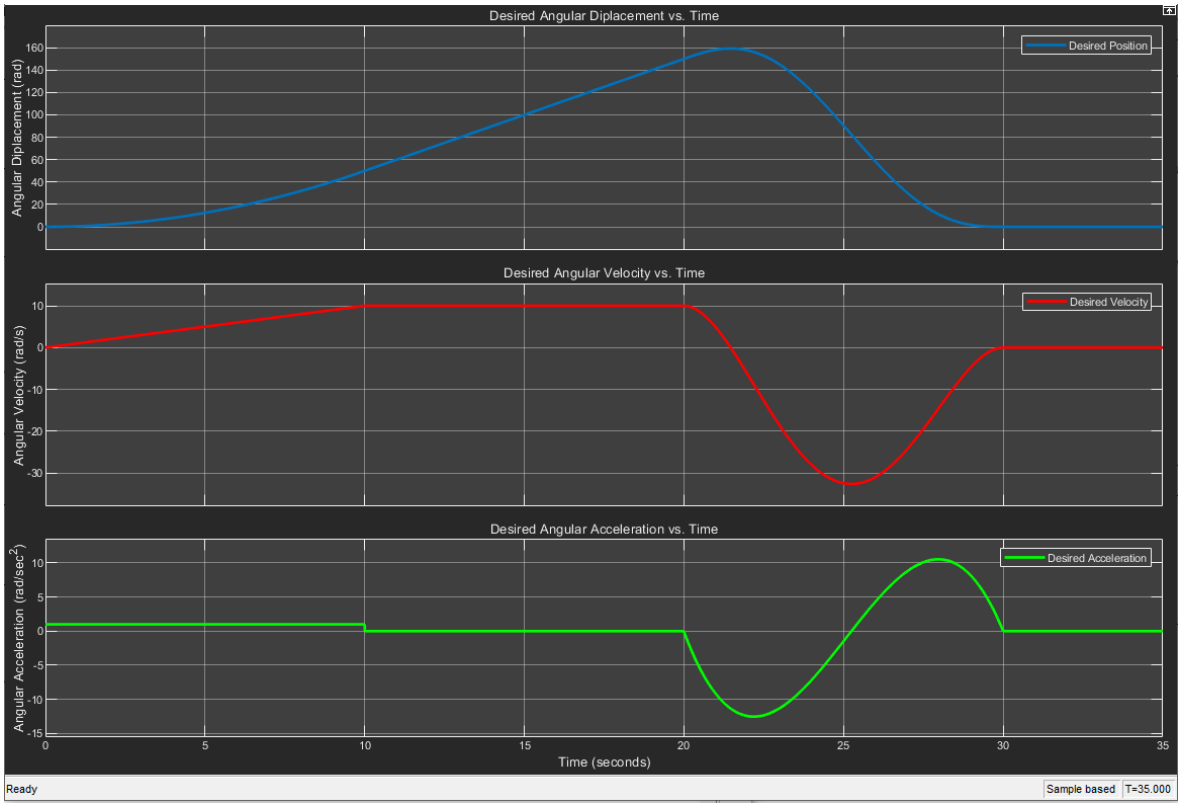


Figure 25: Kinematic Profile for the Project

Based on Figure 25, a quintic polynomial is employed to represent our desired velocity profile and computed the corresponding displacement and acceleration profiles as described below:

- In the interval 0 – 10 sec, the speed is ramping up to 10 rad/s at a constant rate. Therefore, the acceleration in this interval is constant while the angular position is increasing parabolically.
- In the interval 10-20 sec, the speed maintained at 10 rad/s. Therefore, the acceleration in the interval is zero while the angular displacement increases linearly.
- In the interval 20 – 30 sec, the ball is launched at an angular velocity of 10 rad/s and the structure experiences a negative acceleration to return to the initial state for the next target.
- In the interval 30 – 35 sec, the structure has returned to its initial state as all quantities go to zero.

4.1. Simulation Results

4.1.1. Open-Loop System Response

To simulate the system response, we provide our system with a predefined kinematic profile for open-loop analysis. Each PID block has the gain values Kp = 1, Ki = 0, and Kd = 0, i.e., a simple gain with no controller. The current, speed, and position controller blocks are cascaded together in Simulink as shown in Figure 26.

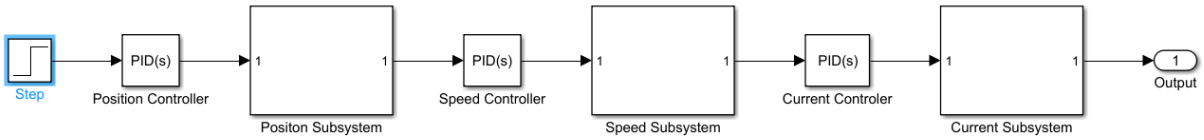


Figure 26: Cascaded position, speed, and current controls.

Figure 26 shows an intuitive system block diagram for the open-loop response. The system is without a feed-back and behaves erratically. Figures 27, 28, and 29 are the respective plots for the current, speed, and position subsystems. These plots are analyzed in the Performance Evaluation section.

4.1.1.1. Current Control

Figure 27 depicts the open-loop current response with the desired kinematic profile.

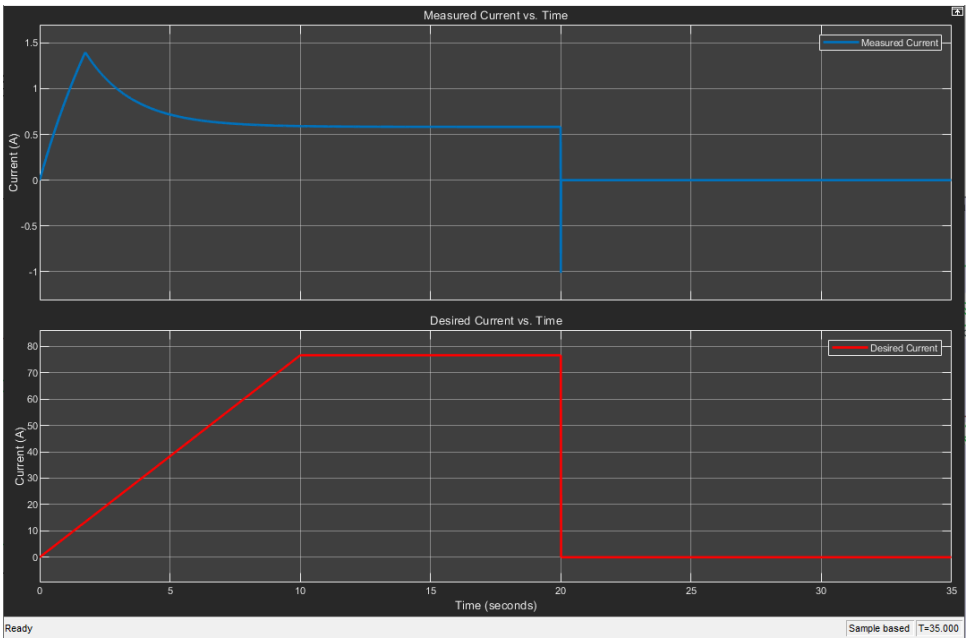


Figure 27: Open-loop current response.

4.1.1.2. Speed Control

Figure 28 depicts the open-loop speed response with the desired kinematic profile.

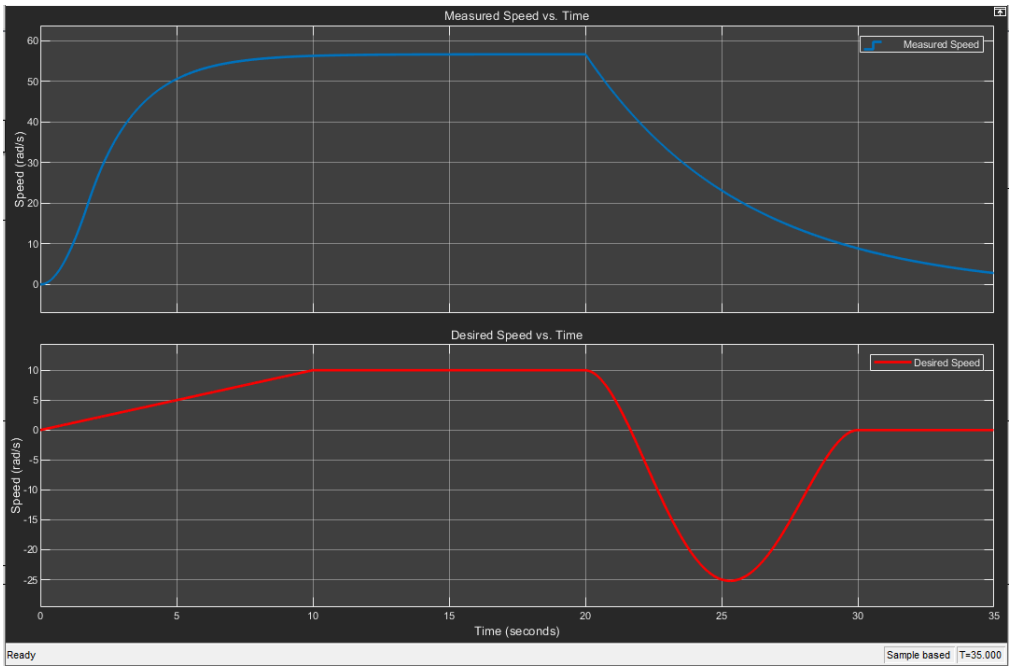


Figure 28: Open-loop speed response.

4.1.1.3. Position Control

Figure 29 depicts the open-loop position response with the desired kinematic profile.

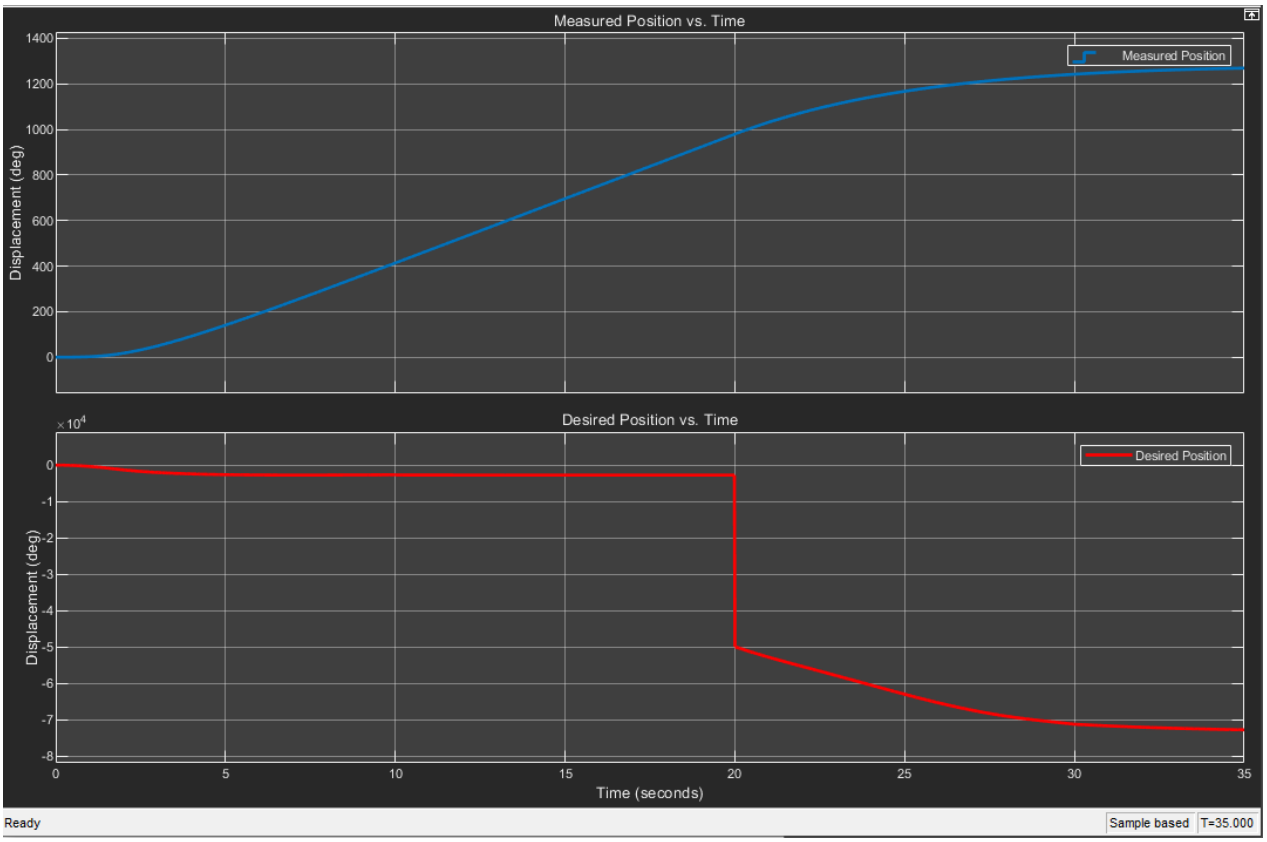


Figure 29: Open-loop position response.

4.1.2. Closed-Loop System Response

For the closed-loop analysis of each controller subsystem, the three respective controllers are in a nested loop as presented in Figures 21, 22, and 23, where the current controller (fast) block is inside the speed controller (moderate) block, and speed controller block is inside the position controller block (slow).

4.1.2.1. Current Control

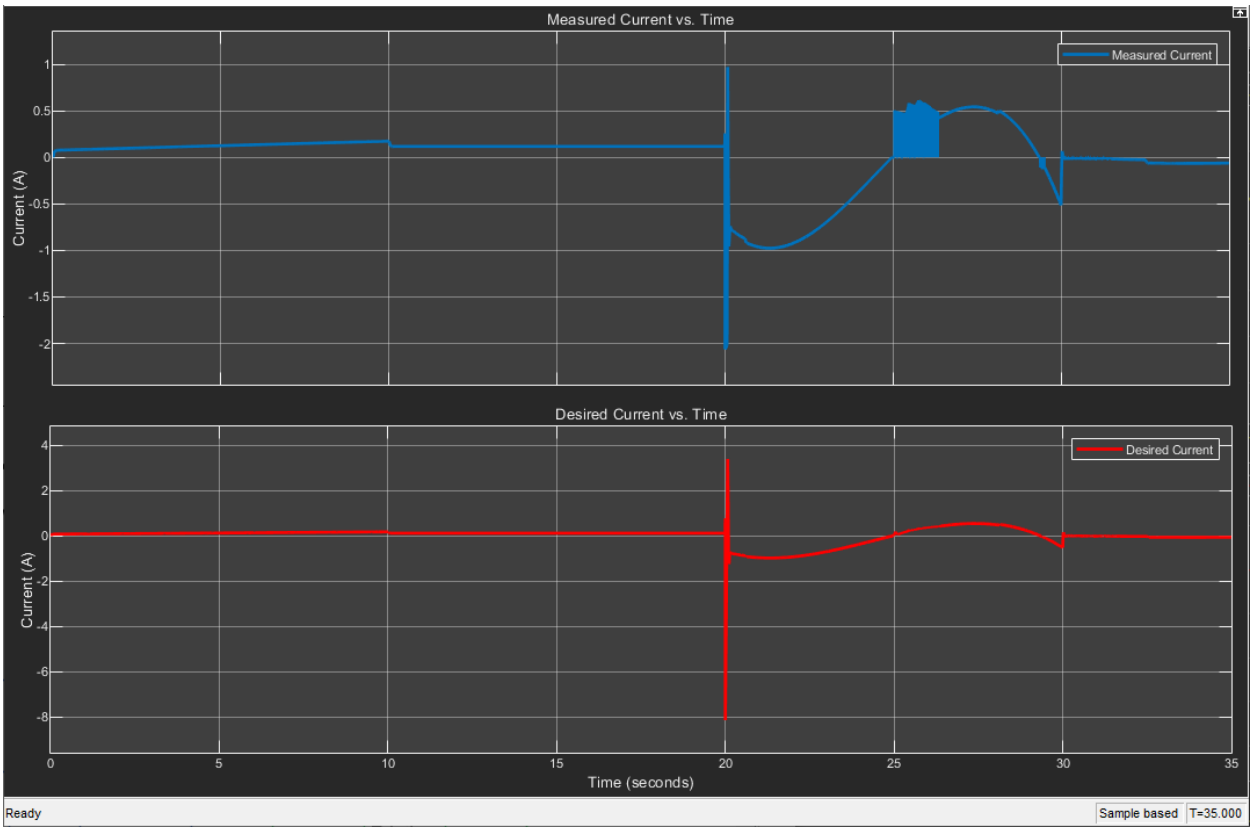


Figure 30: Closed-Loop Current Response

Figure 30 depicts the desired kinematic profile of the closed-loop current response. We tune the current controller parameters as shown in Table 7 based on the discussions in Section 3 and Current Control about the bandwidth frequency and dominance of the electrical time constant.

Table 7: Current Loop controller gains.

Gain	Value
K_p	$2 * \pi * f_s * L$
K_i	$2 * \pi * f_s * R * (1/20)$
K_D	0

Where, $(1/20)$ is the correction factor for the integral controller and f_s is the sampling frequency used in the simulation. We set our sample time to be $\frac{1}{4000}$ and this gave us smooth current, velocity, and position plots.

4.1.2.2. Speed Control

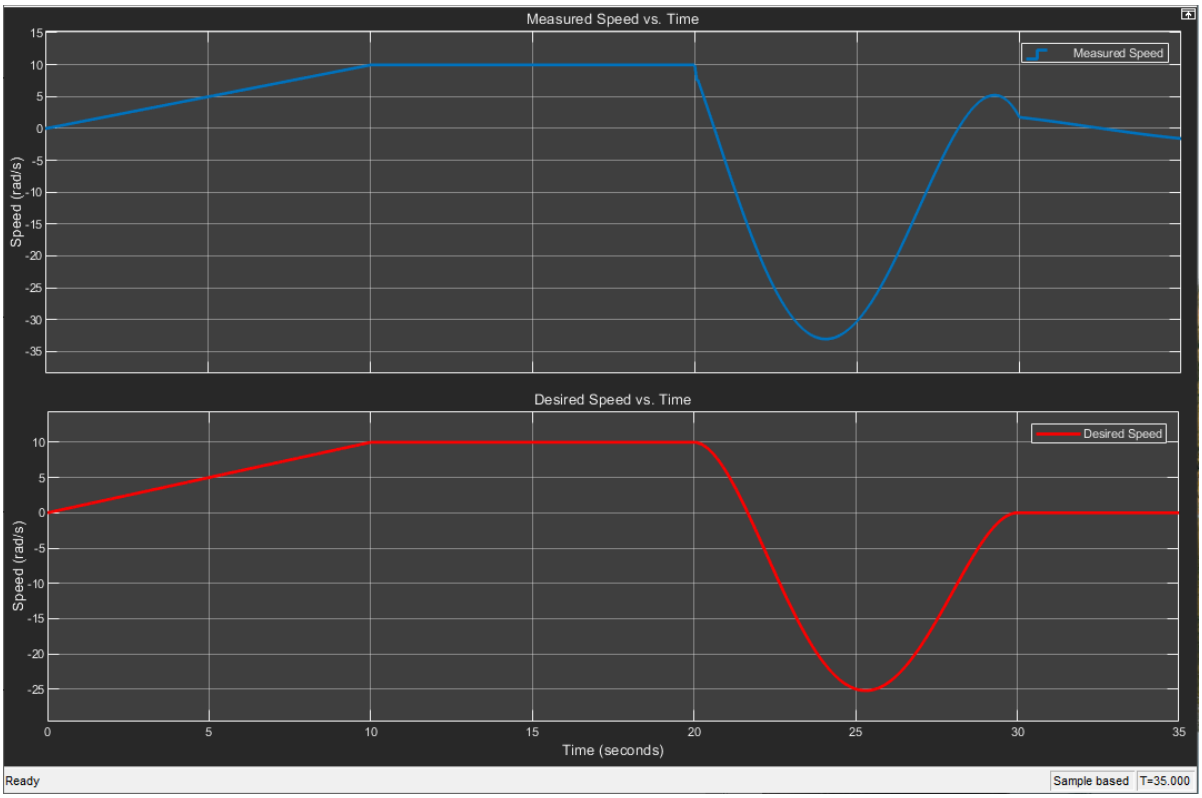


Figure 31: Closed-Loop Speed Response

Figure 31 shows the closed-loop speed response with desired kinematic profile. Based on the discussions in Section 3 and Speed Control section, regarding the bandwidth frequency of this speed loop being orders of magnitude lower than the inner current loop, we tune the current controller parameters as shown in Table 8.

Table 8: Speed Loop controller gains.

Gain	Value
K_p	$2 * \pi * f_s * L * (1/10)$
K_i	0.0132
K_D	0.001

Where, $(1/10)$ is used to reduce the bandwidth of the speed controller by 10 times in comparison to the inner current loop and f_s is the sampling frequency used in the simulation. We set our sample time to be $\frac{1}{4000}$ and this gave us smooth current, velocity, and position plots.

4.1.2.3. Position Control

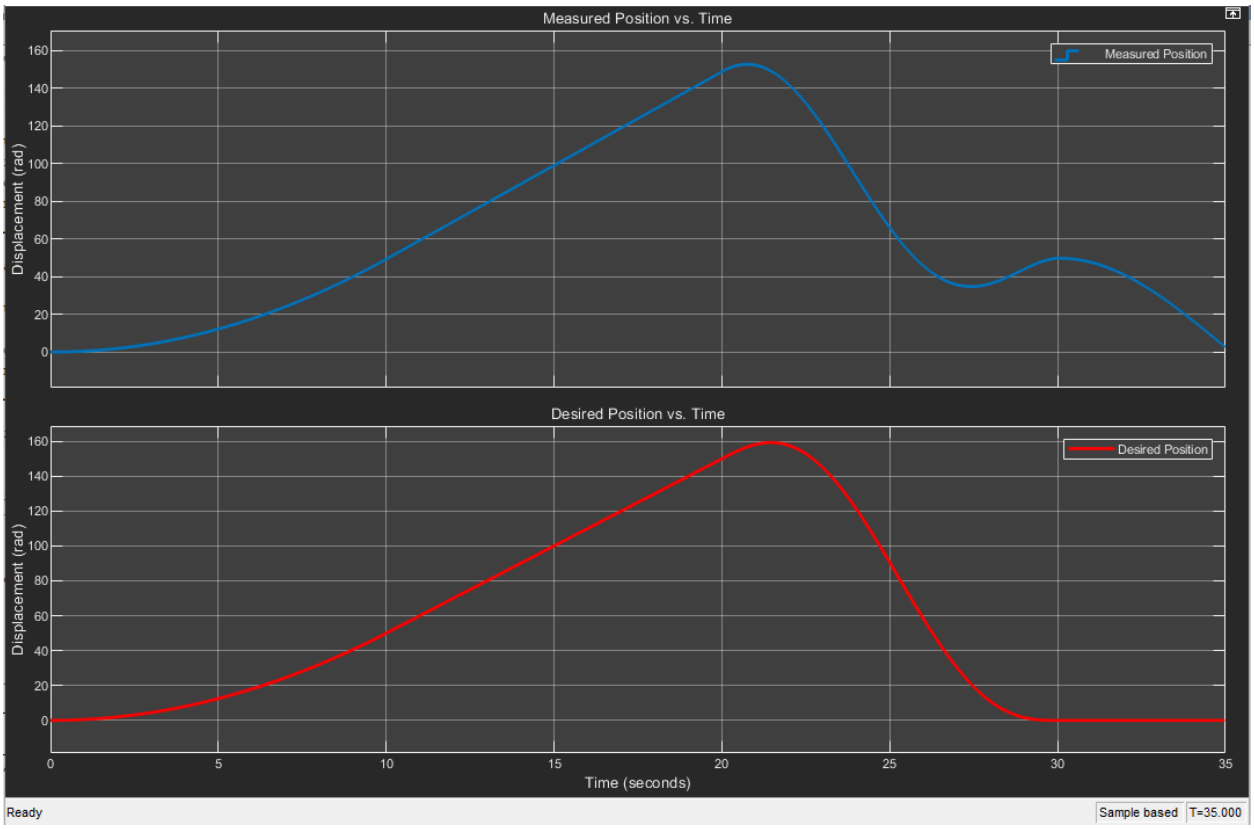


Figure 32: Closed-Loop Position Response

Figure 32 shows the closed-loop position response with desired kinematic profile. Based on the discussions in Section 3 and Position Control section, regarding the bandwidth frequency of this position loop being orders of magnitude lower than the inner speed loop, we tune the current controller parameters as shown in Table 9.

Table 9: Position Loop controller gains.

Gain	Value
K_p	$2 * \pi * f_s * L * (1/100)$,
K_i	0.1
K_D	1

Where, $(1/100)$ is used to reduce the bandwidth of the position controller by 10 times in comparison to the inner speed and current loop and f_s is the sampling frequency used in the simulation. We set our sample time to be $\frac{1}{4000}$ and this gave us smooth current, velocity, and position plots.

The following performance evaluation Section 4.2 contains a detailed discussion of all three controller blocks for both open-loop and closed-loop responses.

4.2. Performance Evaluation

In this section we analyze the plots shown for Open-loop and closed-loop responses of the cascaded current, speed, and position controllers.

4.2.1. Open-Loop Analysis:

Table 10 summarizes the results of the open loop step response.

Table 10: Open Loop Analysis

Controller Block	Proportional Gain (K_p)	Integral Gain (K_i)	Derivative Gain (K_d)
Current Control	1	0	0
Speed Control	1	0	0
Position Control	1	0	0

Figures 27, 28, and 29 clearly show that the system response in open loop is unstable. For starters, there is no correction factor, so the system response is abnormal.

When the desired and measured plots for all three figures are compared, we can see that the actual plots are completely out of bounds in comparison to the desired ones. In Figure 27, the resultant desired current value is so high that the motor is unable to generate it and is cutoff around 1.5A, while the desired current continues to ramp up from 0 to 10 seconds. Due to the provided negative acceleration, the desired current value drops to zero after 20 seconds of retraction, and the measured current gives a negative spike at the instant and then drops to zero, resulting in the motor shutting down.

The kinematic profile generates the desired angular speed for the motor, but because there is no feedback to the speed controller, the motor behaves erratically, resulting in an exponential increase in measured speed for the first 10 seconds. Because the desired speed is constant, the motor speed goes to constant between 10 and 20 seconds; however, both constant values are different because

the controller lacks a feedback loop. Because of the preceding reason, at retraction time, after 20 seconds of negative acceleration, the motor comes to a halt.

4.2.2. Closed-Loop Analysis:

Table 11 summarizes the results of the closed loop step response.

Table 11: Closed Loop Analysis

Controller Block	Proportional Gain (K_p)	Integral Gain (K_i)	Derivative Gain (K_d)
Current Control	2.808	259.8	0
Speed Control	0.2808	0.0132	0.001
Position Control	0.02808	0.1	1

To attain the desired outputs, based on our kinematic profile shown in Figure 25, a feedback loop is used in each controller block to generate the plots shown in Figures 30, 31, and 32.

For the current control plot in Figure 30, the measured current ramps up proportionally to the desired current for the first 10 seconds before becoming constant for the remaining 10 seconds. The measured current closely resembles the desired current profile, with slight positive and negative peaks and overshoots. The desired current peaks between 4 A and -8 A due to the negative acceleration applied at 20 seconds, whereas the measured current peaks between 1 A and -2 A. The controller prevents the motor from overshooting its values, making it more dynamically stable.

According to Figure 31, for the speed profile, the controller performs well for the first 20 seconds (where angular speeds ramp and cruise), but there are undershoots between 20 and 30 seconds. The measured speed lags the desired speed because of these undershoots, and to compensate for this undershoot, the measured speed overshoots at 30 seconds and ramps down to zero, whereas the desired speed goes to zero and remains at zero until 35 seconds (end of simulation). The mechanical vibration and reluctance experienced by the inertial load when the motor shaft moves in the opposite direction cause the undershoot.

According to Figure 32, the position controller responds to the system within the range of desired input for the displacement profile. The measured position follows the desired position with minimal error until the retraction time (i.e., 0 26 seconds) where negative acceleration is applied.

The measured position plot then begins to deviate from the desired position plot, but both eventually return to zero at 35 seconds, when the simulation ends.

5. Conclusion

This report contains a detailed analysis and simulation of the electronics and control design for a one-degree-of-freedom throwing arm. The primary purpose of this report was to integrate our mechanical system with an electronic and control system.

We conducted an in-depth analysis of motor drive circuitry in the first section of this report. Three motors were initially chosen, and their systematic behavior was compared to select the best one for our throwing arm.

The motor we chose was the “GM8224S009”, and it was chosen based on the time it took to reach its steady state value, the current required to generate torque, and the RPM required by the system. Following the selection of the motor, three different cases were investigated to learn more about the motor's behavior with/without heatsink and with different modes of simulation. After simulating all these cases we made following conclusions:

1. In averaged mode system reaches steady state in less time than in PWM mode.
2. Every electronic component generates heat during operation; using a heat sink, which is one of the most effective methods for managing power dissipation, greatly reduces the risk of overheating.

The implementation of a control algorithm on the integrated system was the final part of this project report. In this section, we began with a feedback loop for current control and progressed to a cascade system with velocity and position control. Figure 25 depicts the desired profile for current, position, and speed; these desired profiles served as our system's reference signals. We used a PI and PID controller for our feedback control system, which we tuned after determining the root locus of our desired poles. Despite minor differences, the tuned parameters were effective in incorporating the desired inputs.

6. References

[1] H. DO, H. WORKS and I. Q, "[Explained] H-Bridge Circuit design, Applications, Advantages", ETechnoG - Electrical, Electronics and Technology, 2021. [Online]. Available: https://www.etechnog.com/2019/03/h-bridge-circuit-working-application-advantage.html

[2] P. Jain, "Current Sensors", Engineers Garage, 2021. [Online]. Available: https://www.engineersgarage.com/current-sensors/.

[3] MSE-312 Lectures Mehrdad Moallem.

[4] Moog.com, 2021. [Online]. Available: https://www.moog.com/content/dam/moog/literature/MCG/moc23series.pdf.

[5] "PC280LG-011 datasheet", Datasheetspdf.com, 2021. [Online]. Available: https://datasheetspdf.com/pdf-file/1303801/JohnsonMotor/PC280LG-011/1.

[6] Servocomponents.com, 2021. [Online]. Available: http://www.servocomponents.com/_literature_114806/GM8224S009_PDF.

7. Appendix A

7.1. Motor 1: GM8224S009 Datasheet

Assembly Data	Symbol	Units	Value	
Reference Voltage	E	V	12	
No-Load Speed	S_{NL}	rpm (rad/s)	720	(75.4)
Continuous Torque (Max.)[1]	T_C	oz-in (N-m)	15	(1.0E-01)
Peak Torque (Stall)[2]	T_{PK}	oz-in (N-m)	42	(3.0E-01)
Weight	W_M	oz (g)	11.2	(316)
Motor Data				
Torque Constant	K_T	oz-in/A (N-m/A)	3.09	(2.18E-02)
Back-EMF Constant	K_E	V/krpm (V/rad/s)	2.29	(2.18E-02)
Resistance	R_T	Ω	4.33	
Inductance	L	mH	2.34	
No-Load Current	I_{NL}	A	0.18	
Peak Current (Stall)[2]	I_P	A	2.77	
Motor Constant	K_M	oz-in/√W (N-m/√W)	1.49	(1.05E-02)
Friction Torque	T_F	oz-in (N-m)	0.35	(2.5E-03)
Rotor Inertia	J_M	oz-in-s^2 (kg-m^2)	2.3E-04	(1.6E-06)
Electrical Time Constant	τ_E	ms	0.54	
Mechanical Time Constant	τ_M	ms	14.7	
Viscous Damping	D	oz-in/krpm (N-m-s)	0.020	(1.4E-06)
Damping Constant	K_D	oz-in/krpm (N-m-s)	1.6	(1.1E-04)
Maximum Winding Temperature	θ_{MAX}	°F (°C)	311	(155)
Thermal Impedance	R_{TH}	°F/watt (°C/watt)	70.5	(21.4)
Thermal Time Constant	τ_{TH}	min	10.7	
Gearbox Data				
Reduction Ratio			6.3	
Efficiency[3]			0.95	
Maximum Allowable Torque		oz-in (N-m)	100	(0.71)
Encoder Data				
Channels			3	
Resolution		CPR	500	

1 - Specified at max. winding temperature at 25°C ambient without heat sink. 2 - Theoretical values supplied for reference only.
3 - Effective gearbox efficiency for this unit improved by use of ball bearings.

7.2. Motor 2: C23-L33 W10 Datasheet

Part Number*		C23-L33				
Winding Code**		10	20	30	40	50
L = Length	inches	3.33				
	millimeters	84.6				
Peak Torque	oz-in	125.0	125.0	125.0	125.0	125.0
	Nm	0.883	0.883	0.883	0.883	0.883
Continuous Stall Torque	oz-in	16.5	16.5	16.5	16.5	16.5
	Nm	0.117	0.117	0.117	0.117	0.117
Rated Terminal Voltage	volts DC	12 -24	12 - 24	12 -36	12 - 60	12 - 60
Terminal Voltage	volts DC	12	12	24	36	48
Rated Speed	RPM	4700	2150	4200	3750	3000
	rad/sec	492	225	440	393	314
Rated Torque	oz-in	7.5	12.6	12.7	14.4	15.8
	Nm	0.05	0.09	0.09	0.10	0.11
Rated Current	Amps	4.75	4.3	3	2	1.4
Rated Power	Watts	26.1	20.0	39.5	40.0	35.1
	Horsepower	0.03	0.03	0.05	0.05	0.05
Torque Senstivity	oz-in/amp	2.65	4.25	6.2	10.25	15.75
	Nm/amp	0.0187	0.0300	0.0438	0.0724	0.1112
Back EMF	volts/KRPM	2	3.15	4.6	7.6	11.5
	volts/rad/sec	0.0191	0.0301	0.0439	0.0726	0.1098
Terminal Resistance	ohms	0.60	1.00	1.70	4.00	9.00
Terminal Inductance	mH	0.35	0.94	2.00	5.50	13.00
Motor Constant	oz-in/watt^1/2	3.4	4.3	4.8	5.1	5.3
	Nm/watt	0.024	0.030	0.034	0.036	0.037
Rotor Inertia	oz-in-sec^2	0.0022	0.0022	0.0022	0.0022	0.0022
	g-cm^2	155.4	155.4	155.4	155.4	155.4
Friction Torque	oz-in	3.0	3.0	3.0	3.0	3.0
	Nm	0.02	0.02	0.02	0.02	0.02
Thermal Resistance	°C/watt	6.2	6.2	6.2	6.2	6.2
Damping Factor	oz-in/KRPM	0.1	0.1	0.1	0.1	0.1
	Nm/KRPM	0.001	0.001	0.001	0.001	0.001
Weight	oz	27	27	27	27	27
	g	765	765	765	765	765
Electrical Time Constant	millisecond	0.5833	0.9400	1.1765	1.3750	1.4444
Mech. Time Constant	millisecond	26.07623	17.2056	13.72994	11.82747	11.44547
Speed/Torque Gradient	rpm/oz-in	-113.2075	-74.69655	-59.60729	-51.34788	-49.68944

7.3. Motor 3: PC280LG-011 Datasheet

Specifications:

Dimensions	: Ø 24.4 X 30.8 mm
Shaft Diameter	: Ø 2.000 mm
Input Voltage	: 12.0 V DC
No Load Speed	: 8200 rpm
No Load Current	: 0.65 A
Stall Torque	: 25.50 mNm
Stall Current	: 2.10 A
Maximum Output Power	: 5.50 W
Maximum Efficiency	: 62 %
Speed at Maximum Efficiency	: 7000 rpm
Life (typical)	: 50 hr
Weight	: 52 g
Operation Temperature	: -10 to 55 °C
Storage Temperature	: -20 to 80 °C
Electrical Connection	: Terminal

Performance Data:

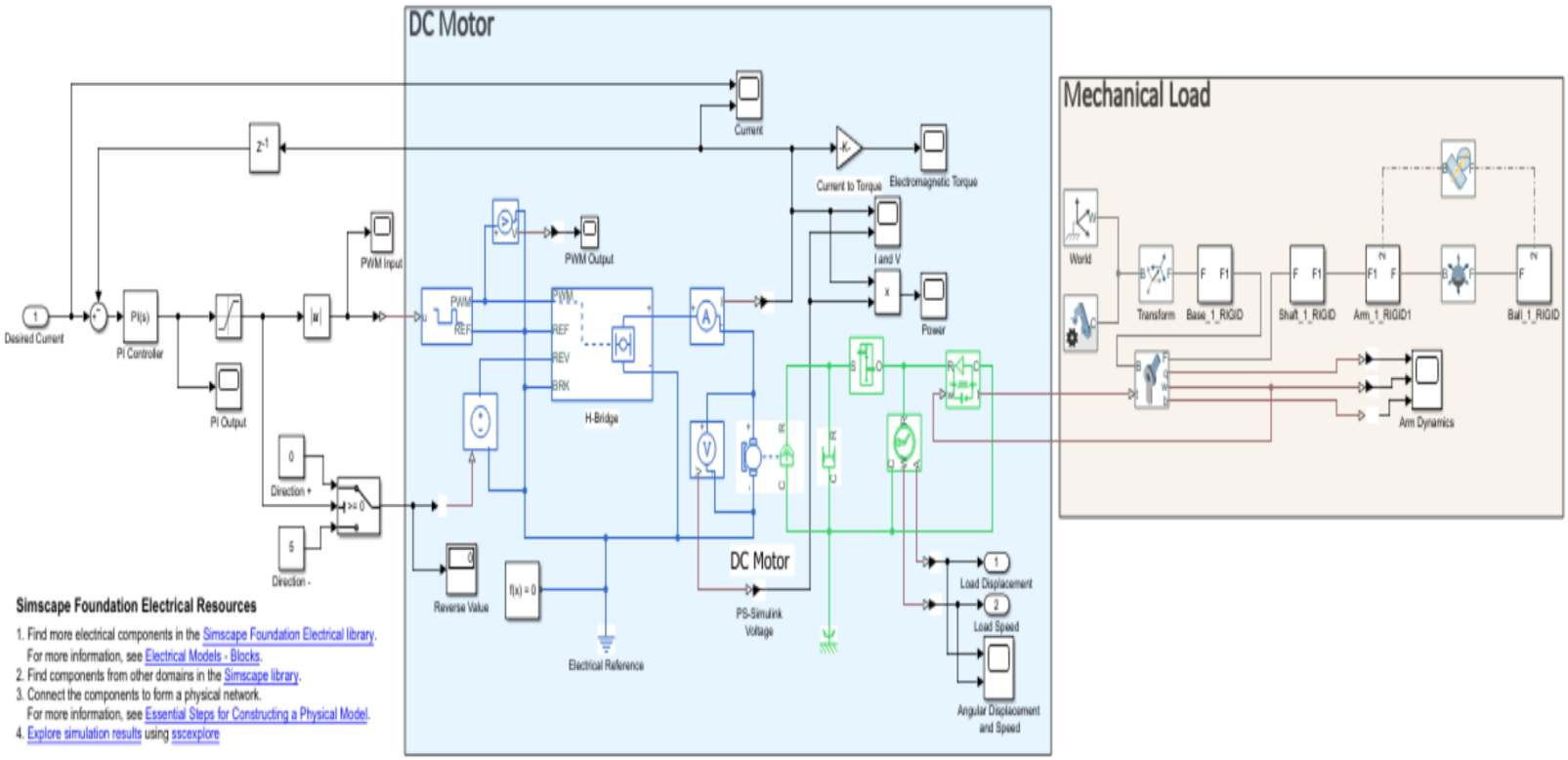
	No Load	Stall	Max Efficiency	Max Power
Current (A)	0.65	2.10	0.37	1.10
Efficiency (%)	-	-	62	-
Output Power (W)	-	-	2.80	5.50
Speed (rpm)	8200	-	7000	4100
Torque (mNm)	-	25.50	3.80	12.80

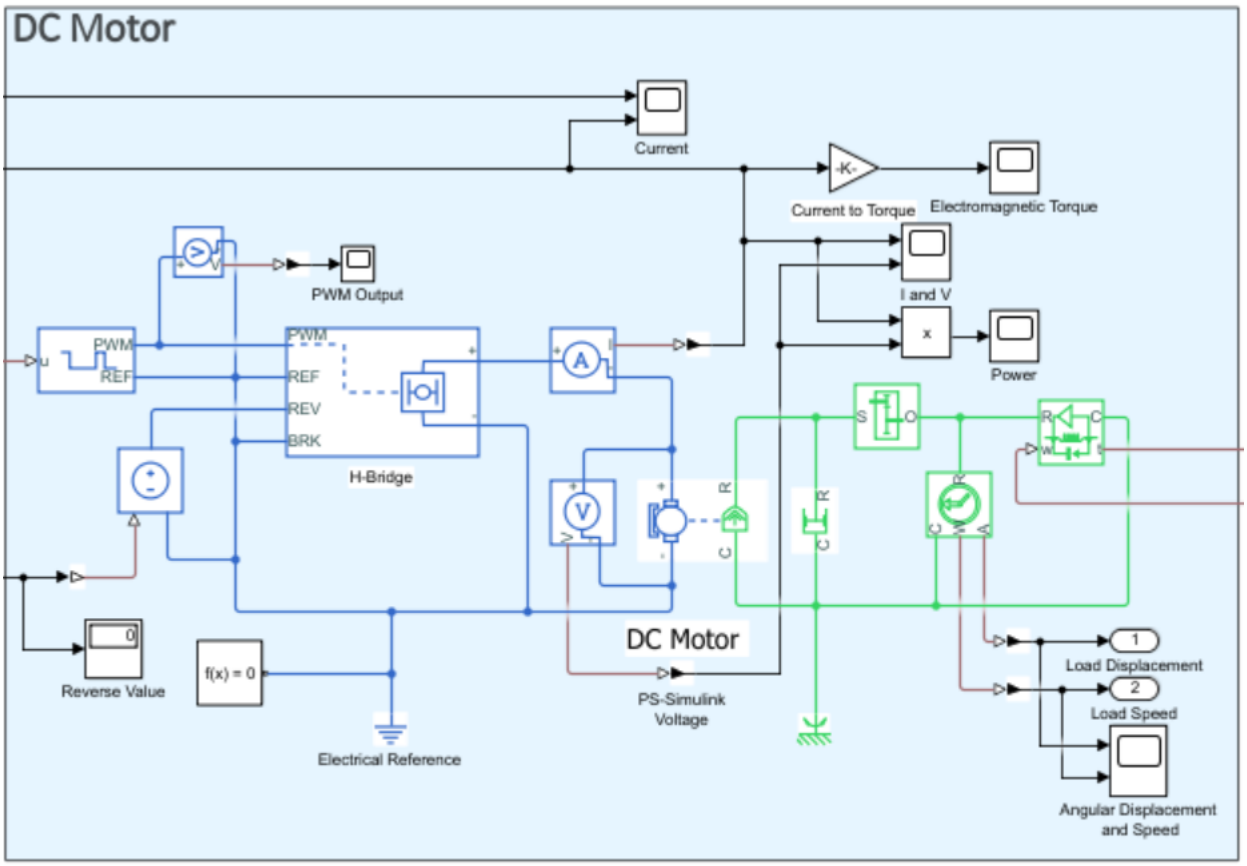
8. Appendix B

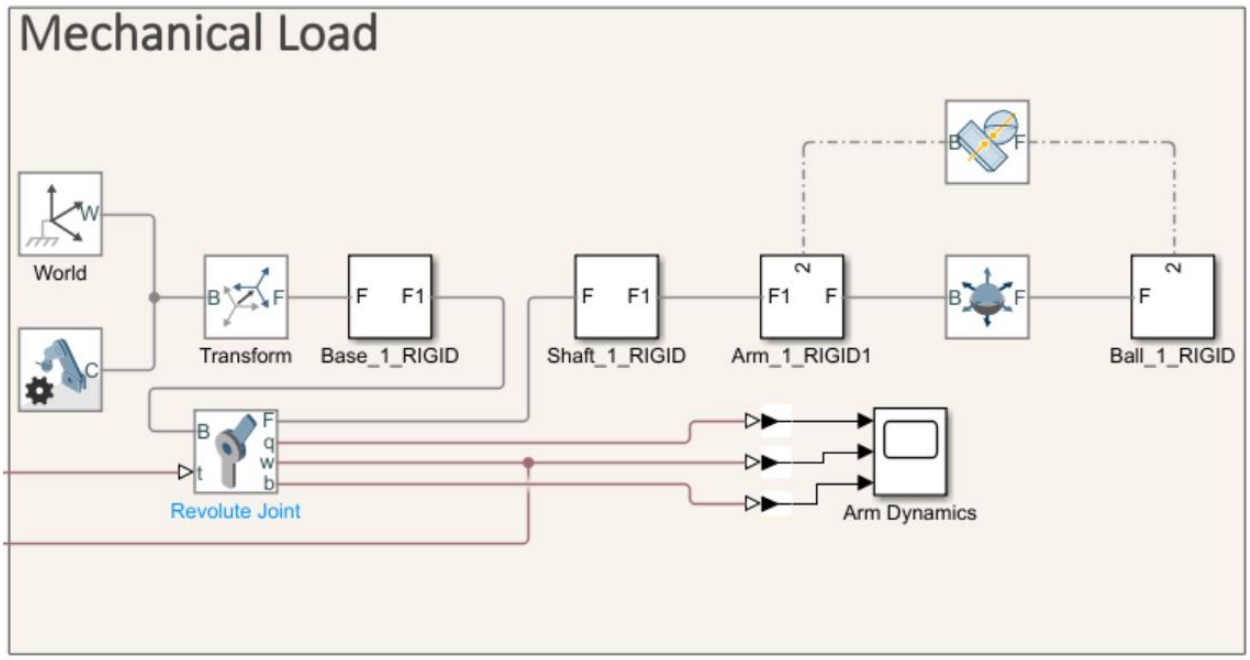
```
function [tau, e_pid, ed_pid, q_d, qd_d, qdd_d] =
fcn(command, pcommand, theta, t, T_ramp, W_cruise,
T_fin, T_ret, a_ret, q_ret, omega)

    if (t <= T_ramp)
        q_d = 0.5 * W_cruise * t^2/T_ramp;
        qd_d = W_cruise * t/T_ramp;
        qdd_d = W_cruise/T_ramp;
        e_pid = qd_d - omega;
        tau = command;
        q_0 = []; q_1 = []; q_2 = []; q_3 = [];
ed_pid = 0;

    elseif (t > T_ramp && t < T_ret)
        q_d = 0.5 * W_cruise * T_ramp + W_cruise *
(t-T_ramp);
        qd_d = W_cruise;
        qdd_d = 0;
        e_pid = qd_d - omega;
        tau = command;
        q_0 = []; q_1 = []; q_2 = []; q_3 = []; q_4
= []; q_5 = []; ed_pid = 0;

    elseif (t >= T_ret && t < T_fin)
        Del = T_fin - T_ret;
        q_2 = a_ret * 0.5;
        q_1 = W_cruise;
        q_0 = q_ret;
        %q_0 = 0.5*(W_cruise/T_ramp)*(T_ramp*2) +
W_cruise*(T_ramp);
        %q_0 = 150;
        ii = inv([Del^3 Del^4 Del^5; 3*Del^2
4*Del^3 5*Del^4; 6*Del 12*Del^2 20*Del^3])*([-q_0-
q_1*Del-q_2*Del^2; -q_1-2*q_2*Del; -2*q_2]);
        q_3 = ii(1); q_4 = ii(2); q_5 = ii(3);
        Del_T = t - T_ret;



        q_d = q_0 + q_1*Del_T +q_2*Del_T^2 +
q_3*Del_T^3 + q_4*Del_T^4 + q_5*Del_T^5;
        qd_d = q_1 + 2*q_2*Del_T + 3*q_3*Del_T^2 +
4*q_4*Del_T^3 + 5*q_5*Del_T^4;
        qdd_d = 2*q_2 + 6*q_3*Del_T +
12*q_4*Del_T^2 + 20*q_5*Del_T^3;
        e_pid = q_d - theta;
        ed_pid = qd_d - omega;
        tau = pcommand;

    else
        q_d = 0;
        qd_d = 0;
        qdd_d = 0;
        e_pid = q_d - theta;
        ed_pid = qd_d - omega;
        tau = pcommand;
    end
end
```

9. Appendix C

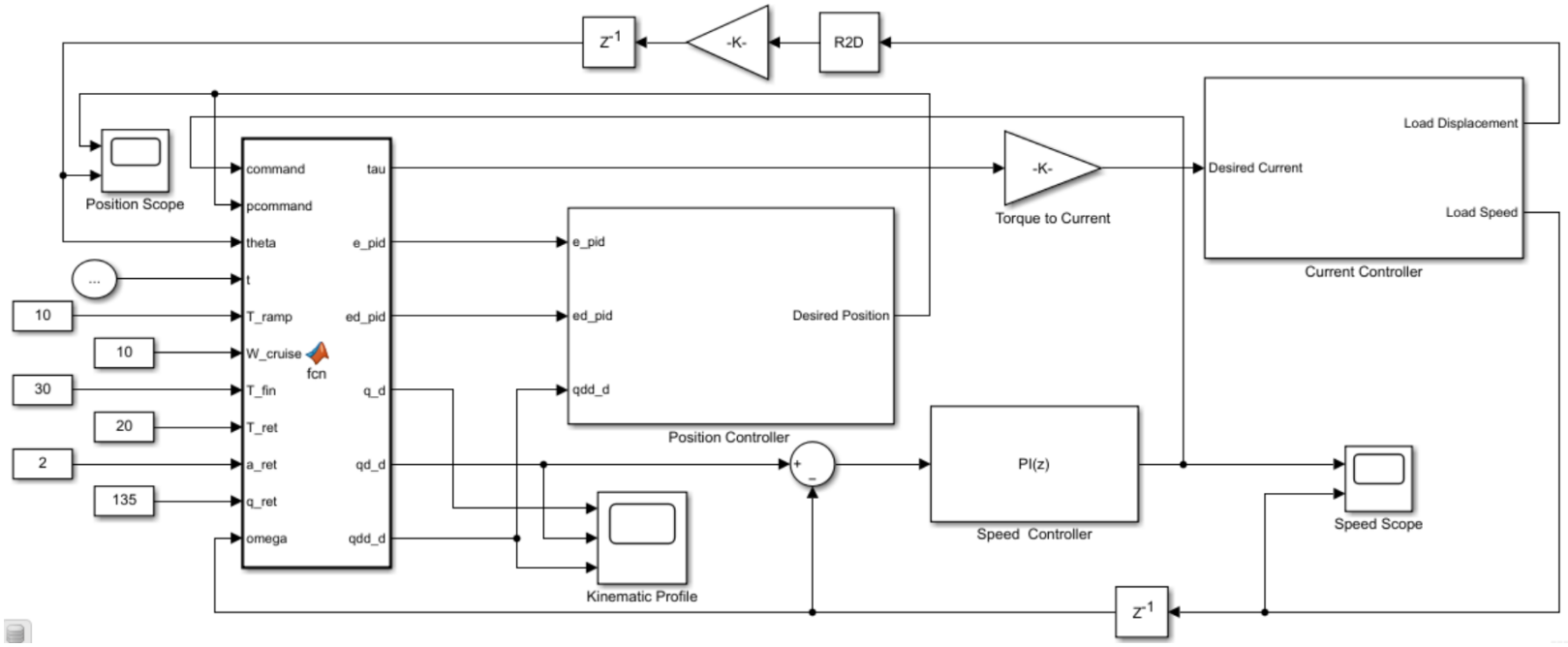
9.1. Electrical Model and Subsystems







9.2. Controller Model and Subsystems



9.3. Position Controller Subsystem