Electric Propulsion Analysis of an Autonomous Racing Quadcopter

a project presented to

The Faculty of the Department of Aerospace Engineering

San Jose State University

in partial fulfillment of the requirements for the degree

Master of Science in Aerospace Engineering

By

Brayan Ernesto Mendez

May 2021

approved by

Dr. Thomas Lombearts

ABSTRACT

Electric Propulsion Mission Analysis of an Autonomous Racing Quadcopter

by Brayan Ernesto Mendez

The thesis details and analyzes the design of an electric propulsion system for an autonomous racing quadcopter application. Results were achieved by open-source hardware and developed software for a driven simulation within the Unity3D environment. A simulation that contains the sized quadcopter vehicle which is representative of the overall design was also recreated to assist in the modeling and development of the analysis. The quadcopter vehicle was successfully modeled using theoretical and analytical methods and compared to the provided performance load test data. Performance characteristics such as static thrust, torque, and current estimates from the analytical modeling approach were shown to be favorable for tuning the flight controller within the Unity3D simulation environment.

ACKNOWLEDGEMENTS

I would like to thank my family for their never-ending support throughout my academic career. Their support has given me the opportunity to pursue higher education and to accomplish my academic goals. I would also like to thank Dr. Nikos Mourtos for guiding and advising me throughout my academic career at San Jose State University for both my undergraduate and graduate degrees. In addition, I would like to thank Dr. Sean Swei for helping me apply theoretical concepts from the UAV Design class into practical applications to this project. I would also like to thank Dr. Thomas Lombearts for advising me throughout this project. Lastly, I would like to thank my close friends who have encouraged me throughout the years.

Table of Contents

List of Tables

List of Figures

NOMENCLATURE

a	=	blade section 2-D lift-curve slope
A	=	rotor disk area
c	=	blade chord line
C_L	=	coefficient of lift
C_D	=	coefficient of drag
C_P	=	power coefficient
C_T	=	thrust coefficient
C_Q	=	torque coefficient
D	=	drag
D_P	=	diameter of the propeller
dr	=	radial increment
F_x	=	sectional aerodynamic force component parallel to disk plane
F_z	=	sectional aerodynamic force component normal to disk plane
I_0	=	no load current
J	=	advance ratio
K_t	=	torque gain
Kv	=	voltage gain
L	=	lift
Q_m	=	torque
r	=	rotor radius
R	=	resistance
T	=	thrust vector
U	=	free stream velocity
U_T	=	in-plane velocity
U_P	=	out of plane velocity
V	=	voltage
v_{∞}	=	freestream velocity
v_i	=	induced velocity
θ	=	pitch angle
α	=	effective angle of attack
ϕ	=	inflow angle
ν	=	induced inflow
σ'	=	local solidity
ρ	=	density of the air
α	=	axial induction factor
φ	=	local inflow angle
$\theta_{.75}$	=	collective pitch angle at 0.75% radius
λ	=	rotor inflow ratio
ω	=	rotor rotational speed

1 Introduction

1.1 Motivation

Autonomous technology is a newly emerging field within aerospace vehicles, such as quadcopters. This technology possesses the ability to control these unmanned aerial vehicles (UAVs) through an onboard computer where specific object detection algorithms are computed and sent to an onboard flight controller to maneuver the quadcopter in six-degree-of-freedom (DOF) space. In a racing application, these computations are performed at extraordinary speeds of up to 32 TeraOPS using state-of-the-art hardware such as the Nvidia Jetson AGX Xavier, which is a type of graphical processing unit (GPU), or supercomputer. Not only does this hardware need to operate at high speeds, but so does the physical quadcopter itself during autonomous flights. Essentially, once real-time calculations are computed and sent to the flight controller, the software sends pulse width modulation (PWM) signals to the electronic speed controllers (ESCs) which modulate the current that is being sent to the motor. The varied current controls the speed of the motor and propeller combination which ultimately creates a force vector perpendicular to the propeller plane. Since a quadcopter has four motors, the vehicle requires a set of four electronic speed controllers to modulate the current for each motor and propeller combination. Nonetheless, an increase in the motor current input results in an increase of the propeller thrust force which also increases the velocity of the physical drone during flight. Figure 1.1 illustrates the general powertrain system architecture for a quadrotor vehicle.

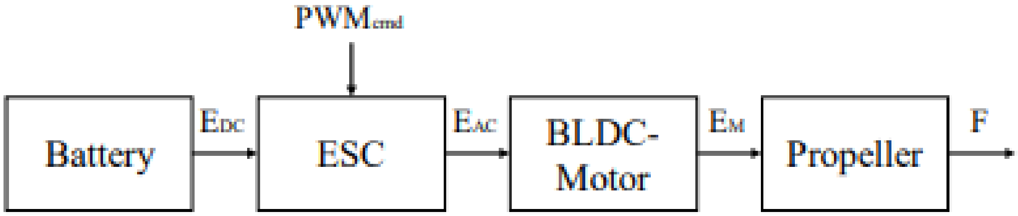

Figure 1.1. Basic powertrain system architecture for a quadcopter UAV [1].

It is important to note that this electric powertrain is illustrated for only one arm of a quadcopter vehicle. Sized correctly, these vehicles can have high thrust to weight ratios but with a tradeoff to low flight times. Moreover, human pilots have demonstrated the high maneuverability of these quadcopter UAVs and the ability to complete dynamically complicated environments, such as racing circuits, at astonishing speeds. In contrast to human-piloted racing drones, autonomous racing quadcopters are far trailing the aspect of piloted performance in the context of robustness, versatility, and speed [2].

The 2019 Lockheed Martin AI Drone Racing Innovation Challenge, hosted by the Drone Racing League (DRL), was a competition focused mainly on artificial intelligence. A fully autonomous drone, programmed by MAVLab, won with the fastest time of 12 seconds in the first part of the competition where 14.6 miles per hour was the highest speed measured throughout the course. The second part of this competition for the winning team was to race a human pilot on the same course. However, the human pilot outperformed the autonomous technology with an overall speed of 24.7 miles per hour and a course completion time of 6.9 seconds which was 4.6 seconds faster than the autonomous racing drone [3]. Despite the loss to a human pilot, the scores of other teams are summarized in Table 1.1 to illustrate the performance of the autonomous technology.

Table 1.1. Summary of team results in the 2019 AI Drone Racing Innovation Challenge.

Rank	**Team**	**Best Heat**	**Time (s)**
1	MAVLab	Finish	00:12
2	UZH Robotics	Finish	00:15
3	Team USRG @ Kaist	Gate 2	00:14
4	RTB - Warsaw Mimotaurs	Gate 1	00:07
5	KEF Robotics	Gate 1	00:08
9	Formula Drone	DNF	~

Similar to a piloted racing drone, autonomous racing quadcopters test the physical limits in the context of course complexity and speed to make progress in state-of-the-art technology [2]. This progress allows autonomous racing technology to advance into the human performance domain. Most quadcopter vehicles discussed in the literature concentrate efforts on powertrain sizing and propeller design. Also, autonomous racing quadcopter design has not been analyzed in the literature to increase the performance of the flight characteristics that have been proposed in this section. Therefore, the main objectives of this project are to analyze a racing quadcopter design to increase performance in speed and provide system monitoring on the electric powertrain throughout the execution of the course.

1.2 Literature Review

1.2.1 Propulsion Design Modeling

A typical performance evaluation for quadcopters consists of propulsion system modeling for various flight stages and measuring performance indices such as maximum-throttle flight and propellers, respectively [4]. A simulated quadcopter is defined by modeling each component of the electric propulsion system, such as the battery, ESC, motor, and propeller, at different stages of the mission. These stages include hovering flight, forward flight, and maximum thrust flight. Problem formulation in [4] includes estimating system efficiency, battery current, ESC input current, ESC input voltage, and the motor speed corresponding to the known variables in the maximum throttle setting. In addition, forward flight modes are analyzed by estimating maximum forward flight speed and the maximum pitch angle corresponding to the known variables [4]. The modeling procedure for this study is shown in Figure 1.2.

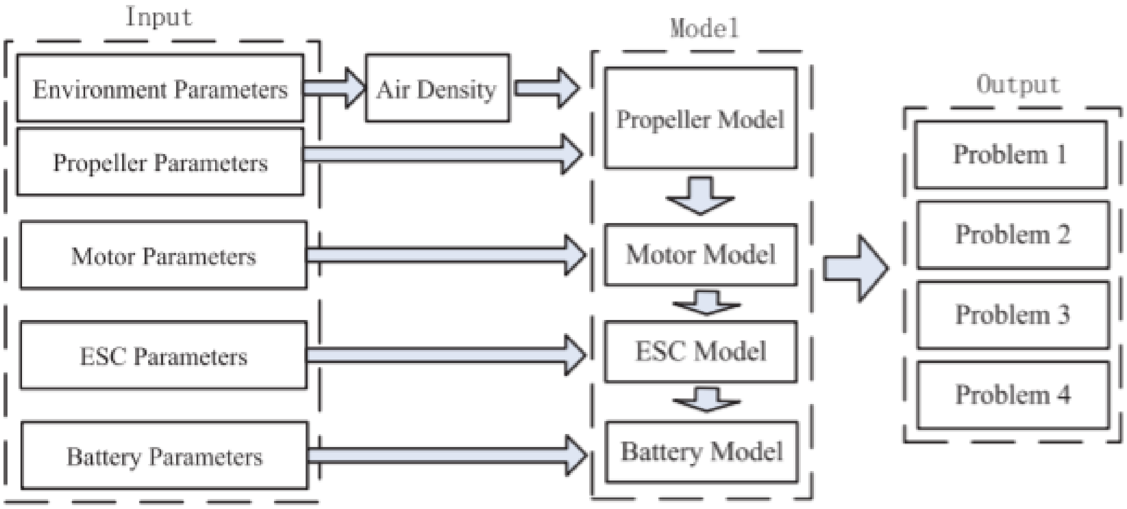

Figure 1.2. Procedure for modeling input-to-output relationship for UAV [4].

Verification of the method effectiveness was performed by means of experimental testing to measure different parameters such as throttle duty cycle and ESC input current. The method used for system evaluation required part specifications provided by component manufacturers [4]. The advantages of this electric propulsion system evaluation method are dominated by the steady state behavior of different flight modes and the energy conversions throughout the mission [4]. It is important to note that for every unmanned aerial platform the propulsion subsystem is the main element.

Identification of the propulsion system, or system identification, is another method used to describe the dynamical model. Simplifications in common modeling approaches have a significant effect on the flight controller design by overlooking critical dynamics [5]. In addition, a block-oriented, nonlinear model is chosen for simulation because aerodynamic and propeller load conditions are nonlinear throughout the modeling procedure [5]. Figure 1.3 illustrates the final electric propulsion model that consists of a multiple input- multiple output (MIMO) system with two inputs and two outputs.

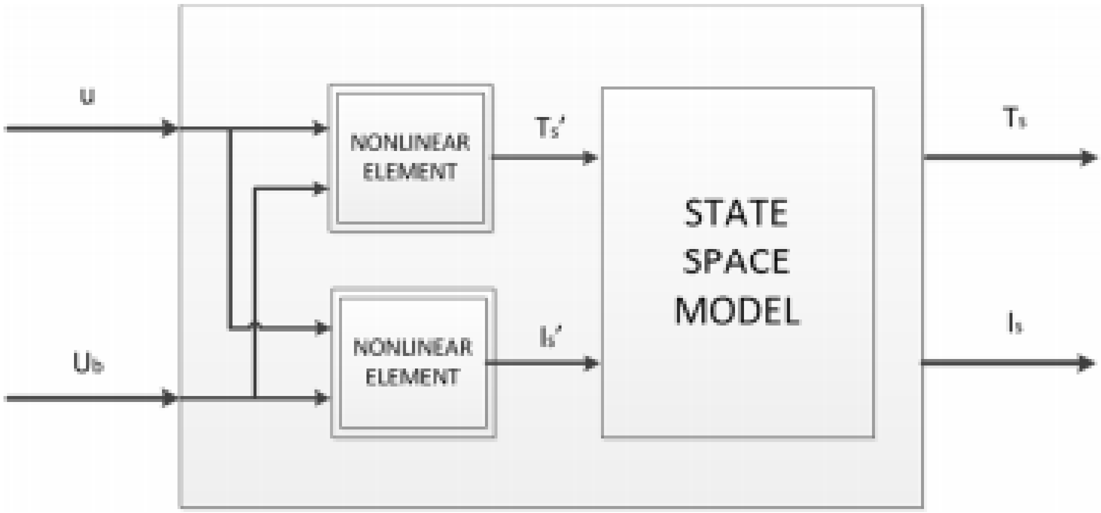

Figure 1.3. MIMO system model of the electric propulsion system [5].

However, this procedure is heavily dependent on experimental testing and experimental data, but the model obtained is a complete depiction of the electric propulsion system [5]. This includes the electronic speed controller, motor and propeller combination, current consumption for each component, and the thrust force vector that makes the system dynamic. One main flight characteristic that can be estimated from this procedure is range. Range estimates are necessary to ensure adequate power is available for return flights throughout the mission [5].

Furthermore, to optimize the propulsion system, modeling is first done using an equivalent circuit as shown in Figure 1.4 to represent the four main elements which are the battery, ESC, motor, and propeller [6]. Essentially, the authors propose a straightforward option in optimizing a propulsion system with a given set of mission requirements. This is done by decoupling and reducing the system into multiple sets of subsystems, or sub-problems. Similar to the previous two methods [4] and [5], the sub-problems are expressed as modeling equations that can be obtained from the equivalent circuit.

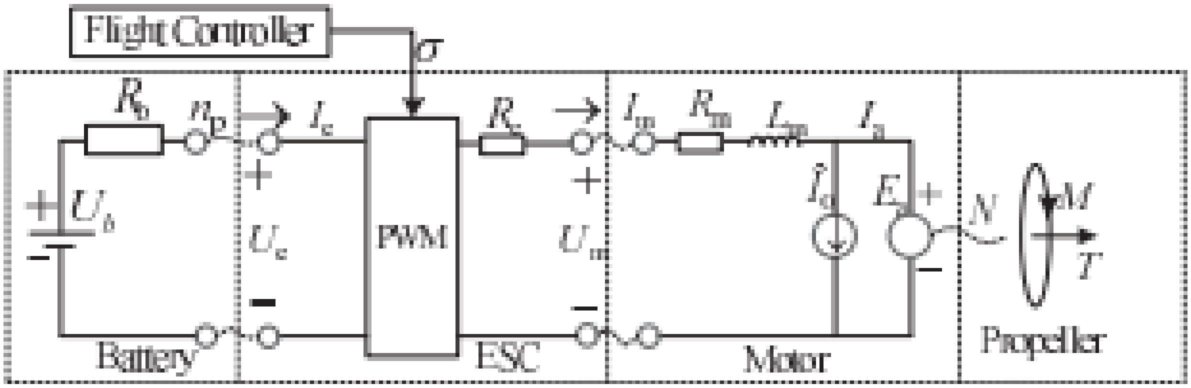

Figure 1.4. Equivalent circuit to model each subsystem [6].

Unlike permanent magnet synchronous motors, brushless direct-current motors use block commutated, trapezoidal signals to drive the system [7]. The motor equivalent circuit in Figure 1.4 can also be described as a linked star connection since the motor requires electric control of three different phases from the electronic speed controller [7]. In a 3-phase circuit connection, current passes through two of the leads at all times. Torque ripples are common with brushless direct current motors when motor position is not synchronized to the electrical pattern of the signals. However, to have minimal discrepancies hall sensors are often used, but most small application motors do not come equipped with them [7]. Nonetheless, converted drive signals from the induced back electromagnetic force (BEMF) is another method in measuring and minimizing the torque ripples. In addition, the electronic speed controller measures the BEMF to also measure the current speed of the motor and propeller combination [8]. Therefore, the BEMF is strictly associated with the torque measurements in a brushless direct current motor application.

1.2.2 Aerodynamic Propeller Performance

Propeller performance is a critical evaluation parameter suitable for efficient characterization and analysis of the electric propulsion system. A common difficulty in propeller performance studies is approximating the induced velocities that are produced by the motor and propeller combinations at various speeds that are experienced throughout the mission. Modeling equations that provide an evaluation of torque and thrust characteristics are used to determine requirements for the quadcopter motor system and to choose proper geometry for the propellers [9]. Due to the comprehensive approach in estimating a level of accuracy that is sufficient in the quadcopter design methodology, blade element momentum theory (BEMT) was used by the authors in [9] to estimate such requirements. From BEMT, a relationship between the force vector that is perpendicular to the propeller plane and the induced velocity produced at the propeller is established [9]. Figure 1.5 illustrates the perpendicular force vector and the induced velocity in the rotating propeller system.

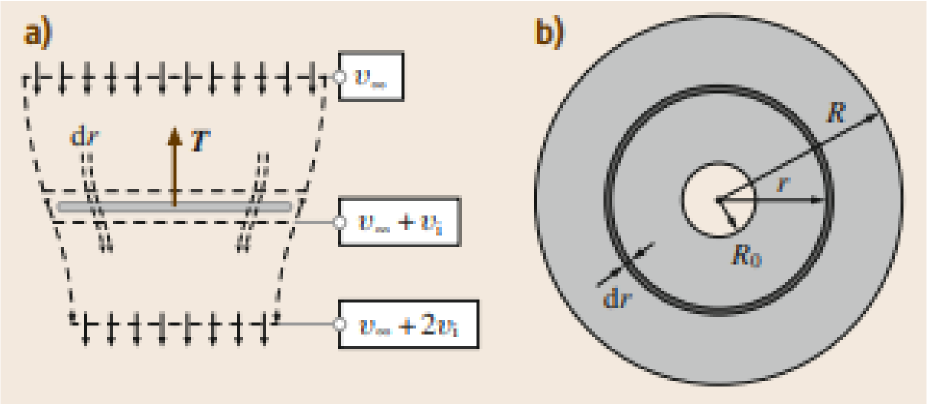

Figure 1.5. Control volume containing the resultant force vector from BEMT [9].

In the illustration above, T is denoted as the thrust vector, v_∞ is the freestream velocity, v_i is the induced velocity, and dr is the radial increment of the propeller. Common assumptions made throughout this modeling procedure consists of having the loading driven by the 2-D airfoil characteristics and having the flow occur in independent streamlines [10]. However, a disadvantage of using BEMT is that it heavily relies on observational data that is not readily available [10]. In addition, other simplifications include inviscid fluid properties and assuming a stationary environment which makes the procedure efficient but less accurate [11]. The main objective from the study done by the authors in [11] was to compare simulation results between BEMT and two higher fidelity computational fluid dynamic (CFD) models, such as actuator line and blade resolved method. Similarities occur between each method of wind fields extracted from an empty domain at the propeller plane, but differences occur at the propeller tip and root [11]. In the context of integral quantities, such as torque and thrust, the standard deviation in simulation results for BEMT is overestimated when compared to the blade resolved method and actuator line method [11]. The authors of this study conclude that similarities in the standard deviation between the two higher fidelity models and the BEMT model can be achieved with an added correction of wind fields in the BEMT model. Nonetheless, the produced thrust and torque can be formulated as described in Equations 1.10 and 1.11, respectively.

$$dT = \sigma' \pi \rho \frac{v_\infty^2 (1+\alpha)^2}{\sin^2 \varphi} (C_L \cos\varphi - C_D sin\varphi) r dr \qquad 1.10$$

$$dQ = \sigma' \pi \rho \frac{v_\infty^2 (1+\alpha)^2}{\sin^2 \varphi} (C_L \sin\varphi + C_D cos\varphi) r^2 dr \qquad 1.11$$

A performance model of a propeller is formulated when combining the analyses of momentum theory and blade element theory as described in [12]. σ' is the local solidity of the blade, ρ is the density of the air, α is the axial induction factor, φ is the local inflow angle measured in radians, C_L is the coefficient of lift, and C_D is the coefficient of drag. It is important to note that this analysis assumes that the size, twist distribution, and airfoil properties are known [12]. This formulated mathematical model is useful in estimating the time-averaged, radial distribution of induced velocities of a propeller with blades of predetermined airfoil characteristics and geometry [13]. However, to have more accuracy in the modeling procedure, additional models are used to account for operating conditions rather than just axisymmetric approaches [14]. These additional models act as corrections to the BEMT and consist of tip corrections, heavily loaded rotor corrections, yaw corrections, and dynamic wake corrections as proposed by the authors in [14]. These act as corrective add ons to the standard BEMT method in addition to the study proposed in [11] to increase the accuracy of the model. One approach to these corrections is the tip loss factor which makes the lift approach zero due to pressure equalization as the radial location on the propeller goes to the tip [15]. A function for the tip loss factor is formulated as a function of the localized correction factor f and shown in Equations 1.12 and 1.13.

$$F = \frac{2}{\pi}\cos^{-1}(e^{-f}) \qquad 1.12$$

$$f = \frac{b}{2}(\frac{1-r}{r\varphi}) \qquad 1.13$$

This relationship was derived from Prandtl where b is the number of blades in the propeller system, r is the radial distance at the section, and φ is the local inflow angle which is estimated through BEMT in an iterative process [15]. In addition, the authors in this study conclude that the tip loss factor correction illustrates a decrease in lift taking place at the tip of the propeller due to the described pressure equalization.

1.3 Project Proposal

Autonomous racing vehicles use carefully selected electric propulsion systems that are efficiently characterized based on mission design requirements to navigate through mapped courses. The main flight characteristic that pertains to the electric propulsion system in this application of interest is speed. Simulation environments are created within Unity3D to test autonomous capability in mapped courses and to monitor system performance throughout mission execution. The literature review discussed in the previous sections proposes methods and techniques used to efficiently characterize electric propulsion systems based on mission requirements and known propeller parameters. Therefore, propulsion system design and analysis are proposed on an existing prototype with monitoring applications in the simulated environment. The objective of this project is to study the aerodynamics of propeller performance at varying command signals and derive empirical relationships for the thrust and torque coefficients as functions of geometry and input power.

Furthermore, aerodynamic performance is critical for any unmanned aerial vehicle application, especially for quadcopter racing technology. To enhance certain flight performance parameters such as speed, parasitic drag acting on the surface of the quadcopter needs to be minimized as the thrust produced by the propellers increases.

1.4 Methodology

To ensure the practicality and feasibility of the existing prototype, further analysis needs to be performed. In addition to the simulated environment, a model of the prototype has been flown in a three-dimensional virtual course and tested for stability control utilizing PhysX engines developed by Nvidia. However, a detailed methodology of the proposed ideas is listed below.

1) Electric Propulsion Selection Based on Mission Requirements and Performance Analysis
    a) Determine mission requirements for autonomous racing quadcopter based on desired speeds
    b) Discuss existing prototype under development and define modeling equations for the six-degree-of-freedom rigid body
    c) Perform a literature survey on propeller performance analysis at varying speeds throughout mission execution on autonomous racing technology
    d) Model existing propulsion system based on literature techniques and approaches such as blade element momentum theory with corrections
    e) Analyze propeller system performance and compare results to simulations
    f) Monitor electrical system parameters such as battery current, battery voltage, ESC current, ESC voltage, and motor input current.
2) Aerodynamic Performance Design and System Monitoring
    a) To reduce parasitic drag on the body of the vehicle a computational fluid dynamics simulation of the quadcopter vehicle will be simulated for an airframe design
    b) Literature review on shape design for quadcopter vehicles in subsonic flows
    c) Comparison between theoretical and experimental results of body aerodynamics such as induced drag

2 Preliminary Design

2.1 Requirements

Electric propulsion and vehicle requirements are defined in this section. It is important to outline and establish requirements in the early phases of the project development. These requirements are constrained to the AlphaPilot 2019 Innovation Challenge as mentioned in previous sections. In addition, requirements pertaining to the hardware are established to meet Federal Aviation Administration (FAA) safety guidelines and regulations for high-speed unmanned aerial vehicles. It is critical to meet the safety requirements before any preliminary testing. The vehicle and electric propulsion system requirements are outlined below.

FAA Safety Guidelines and Regulations [16]:

- Viewer Requirements: The vehicle may not exceed the visual line of sight of the viewer, or pilot. In addition, the viewer must be able to see all surrounding airspace
- Flight: The vehicle cannot be flown over dense, populated areas, near airports, Special Use Airspace, stadiums, Washington, DC, or Security Sensitive Airspace (SSA) unless the airworthiness level allows
- Autonomous: Although autonomous systems consider the pilot to be out-of-the-loop, the vehicle must have pilot intervention capabilities
- Speed: Maximum flight speed for the vehicle is 100 mph (87 knots)

Vehicle Requirements:

- Structure: The frame size is constrained to a 7-inch propeller design; therefore, the minimum frame size is constrained to 330 millimeters by 350 millimeters. The frame sizing also pertains to the other electronics that will be mounted to it, such as the battery, ESCs, and motors.
- Speed: The achievable flight speed is to fall within the minimum and maximum flight speeds that were achieved during the 2019 Innovation Challenge. The flight speed range is between 6-15 miles per hour.
- Weight: The overall weight of the vehicle is constrained to the AlphaPilot requirements set on the standardized drones. The vehicle must not exceed an overall weight of 2.0 kilograms.

Electric Propulsion System Requirements:

- Power to Weight Ratio: A power to weight ratio between 2:1 and 6:1 is constrained by the physical limitations of the object detection hardware such as cameras and inertial measurement unit (IMU) sensors.

Battery: A lithium polymer (Li-Po) battery is used to provide consistent power to the powertrain system.

2.2 Component Section

The components for the electric propulsion system are sized and described in the following sections. As illustrated in Figure 1.1, the main elements of the electric powertrain system are the battery, electronic speed controller, motor, and propeller. A power distribution board is used to distribute power to all four ESCs since an individual battery is utilized.

2.2.1 Propeller Selection

Choosing the correct size propeller is critical in achieving system requirements. Due to the constraints set out by the 2019 AI Innovation Challenge, a propeller diameter of 7 inches was chosen. In addition, a 3-blade propeller system was chosen over a highly common 2-blade system because at lower revolutions per minute the same thrust can be achieved [17]. This allows for a lower amperage draw for the motor but with an increase in thrust. Therefore, polycarbonate 7x4x3 inch propellers are chosen for the propulsion system. Figure 2.1 shows the component.

Figure 2.1. Chosen 7x4x3 inch Polycarbonate propeller.

2.2.2 Motor Selection

The sizing of the motor needs to be well-paired to both the battery and the propeller. Specifically, the K_v rating, or speed constant, is a critical parameter of the motor that is measured in revolutions per minute per volt (RPM/V). In other words, this parameter is an interpretation of how quickly the brushless motors will rotate when started per supplied voltage through the wires [17]. An advantage of higher K_v with lower-pitched propellers is an increase in performance for acrobatic flying with a cost to efficiency [17]. Therefore, due to low weight, a 1380 K_v brushless direct-current motor is chosen as shown in Figure 2.2.

Figure 2.2. 1380 Kv brushless direct-current motor chosen for the vehicle.

2.2.3 Electronic Speed Controller Selection

An essential parameter for sizing the ESC is the maximum rated amperage because this value must exceed the required input current for each individual motor to prevent the field-effect transistors (FET) from overheating [17]. Essentially, as mentioned in previous sections, the ESC receives PWM signals and transforms them to AC current. Embedded FETs switch the frequency which rotates the rotor. Therefore, with the selection of the 1380 K_v motor with a maximum current draw of 43.7 Amps, 60-Amp ESCs were chosen for the vehicle. Figure 2.3 shows the component.

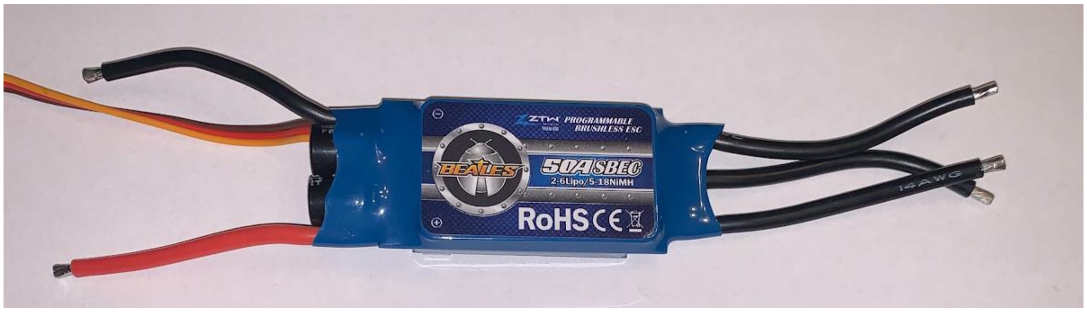

Figure 2.3. 60-Amp Brushless ESC chosen for the vehicle.

2.2.4 Battery Selection

To properly size the battery for the given requirements, the battery capacity and cell configuration are considered. To supply sufficient voltage to the electric propulsion system, batteries are built in either series (S) to increase the voltage or in parallel (P) to increase the capacity [17]. In addition, the maximum charged voltage of the battery pack must agree with the selections of the propeller and speed constant of the motor to produce adequate thrust and RPM values to effectively lift the UAV [17]. Therefore, a 5-S, 5000 mAh battery with a maximum charge voltage of 18.5 V was chosen as the main power supply of the vehicle and is shown in Figure 2.4.

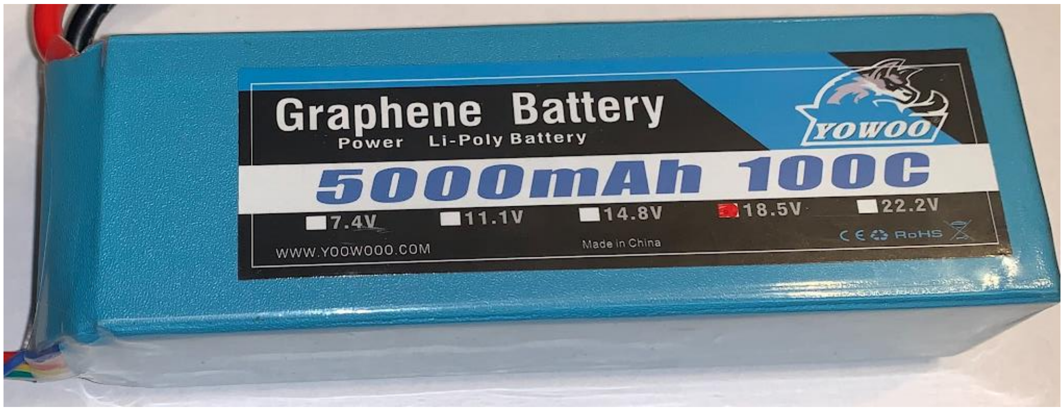

Figure 2.4. 5-S, 5000 mAh Li-Po battery chosen for the main power supply.

2.2.5 Sensor Selection

Three main sensors are used onboard the vehicle that feed data to the flight controller: an IMU, ultrasonic sensor, and stereoscopic camera suite. The Adafruit BNO055 IMU is chosen because it comes equipped with a nine degree-of-freedom (DOF) sensor and 3 accelerometers to measure angular acceleration and orientation. The SainSmart HC-SR04 ultrasonic sensor is chosen due to its performance within a 2-meter forward range. Lastly, the Arducam BO200 stereoscopic camera is chosen for the use of the onboard computer vision and there are two of them. They face directly forward and are mounted on the topside of the vehicle.

2.2.6 Flight Controller Selection

Two single-board computers (SBCs) are used in combination for the onboard computer vision and the flight controller which are the Nvidia Jetson Nano and Raspberry Pi 4 Model B+, respectively. The 128-core Maxwell GPU that is integrated within the Jetson Nano is optimized for Machine Learning and video processing which requires a 5-Volt power supply. Moreover, the Raspberry Pi 4 Model B+ acts as the controller and sends PWM signals to spin the motors. General-purpose input-output (GPIO) pins are used to send the signals. Similar to the Nano, the Raspberry Pi also requires a 5-Volt power supply.

2.2.7 Structural Frame Design

In contrast to a traditional x-configuration frame, a radical t-configuration is chosen to provide a non-obstructed face view to the vehicle’s object detection software. This allows the placement of two forward-facing cameras on the top side of the frame at the forward most position. The model was designed using PTC CREO Parametric and is illustrated in Figure 2.5.

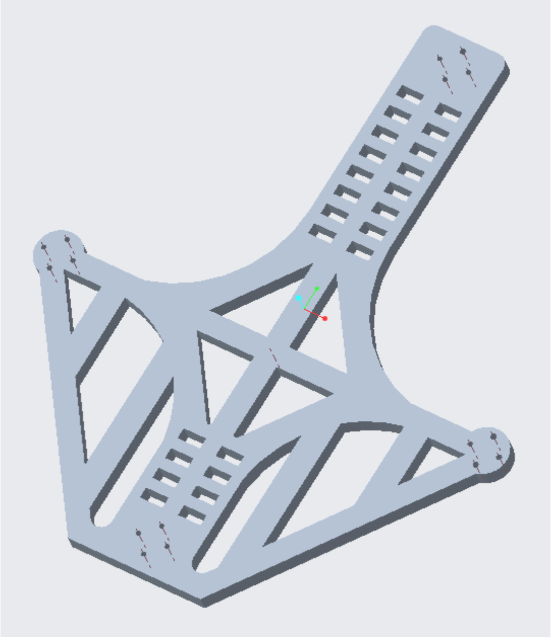

Figure 2.5. 3D CAD Model of frame design showing a t-configuration.

From the 3D CAD model, the actual length from motor center to opposite motor center is 365.57 (14.4 inches) and 319.31 (12.60 inches) for front to back motor and left to right motor, respectively. As modeled, the frame meets the minimum frame requirement that pertains to a 7-inch propeller. It is important to note that the frame has a uniform thickness of 0.5 inches to securely mount hardware. The material of choice is polylactic acid (PLA) which is commonly used in additive manufacturing processes.

2.3 Weight Estimation

The total estimated weight of the aircraft is discussed here. A methodology is used to separate various subsystems which include the electric propulsion system, flight control system, and structural system. The electric propulsion system weight consists of the battery, motors, ESCs, and propellers and is defined by Equation 2.3.1.

$$W_{EP} = W_{battery} + (W_{ESC}x4) + (W_{motor}x4) + (W_{propeller}x4) \qquad 2.3.1$$

The flight control system consists of hardware that uses machine learning and computer vision to control the behavior of the UAV, such as both stereoscopic cameras, sensors, Raspberry Pi 4 Model B+ flight controller, and the Nvidia Jetson Nano. Equation 2.3.2 defines the overall weight of this system.

$$W_{FC} = W_{Raspberry} + W_{Nano} + W_{sensors} + W_{cameras} \qquad 2.3.2$$

The structural system consists of the frame and all mounting hardware such as bolts, nuts, adhesive strips, and brackets. This system can be described by Equation 2.3.3.

$$W_S = W_{hardware} + W_{frame} \qquad 2.3.3$$

The overall weight of the system can be estimated using Equation 2.3.4. In addition to estimating the overall weight, this methodology is useful in determining subsystem weight for the vehicle.

$$W = W_{EP} + W_{FC} + W_S \qquad 2.3.4$$

By using the methodology described above and Table 2.1, the overall weight of the vehicle is 1846.7 grams. It is important to note that the specific weights of each component were measured using a weight scale with a resolution of 0.05 grams. Nonetheless, the specific weight for the electric propulsion, flight control, and structural systems are 1181, 331.7, and 334, respectively.

Table 2.1. List of component weights.

Component	Quantity	Weight (g)
Frame with PDB	1	334
ESC	4	62
Battery	1	677
HQ 7x4x3 Propeller	4	10
TBS Crossfire Nano RX Receiver	1	2
SainSmart HC-SR04 Ultrasonic Sensor	1	8.7
BNO550 IMU	1	3
Avenger Series Motor	4	54
Arducam B0200 Stereoscopic Camera	2	65.5
Nvidia Jetson Nano	1	140
Raspberry Pi 4 Model B+	1	47
	Total	1846.7

2.4 Blade Element Theory

For theoretical analysis, Blade Element Theory (BET) is utilized for the rotor in hovering and climbing conditions. BET is mainly utilized to estimate the theoretical forces of a blade that correspond to its varying rotation through the air which are used to calculate the performance and forces over the rotor. It is important to note that lifting-line theory (LLT) applied to a spinning wing is the foundation for BET [18]. The rotor is split into a number of infinitesimal segments in BET where the lift and drag can be estimated using two-dimensional airfoil characteristics at each segment based on local flow data. This data includes inflow, climb speed, and the angular velocity of the rotor. To obtain the power consumed and thrust produced by an individual rotor, both the drag and lift are integrated from the root of the airfoil to the tip [18]. Furthermore, to obtain the total power required and total thrust available, both equations are multiplied by the number of blades in the lift producing system. Essentially, every infinitesimal segment along the chord line of the blade experiences an in-plane velocity that is denoted by U_T that is tangent to the rotation plane. Figure 2.6 illustrates the forces, velocities, and geometry of the sectional blade. In a hovering condition, the rotational speed of the rotor is defined as the tangential velocity [18]. In addition, Figure 2.6 shows that every infinitesimal segment has a specific pitch angle with respect to U, the free stream velocity, that is denoted by θ. The effective angle of attack, α, and the inflow angle, ϕ, are affected by induced inflow, ν, and the climb velocity, V, respectively [18]. These variables are defined below by Equations 2.4.1 and 2.4.2.

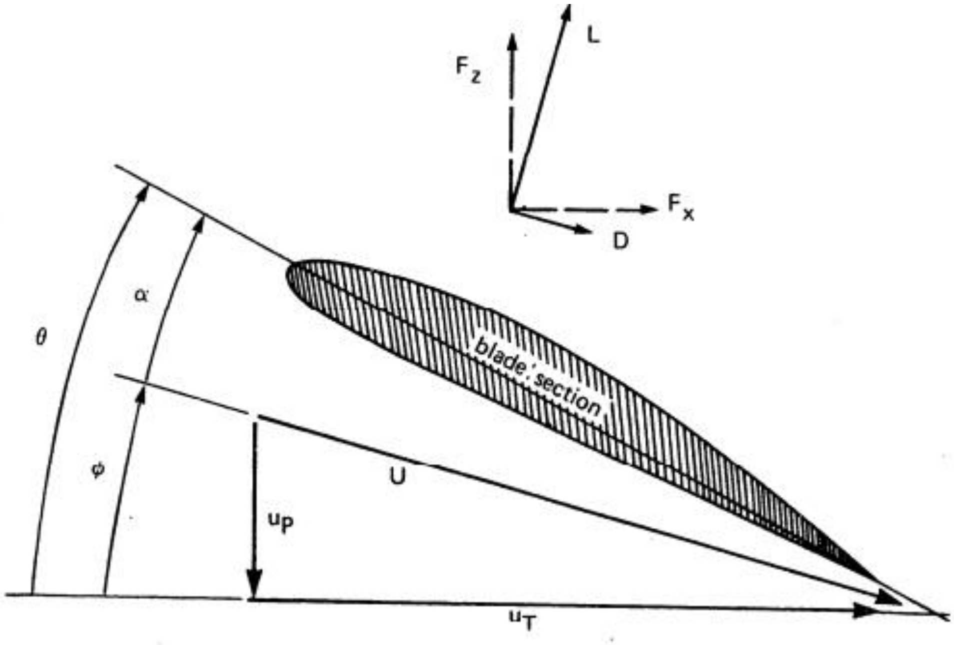

Figure 2.6. Blade segment aerodynamics.

$$\Phi = \arctan\left(\frac{V}{\omega r}\right) \tag{2.4.1}$$

$$\alpha = arctan\left(\frac{V}{\omega r}\right) \tag{2.4.2}$$

The magnitude of the resultant velocity with components of U_T and U_P that the airfoil experiences can be defined by Equation 2.4.3. The inflow angle can also be expressed as functions of the velocity components as defined in Equation 2.4.4.

$$U = \sqrt{U_T^2 - U_P^2} \tag{2.4.3}$$

$$\Phi = \tan^{-1}\left(\frac{U_P}{U_R}\right) \tag{2.4.4}$$

The closed-form equations of the lift and drag are defined by Equations 2.4.5 and 2.4.6, respectively, where C_l is computed by a simplified lift expression of $C_l = a\alpha$ and $\alpha = \theta - \phi$. The lift and drag forces, as shown in Figure 2.4, are normal to and parallel to the resultant velocity [19].

$$L = \frac{1}{2}\rho U^2 c C_l \tag{2.4.5}$$

$$D = \frac{1}{2}\rho U^2 c C_d \tag{2.4.6}$$

The density of air and chord line of the blade are denoted by ρ and c, respectively. Simplified forms of C_l and C_d, which are the sectional coefficients, are used since in general they are functions of the Mach number and angle of attack [19]. F_x and F_z are the elements of the overall aerodynamic force that are parallel and normal to the blade disk plane which are defined by Equations 2.4.7 and 2.4.8, respectively.

$$F_z = Lcos(\Phi) - Dsin(\Phi) \qquad 2.4.7$$

$$F_x = Lsin(\Phi) - Dcos(\Phi) \qquad 2.4.8$$

Multiplying by the number of blades, N, and differentiating with respect to the blade radius, r, gives expressions for the elemental power, torque, and thrust on the blade. Furthermore, to obtain the complete forces acting on the rotor, these expressions are integrated along the span of the blade from root to tip [19].

$$dT = NF_z dr \qquad 2.4.9$$

$$dQ = NF_x rdr \qquad 2.4.10$$

$$dP = \omega dQ = NF_x \omega rdr \qquad 2.4.11$$

In hovering and vertical flight conditions, the velocity vector normal to the disk plane, U_P, is in terms of the induced velocity, v_i, and the climb velocity which is equal to zero at hover. In addition, U_T is due to a rate ω that the blades are rotating at. Thus, $U_T = \omega r$ and $U_P = (V+v_i)$. Assumptions and simplifications are made to reduce the sectional forces of the blade and are listed below [18][19]:

- Small-angle assumptions are made where $\boldsymbol{\alpha} << 1$, $\theta << 1$, and $\phi << 1$
- Compressibility effects are negligible
- Stall effects are negligible
- The angle of attack and lift coefficient are linearly related
- Blade tip losses are neglected
- The rotor is twisted linearly
- C_d is small

The segment forces acting on the blade are then reduced and defined by Equations 2.4.12 through 2.4.15.

$$L \cong \frac{1}{2}\rho U_T^2 ca(\theta - \frac{U_P}{U_T}) \qquad 2.4.12$$

$$D \cong \frac{1}{2}\rho U_T^2 cC_d \qquad 2.4.13$$

$$dT \cong NLdr \qquad 2.4.14$$

$$dQ = N(L\Phi + D)rdr \qquad 2.4.15$$

Normalization with respect to the rotor radius, rotor speed, and local air density converts all terms into dimensionless quantities [19]. Equations 2.4.16 and 2.4.17 are the sectional thrust and power equations in coefficient form where the solidity ratio $\sigma = Nc/\pi r$ and the inflow ratio $\lambda = (V+v)/\Omega r$.

$$dC_T = \frac{\sigma a}{2}(\theta U_T^2 - U_T U_P)dr = \frac{\sigma a}{2}(\theta r^2 - \lambda r)dr \qquad 2.4.16$$

$$dC_P = dC_Q = \left[\frac{\sigma a}{2}(\theta U_T U_P - U_P^2) + \frac{\sigma C_d}{2}U_T^2\right]rdr \qquad 2.4.17$$
$$= \left[\frac{\sigma a}{2}(\theta r\lambda - \lambda^2) + \frac{\sigma C_d}{2}r^2\right]rdr$$

With constant drag coefficient, constant chord, uniform flow, and other simplifying assumptions, an analytical solution may be achieved for a general rotor system when integrated over the span of the blade [19]. BEM thus gives an expression for rotor thrust coefficient as defined in Equation 2.4.18 in integral form. With the underlying assumptions of $\theta = \theta_0 + r\theta_{tw} = \theta_{.75} + (r - 0.75)\theta_{tw})$ and λ is constant, Equation 2.4.18 becomes 2.4.19.

$$C_T = \int_0^1 \frac{\sigma a}{2}(\theta r^2 - \lambda r)dr \qquad 2.4.18$$

The pitch of the blade with constant chord, $\theta_{.75}$, is defined as the pitch of the blade at 75% of the radius [19].

$$C_T = \frac{\sigma a}{2}(\frac{\theta_{.75}}{3} - \frac{\lambda}{2}) \qquad 2.4.19$$

The induced velocity is given by BET and defined in Equation 2.4.22 where it is expressed as a function of the thrust coefficient and the climb inflow ratio for both climbing and hovering conditions. In hover, the induced velocity expression can also be expressed in terms of the blade pitch angle only if the thrust coefficient is required and is defined in Equation 2.2.23. Using Equation 2.4.24, the blade pitch angle can be defined.

$$v_i = \frac{\lambda_c}{2} + \sqrt{\left(\frac{\lambda_c}{2}\right)^2 + \frac{C_T}{2}} \qquad 2.4.22$$

$$v_i = \sqrt{\left(\frac{C_T}{2}\right)} = \frac{\sigma a}{16}[\sqrt{1 + \frac{64}{3\sigma a}\theta_{.75}} - 1] \qquad 2.4.23$$

$$\theta_{.75} = \frac{6C_T}{\sigma a} + \frac{3}{2}\sqrt{\frac{C_T}{2}} \qquad 2.4.24$$

Once integrated, the differential power coefficient, dC_P, can be expressed as shown in Equation 2.4.25. The first and second terms in this expression are defined as the induced power loss and the profile power loss, respectively [19]. Due to the induced angle of attack and viscous drag forces, the component of lift that is in-plane gives rise to the induced power loss and the profile power loss, respectively [19]. In Equation 2.4.26, k and C_{do} are an empirical correction factor and a proper mean drag coefficient [19].

$$C_P = \int \lambda dC_T + \int_0^1 \frac{\sigma C_D}{2} r^3 dr \qquad 2.4.25$$

$$C_P = \frac{kC_T^{\frac{3}{2}}}{\sqrt{2}} + \frac{\sigma C_{D_0}}{8} \qquad 2.4.26$$

Combining Momentum theory with Blade Element Theory, the coefficient of thrust, power, and torque can be expressed as a function of the nondimensional velocity, J, which is the advanced ratio in Blade Element Momentum Theory (BEMT) [21]. Equations 2.4.27 through 2.4.30. shows the relationships.

$$C_T = \frac{\pi}{8}\int_{x_b}^1 (J^2 + \pi^2 x^2)\sigma[C_l \cos(\Phi + \alpha_i) - C_d \sin(\Phi + \alpha_i)]dx \qquad 2.4.27$$

$$C_P = \frac{\pi}{8}\int_{x_b}^1 \pi x(J^2 + \pi^2 x^2)\sigma[C_l \sin(\Phi + \alpha_i) + C_d \cos(\Phi + \alpha_i)]dx \qquad 2.4.28$$

$$C_P = 2\pi C_Q \qquad 2.4.29$$

$$J = \frac{U}{nD} \qquad 2.4.30$$

In addition, the efficiency of the propeller can be expressed as a function of these coefficients and the advance ratio. It is important to note that an increase in the advance ratio results in an increase in the total propeller efficiency as described by the linear relation defined by Equation 2.4.31 [21].

$$\eta = \frac{C_T J}{C_P} \qquad 2.4.31$$

3 Modeling

The modeling of the autonomous unmanned aerial vehicle is derived in this section. It is important to understand the dynamics of the drone and the behavior it experiences from the thrust and torque produced from various motors.

3.1 Equations of Motion

To model the dynamics of a quadcopter and understand how various forces act on the body, it is necessary to derive the equations of motion that express the vehicle. The motion of the quadcopter can be expressed through a set of non-linear differential equations. These equations create difficulty in applying controllers and accurate simulation for the quadcopter system. The first step in deriving the equations of motion of the quadcopter system is defining sets of coordinate systems since the vehicle is subject to the concept of variable thrusts and torques. Figure 3.1 illustrates both the inertial and body-fixed frames used to determine the orientation of the vehicle. Essentially, the motor and fixed-pitch propeller combinations are organized in pairs along both the vertical and horizontal axis.

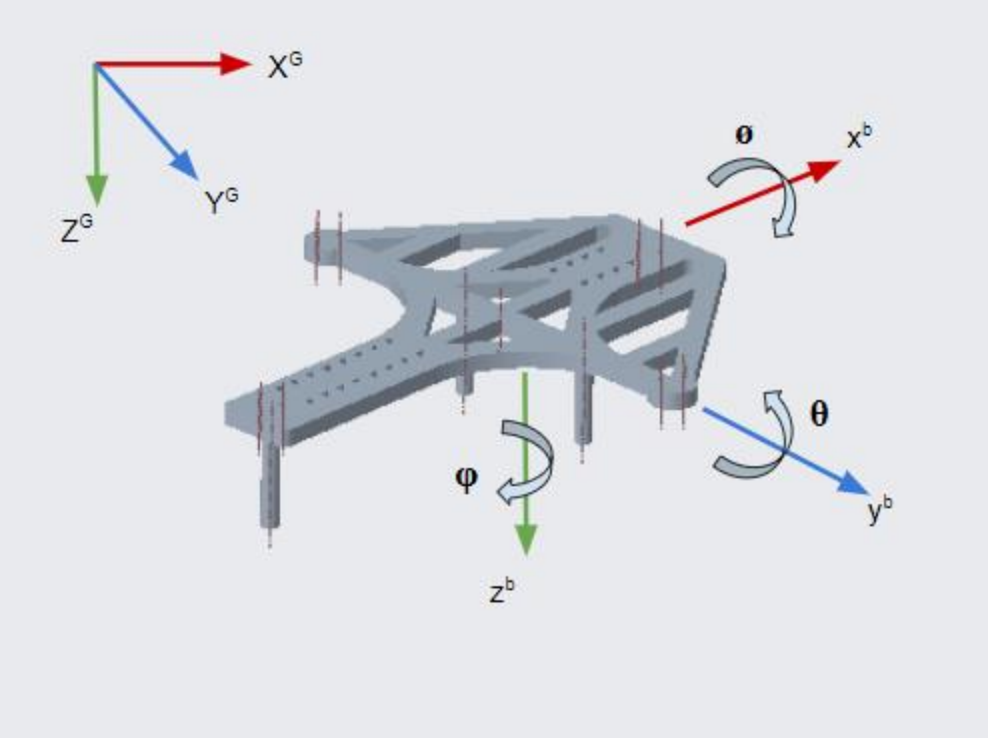

Figure 3.1. Quadcopter coordinate system orientation definition.

Each pair is counter-rotating which produces a reaction torque on the body frame that is equally opposed at different motor speeds since the rigid body of the frame is not perfectly symmetrical. By having this cancellation in moment, the vehicle can sustain a consistent heading when hovering. In addition, a yawing motion is produced when the speed of a set of counter-rotating motor and fixed-pitch propeller combinations is either increased or decreased. This variation in speed produces a net counter torque on the rigid body that is non-zero. Furthermore, the altitude is adjusted with the variation of the thrust from all rotors while maintaining a zero-net moment. Nonetheless, the quadcopter platform is considered an under-actuated system since it can maneuver in six-degrees-of-freedom, but with only four control inputs. These control inputs are directly related to the overall amperage applied to each motor.

Table 3.1. Summary of variables used for quadcopter equations of motion.

Variable	Description
p	Roll rate
q	Pitch rate
r	Yaw rate
Φ	Euler roll angle
θ	Euler pitch angle
φ	Euler yaw angle
u	Velocity in the body frame along x^b
v	Velocity in the body frame along y^b
w	Velocity in the body frame along z^b
x	Inertial frame position along X^G
y	Inertial frame position along the Y^G
z	Inertial frame position along the Z^G

Table 3.2. Summary of quadcopter equations of motion.

Translational Kinematic Equations (3.1) $$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} uc_{\varphi}c_{\theta} + v\left(s_{\varphi}c_{\emptyset} + s_{\theta}c_{\varphi}s_{\emptyset}\right) + w(s_{\varphi}s_{\emptyset} - s_{\theta}c_{\varphi}c_{\emptyset}) \\ -us_{\varphi}c_{\theta} + v\left(c_{\varphi}c_{\emptyset} - s_{\theta}s_{\varphi}s_{\emptyset}\right) + w(c_{\varphi}s_{\emptyset} + s_{\theta}s_{\varphi}s_{\emptyset}) \\ us_{\theta} - vc_{\theta}s_{\emptyset} + wc_{\theta}c_{\emptyset} \end{bmatrix}$$
Force Equations (3.2) $$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} + \frac{1}{m}\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}^b$$
Rotational Kinematic Equations (3.3) $$\begin{bmatrix} \dot{\emptyset} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} p - qs_{\varphi} - \ rc_{\varphi}s_{\theta} \\ qc_{\varphi} + rs_{\varphi}s_{\theta} \\ rc_{\theta} \end{bmatrix}$$
Moment Equations (3.4) $$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \dfrac{L - qr(I_z - I_y)}{I_x} \\ \dfrac{M + pr(I_z - I_x)}{I_y} \\ \dfrac{N - pq(I_y - I_z)}{I_z} \end{bmatrix}$$

Table 3.1 and 3.2 summarize the variables and equations derived for the rigid body quadcopter dynamics, respectively. It is important to note that attitude determination utilizes the body-fixed reference frame while the translational position determination utilizes the global fixed reference frame.

Utilizing the CAD software, values for the mass and inertia of the vehicle are gathered and listed in Table 3.3. In addition, the forces corresponding to each axis in the inertial frame are given. It should be noted that there is a positive force parallel to the z-axis in the body frame of the vehicle due to the fixed motor position that only provides an upward force vector, f_z. The operating range of this value is between 0 and 21.37 Newtons as described in Section 3.3.

Table 3.3. Summary of CAD values for mass and inertia.

Symbol	Value	Unit
m	1.90	Kg
g	9.81	m/s^2
I_x	0.023241891	Kg/m^2
I_y	0.054302332	Kg/m^2
I_z	0.073220297	Kg/m^2
f_x	0	Kg-f
f_y	0	Kg-f
f_z	2.178	Kg-f

Uncertainties emerge with the estimated modeling values due to user errors in the modeling software. It is important to note that the material choices for each component in the assembly were in-family to the physical part. Nonetheless, the physically measured weight as discussed in Section 2.3 is used in the modeling software. Due to the resolution of the weight scale used for measuring all hardware parts, the weight used in the CAD model has ±1.05 grams of inaccuracy.

3.2 Brushless Direct-Current Motor and Propeller Model

To begin modeling of a brushless direct-current motor a typical schematic is used to illustrate the powertrain of the electric vehicle. Figure 3.2 shows a typical flow scheme of the electric powertrain using a three-phase motor.

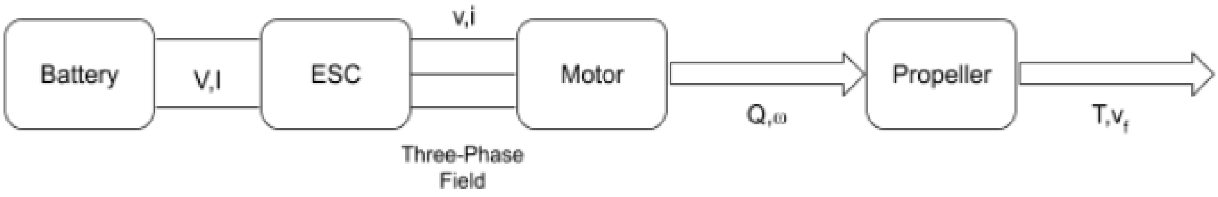

Figure 3.2. Electric powertrain scheme typically used for quadcopter UAV.

To simplify this model, both the electronic speed controller and battery are treated as constant in which the performance characteristics are directly related to the inherent technology rather than operational circumstances [22]. Therefore, the motor and propeller are of main significance in the modeling of this powertrain. Mono-phase equivalent circuits can be used to model brushless direct-current motors assuming a trapezoidal flux distribution [22]. In addition, it is assumed that the induced currents are neglected which results in a mathematical model expressed by Equations 3.2-3.4:

$$Q_m = K_t(I - I_0) \tag{3.2}$$
$$\omega = K_v(V - RI) \tag{3.3}$$
$$\eta = \frac{P_m}{P_{in}} = \frac{Q_{,}\omega}{VI} \tag{3.4}$$

where in Equation 3.2, K_t, is the torque gain expressed in torque per unit area, I_0, is the no-load current expressed in amperes, and I is the current drawn from the battery expressed in amperes. In Equation 3.3, K_v is the voltage gain measured in revolutions per volt, R is the resistance in the winding of the motor measured in Ohms, and V is the voltage supplied by the battery measured in Volts. In Equation 3.4, ɳ is the efficiency of the motor and is a ratio of the output mechanical power over the input mechanical power measured in Watts.

Datasheets of hardware specification typically provide characteristic parameters such as K_v, K_t, I_0, and R. If these parameters are not available through a manufacturer's datasheet where they can be inaccurate due to small scale UAV applications, they can be obtained through experimental means [23].

Nonetheless, utilizing the expressions of the motor and Ohm’s Law, the power that the motor consumes can be expressed in Equation 3.5. Equation 3.2 is rearranged to solve for the current and substituted into Equation 3.3, while Equation 3.3 is rearranged to solve for the voltage.

$$P = IV = \frac{(Q\ +\ K_tI_0)(K_tI_0\,R\ +\ QR\ +\ K_tK_v)}{K_t^2} \tag{3.5}$$

To support the simple mathematical model key simplifications and assumptions are made and listed below:

- The motor resistance is neglected, which makes the angular velocity proportional to the power consumed
- Since I_0 is small in a no-load circumstance, K_tI_0 << Q

Therefore, the simplified power equation for the motor is expressed as [23]:

$$P\ =\ \frac{K_v}{K_t}Q\omega \tag{3.6}$$

Utilizing conservation of energy, the motor draws energy over time which is equivalent to the force produced by the rotors multiplied by the distance that the air flowing through the propeller is moved. In other words, the power drawn multiplied by a change in time is equal to a force multiplied by a change in distance. This relationship is expressed in Equation 3.7 as:

$$P\,dt = F\,dx \qquad 3.7$$

where F is the thrust, T, and the ratio of the change in distance over the change in time is described as the induced velocity of the rotor. The equation can be rearranged as:

$$P = \frac{K_v}{K_t}Q\omega = \frac{K_v K_T}{K_t}T\omega = \frac{T^{\frac{3}{2}}}{\sqrt{2\rho A}} \qquad 3.8$$

and solving for thrust T:

$$T = \left(\frac{K_v K_T \sqrt{2\rho A}}{K_t}\omega\right)^2 = \alpha\omega^2 \qquad 3.9$$

It is obtained that the thrust and the square of angular velocity are proportional. K_T is determined by the blade parameters and configuration and is a constant proportional to the thrust. The constant α can be determined experimentally through load test data provided by the manufacturer.

3.3 Load Test Data

Load test data was provided by the manufacturer that utilized a 1380 Kv motor. Paired with the motor is a 7x4x3 inch propeller. This test configuration has a similar setup to the one presented in this paper. The only minor difference is that this test configuration uses a 50-Amp electronic speed controller whereas this paper uses a 60-Amp electronic speed controller. It is of minor significance since the electronic speed controller is treated as constant as mentioned in the previous section.

The test data provides values for voltage, current, RPM, thrust, and input power at various throttle settings. From the data, it is important to note that the maximum current draw of the motor is 43.7 Amps. A summary of the load test data is tabulated in Table 3.4. In addition, it is observed that the voltage from the battery decreases by 0.2 V over a maximum throttle input while the amperage draw from the battery significantly increases. The electronic speed controller was commanded in a similar method as presented in this paper which uses pulse-width modulation signals. Using this data an identification is made for the propeller thrust model. The tabulated load test data follows a quadratic relation and is illustrated in Figure 3.3. Therefore, this data adheres to the mathematical model expressed in Equation 3.9:

$$T = \alpha\omega^2 \qquad 3.9$$

where T is the thrust that the propeller and motor combination produce and ω is the speed of the motor measured in radians per second. It is then identified that the constant α is

$$a = 5.50E^{-6}$$

Table 3.4. Load test data of motor and propeller combinations.

	Throttle (%)	Voltage (V)	Current (A)	RPM	Thrust (N)	Input Power (W)
1380 Kv Motor with 7x4x3 inch Propeller	30	20.0	2.8	7681	3.09	56.00
	40	20.0	4.5	9128	4.97	90.0
	50	20.0	8.1	11195	6.94	162.0
	60	20.0	10.9	12590	8.49	218.0
	70	19.9	17.5	14345	11.30	348.25
	80	19.9	25.7	15991	14.72	511.43
	90	19.8	34.9	16550	19.05	691.02
	100	19.8	43.7	18820	21.37	865.26

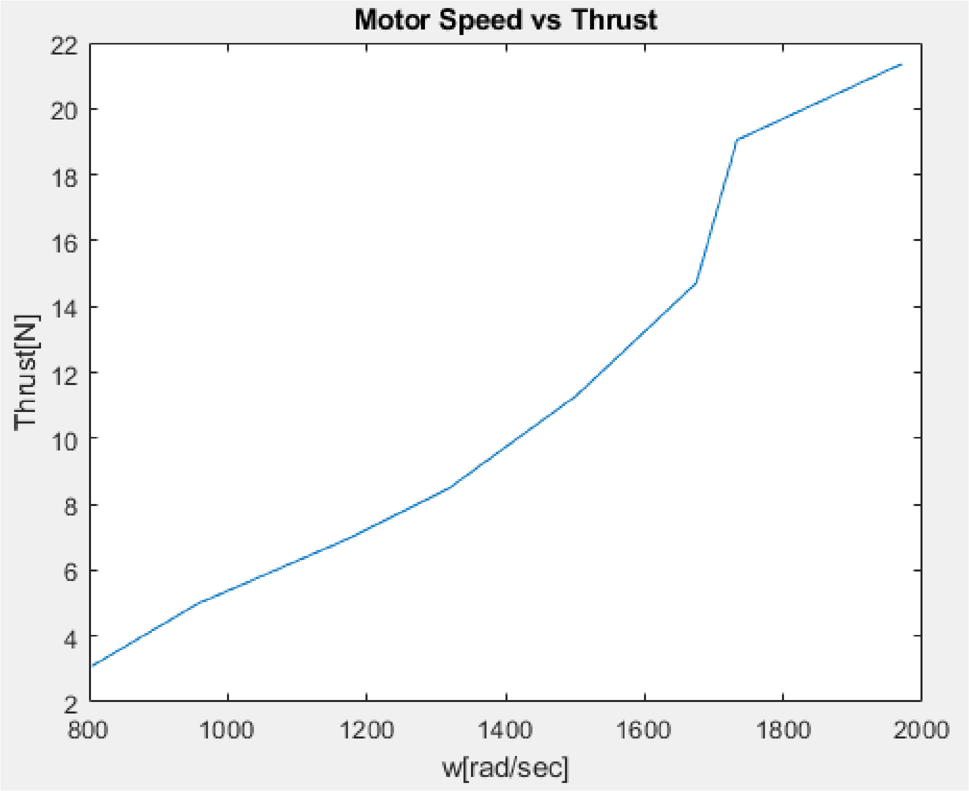

Figure 3.3. Static thrust curve for load test data.

3.4 Propeller Modeling

A quadcopter vehicle typically utilizes fixed-pitch propellers that are integrated with the propulsion system to provide rigidity throughout operation. Two key characteristics measure the performance of the propeller: torque and thrust. By taking equations 2.4.16 and 2.4.17, and rearranging them

$$Q = C_Q \rho \left(\frac{\omega}{2\pi}\right)^2 D_P^5 \qquad 3.10$$

$$T = C_T \rho \left(\frac{\omega}{2\pi}\right)^2 D_P^4 \qquad 3.11$$

where the torque and thrust coefficients are estimated using geometric parameters provided by the manufacturer. In Equations 3.10 and 3.11 the density changes with respect to the local height of the vehicle and the ambient temperature. Since the altitude and temperature have negligible changes during flight tasks, both the temperature and height can be considered constant for the purposes of this performance evaluation. In terms of propeller geometry such as diameter, pitch, number of blades, and weight, the coefficients are calculated using a geometric approach. Figure 2.6 can also be illustrated as shown in Figure 3.4 to highlight the absolute angle of attack, α_{ab}.

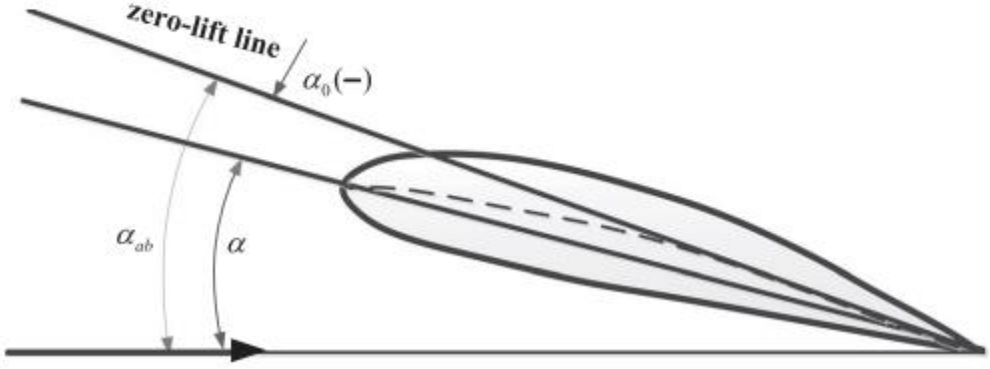

Figure 3.4. Illustration of absolute angle of attack and geometry.

Using Figure 2.6 and the geometry of the airfoil section, the pitch angle can also be expressed as

$$\theta = arctan\frac{H_p}{\pi D_P} \qquad 3.12$$

where D_P is the propeller diameter and H_P is the propeller pitch.

Using Figure 3.4, Equation 2.4.2 the effective angle of attack, is expressed as

$$\alpha = \varepsilon(\theta - \phi_0) \tag{3.13}$$

where epsilon is considered a correction factor that is produced with the presence of downwash. In addition, ϕ_0 is expressed as the propeller helix angle which is assumed to equal zero due to flight characteristics of a quadcopter vehicle [24]. Therefore, the absolute angle of attack as shown in Figure 3.4 is

$$\alpha_{ab} = \alpha - \alpha_0 \tag{3.14}$$

where α_0 is the zero-lift angle. This relationship is related to the lift and drag coefficients as proposed in [24] as

$$C_L = \frac{K_0 \alpha_{ab}}{1 + \frac{K_0}{\pi A}} \tag{3.15}$$

$$C_D = C_{fd} + \frac{1}{\pi A e} C_L^2 \tag{3.16}$$

where A is the aspect ratio ($A = D_P/c$) and K_0 is the slope of the lift curve. In addition, e is the Oswald efficiency factor and C_{fd} is considered the zero-lift drag coefficient. C_{fd} relates the relative angle of attack, Reynolds number, and the thickness of the blade [24]. At this point, the lift produced by the blade airfoil is expressed in terms of the lift coefficient, density, and the rotational speed. Rearranging Equation 2.4.5 and expressing it in terms of the blade area gives

$$L = \frac{1}{2} C_L \rho S U^2 \tag{3.17}$$

$$S = \frac{B_P}{2} \gamma D_p c \tag{3.18}$$

$$U \approx \frac{\pi \zeta D_P \omega}{60} \tag{3.19}$$

where S is the blade area, U is the average rotor linear velocity, and γ is the correction coefficient [24]. As described in section 2.4, the velocity of the blade is decomposed into two components where generally the latter is much lower than the former for quadcopter applications [24]. In addition, Equation 2.4.3 can be rewritten as Equation 3.19 which is in terms of propeller geometry. Blade element theory also suggests that the thrust provided by the rotating blades of the quadcopter does not equal to the produced lift [19]. The thrust is expressed as

$$T = L\frac{\cos(\kappa + \phi_0)}{\cos\,(\kappa - \delta)} \qquad 3.20$$

where κ is the corrected angle that emerges from the downwash effect and is expressed as

$$\kappa = \arctan\frac{C_D}{C_L}\,. \qquad 3.21$$

In addition, the torque can be expressed as shown in Equation 3.22 based on these results. Here the torque is a function of the density, coefficient of drag, area of the blade, and linear velocity of the rotor [24].

$$Q = \frac{1}{4}\rho B_P C_D U^2 S D_P \qquad 3.22$$

Therefore, the torque and thrust coefficients are expressed below according to Equations 3.12-3.22.

$$C_Q = \frac{1}{8A}\pi^2 C_D \zeta^2 \gamma B_P^2 \qquad 3.23$$

$$C_T = \frac{1}{4}\pi^3\gamma\zeta^2 B_P K_0 \frac{\varepsilon arctan\frac{H_P}{\pi D_P} - \alpha_0}{\pi A + K_0} \qquad 3.24$$

Parameters that are not reflected within the geometry of the airfoil such as γ, α_0, ε, ζ, K_0, and C_{fd} are difficult to calculate analytically. Therefore, Table 3.5 lists the values of these parameters as suggested by experimental results from relevant literature [24].

Table 3.5. List of suggested values for propeller parameters from relevant literature.

Symbol	Value
ζ	0.6
α_0	0
ε	0.85
γ	0.9
c	0.02
e	0.9
C_{fd}	0.02
A	8.89
K_0	6.28

To validate the modeling procedure described in this section, the experimental load test data that was provided by the manufacturer is used as a comparison to the analytical data obtained from these propeller models. Figure 3.5 illustrates the results obtained for the propeller thrust.

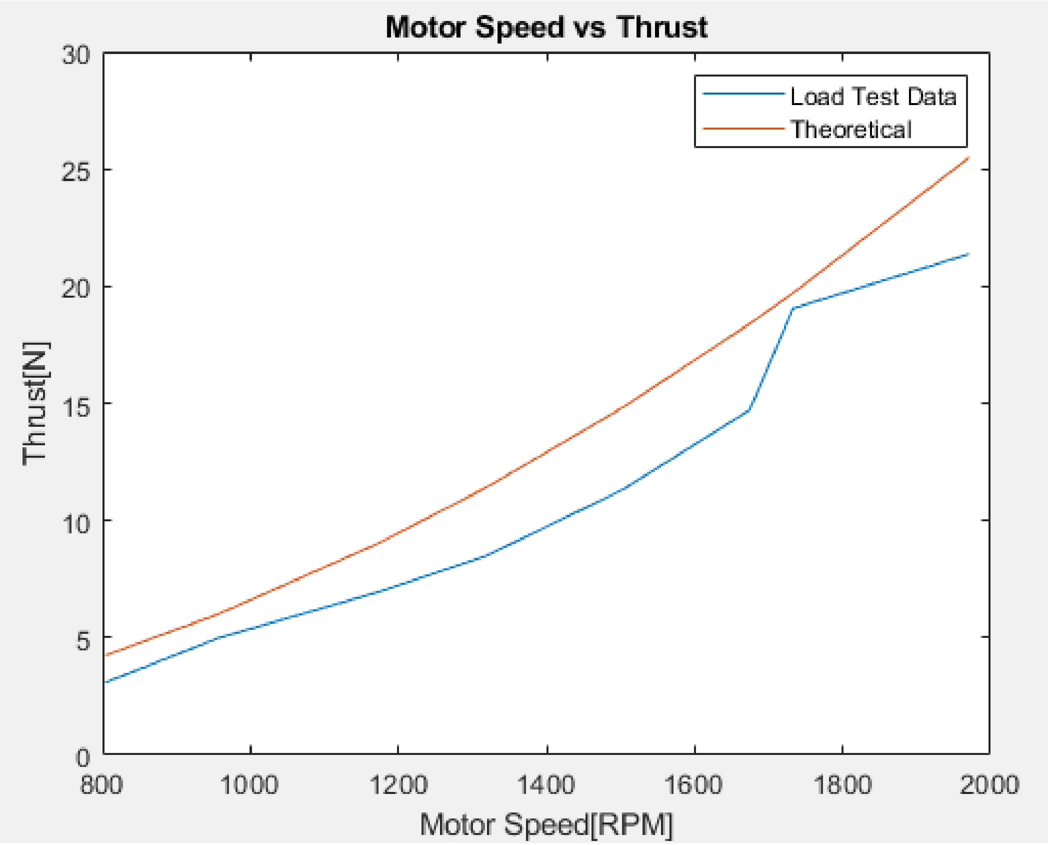

Figure 3.5. Model verification comparison for propeller thrust.

3.5 Motor Modeling

Electric direct-current motors are essential to the performance of a quadcopter vehicle. These motors allow the vehicle to have both translational and rotational motion creating a six DOF rigid body. For the purposes of this propulsion system architecture, brushless DC motors are used for providing the vehicle an orientation in three-dimensional space. Essentially, these motors can be described as synchronous three-phase permanent magnet motors [24]. Therefore, a permanent magnet direct-current motor is used to model the chosen component. The modeling of this component is expressed as an equivalent circuit design that is shown in Figure 3.6.

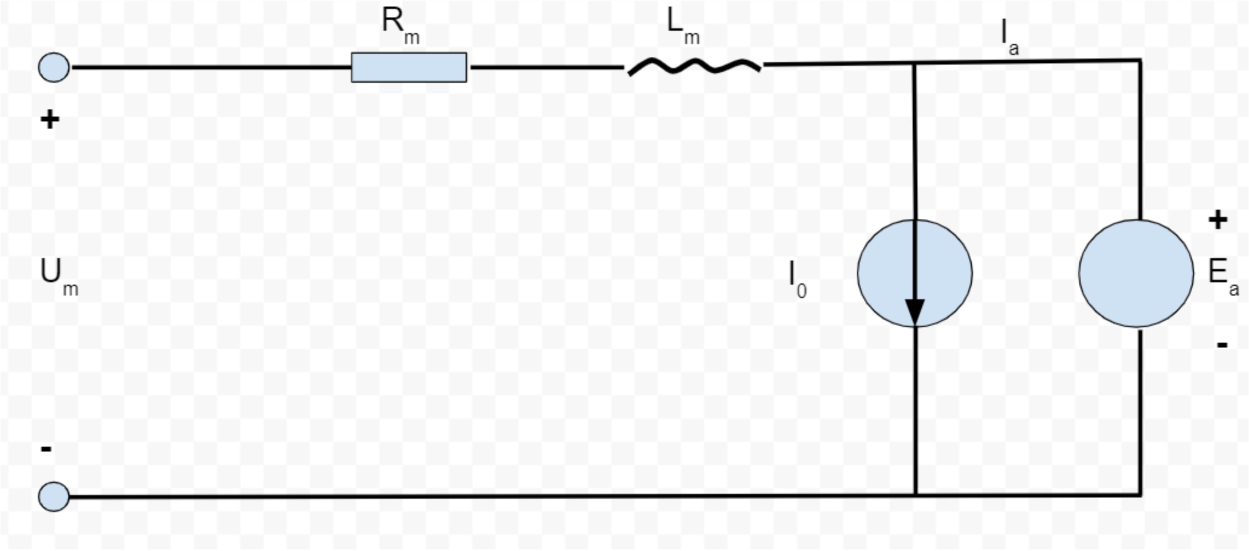

Figure 3.6. Brushless DC motor equivalent model.

It is important to note that in the equivalent motor model both the transient process produced by converted elements and the armature inductance denoted by L_m are negligible [24]. In the motor model I_0 is the no-load current necessary to overcome frictional forces due to the mechanical drivetrain, eddy current losses, and magnetic hysteresis due to magnetic properties. At a specific motor speed, the no-load current is estimated to be constant [24]. Furthermore, E_a is denoted as the back-electromotive force measured in Volts. U_m is the similar supply voltage entering the motor model measured in Volts. R_m is considered as the armature resistance measured in Ohms. I_m is expressed as the similar motor input current and is measured in Amps. Lastly, I_a provides a relationship for expressing the electromagnetic torque within the equivalent motor model. It is important to note that this motor equivalent circuit model focuses on retrieving the similar supply voltage and the similar motor input current. Both I_m and U_m are essential to producing appropriate thrust to maneuver the vehicle in the desired orientation. According to [25], the electromagnetic torque within the motor T_e is expressed as

$$T_e = K_T I_m \qquad 3.25$$

where the motors torque constant is denoted as K_T and measured in Newton-meters per amp. It is important to note that the output torque calculated in the previous section for the propeller model is equal to the output torque generated by the motor and is expressed as

$$M = T_e - T_0 = K_T(I_m - \widehat{I_0}) \qquad 3.26$$

where $\widehat{I_0}$ is the working state no-load current produced by the motor. After rearranging equation 3.26 the similar motor input current is calculated as

$$I_m = \frac{M}{K_t} + \widehat{I_0} \tag{3.27}$$

which is a function of the propeller torque, torque constant, and no-load current in working state. I_m must not exceed the manufactures specifications outlined in Section 3.3 of $I_{m,max}$ = 43.7 Amps. According to [25], the no-load current and the similar motor input current at initial motor state are assumed to be equal, therefore, $I_{m,0} = \widehat{I_0}$. Furthermore, the back electromotive force within the motor model is expressed as

$$E_a = K_E\omega \tag{3.28}$$

where the back-electromotive force constant is denoted by K_E and ω is defined as the motor speed. The similar motor voltage is the voltage supplied by the battery and according to [25] it is expressed as

$$U_m = K_E\omega + I_m R_m \tag{3.29}$$

where the back-electromotive force is added to the motor voltage. Since K_V is the voltage gain measured in revolutions per volt, $K_{V,0}$ is the nominal no-load voltage constant. This constant is defined as the ratio between the no-load input voltage and the no-load motor speed.

$$N_0 = K_{V,0}U_0 \tag{3.30}$$

It is important to note that the no-load conditions are similar to the initial operating conditions, so it is assumed in [25] that $U_{m,0} = U_m$. After combining and rearranging Equations 3.29 and 3.30 an expression for K_E is calculated

$$K_E = \frac{U_{m,0} - I_{m0}R_m}{K_{V0}U_{m0}}. \tag{3.31}$$

According to electric machine theory, an important relationship can be established between the back-electromotive force constant and the motor torque constant which is expressed as

$$K_T = 9.55K_E. \tag{3.32}$$

Therefore, K_T becomes

$$K_T = 9.55\frac{U_{m,0} - I_{m0}R_m}{K_{V0}U_{m0}}. \tag{3.33}$$

With these relationships that were derived from [25], expressions for the similar motor input current and similar supply voltage can be calculated in terms of the armature resistance, no-load current, no-load voltage, no-load voltage constant, and the propeller torque.

$$U_m = R_m\left(\frac{MK_{V0}U_{m0}}{9.55(U_{m0} - I_{m0}R_m)} + I_{m0}\right) + \frac{U_{m,0} - I_{m0}R_m}{K_{V0}U_{m0}}\omega \qquad 3.34$$

$$I_m = \frac{MK_{V0}U_{m0}}{9.55(U_{m0} - I_{m0}R_m)} + I_{m0}\ . \qquad 3.35$$

3.6 Electronic Speed Controller Modeling

An ESC is used to communicate the amount of voltage and current needed to supply the motor with adequate power. Essentially, the ESC's input from the battery which is DC voltage is converted to a three-phase signal. This alternating signal and rotor rotation are synchronized. In addition, the signal is directly utilized by the armature windings. It is important to note that the ESC regulates the speed of the motor between a specific range that is dependent on the battery voltage and the load capacity. Similar to the approach taken in Section 3.5, Figure 3.7 illustrates and equivalent circuit for the ESC used in this application.

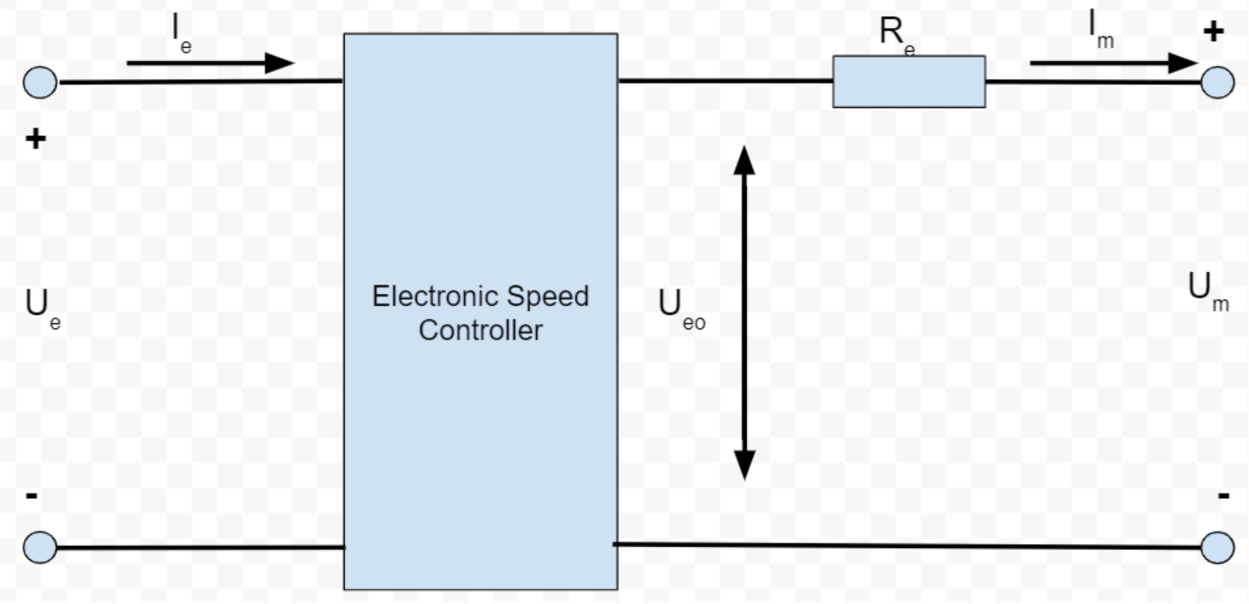

Figure 3.7. ESC equivalent circuit model schematic for brushless dc motor.

In the figure above, U_e is the supplied voltage and I_e is the supplied current provided by the battery pack. The purpose of this modeling procedure is to obtain U_e and I_e as functions of the motor input voltage and input current as defined in Section 3.5. Therefore, as illustrated in Figure 3.7, U_{eo} is defined as the equivalent dc voltage expressed as

$$U_{eo} = U_m + I_m R_e. \tag{3.36}$$

Equation 3.37 is used to describe the duty cycle, D_c, of the system. Essentially, the duty cycle value is measured in percentage where this percentage expresses the specific percentage of time that a desired digital signal is active on an interval of time.

$$D_c = \frac{U_{eo}}{U_e} \cong \frac{U_{eo}}{U_b} \tag{3.37}$$

where the range of the duty cycle is [0,1]. According to [24], the duty cycle described by this electrical system productively creates an integrated dc buck converter because the duty cycle effectively modulates the average motor voltage across a commutating sequence. Therefore, the input current to the ESC is expressed as

$$I_e = D_c I_m. \tag{3.38}$$

It is important to note that the input current is limited by the maximum input current, $I_{e,max}$. Also, the ESC voltage, U_e, can be expressed in terms of the battery voltage, current, and resistance through Ohms Law

$$U_e = U_b - I_b R_b. \tag{3.39}$$

For multi-copter applications, the number of motors is directly equivalent to the number of ESCs unless an ESC combo is used such as a four-in-one ESC. However, for this application there are four motors, therefore, there are four ESCs. Given the number of rotors, N_r, the battery current, I_b, can be expressed as

$$I_b = N_r I_e + I_{control} \tag{3.40}$$

where the current applied to the dedicated flight controller is defined as $I_{control}$. Finally, expressions for U_e, I_e, and D_c are summarized below in Equations 3.37-3.39.

$$D_c = \frac{U_m + I_m R_e}{U_b} \quad 3.37$$

$$I_e = D_c I_m \quad 3.38$$

$$U_e = U_b - I_b R_b \quad 3.39$$

3.7 Battery Modeling

For hovering conditions, the battery model described in this section focuses on estimating the endurance, specifically the time of endurance measured in minutes. Endurance time is denoted by T_{end} and is a function of the battery current and capacity. It is important to note that assumptions were made to simplify the battery discharge process. One key assumption involves keeping the battery voltage constant, so for this specific application the 5-S battery has a constant maximum charge voltage of 18.5 Volts. In addition to simplifying the discharge process, it is assumed that the battery capacity decreases linearly for steady hovering flight. In high-maneuver flight, the battery capacity is expected to change non-linear due to the excess amperage draw from various motors. Nonetheless, this battery model focuses on expressing the time of endurance in terms of manufacturer specifications such as the battery capacity and battery current. Therefore, the time of endurance is

$$T_{end} = \frac{C_b - C_{min}}{I_b} \frac{60}{1000} \quad 3.41$$

where the minimum battery capacity is denoted by C_{min}. The minimum battery capacity is provided by the manufacturer and is determined based on the safety margin of the physical battery itself. In this specific application, the minimum battery capacity is determined to be 750 mAh. However, a constraint is placed on the battery current as expressed in Equation 3.42.

$$I_b \leq K_b C_b \quad 3.42$$

4 Results

The flight simulator used for this system is based on a game engine known as Unity3D which incorporates the Nvidia PhysX engine to model rigid body dynamics. In addition, inputs to the simulation drive the flight controller of the quadcopter which is based on a linearized model in Simulink about the quadcopters hovering condition. Furthermore, the resultant torques and forces calculated from the flight controller are applied to the respective motor which drive the Unity3D simulation. It is important to note that complicated physical behavior such as ground effect, propeller tip losses, and motor windup are not incorporated into either the Nvidia PhysX engine or the simulation parameters. However, drag forces are included as part of the physics simulation provided by Unity3D whereas drag is not included as part of the 6-DOF quadcopter model. However, the fidelity of the rigid body dynamics from the PhysX engine is steady at 1000 Hz.

4.1 Flight Controller

The flight controller utilizes best fit curves of the load test data presented in Table 3.4 to drive the simulation in Unity3D. Curve fit lines for the load test data are obtained through statistical models such as a regression line modeling expressed as

$$m = \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2} \tag{4.1}$$

$$b = \frac{(\sum y) - m(\sum x)}{n}. \tag{4.2}$$

In Equations 4.1 and 4.2, n is denoted as the data sample size, m is the slope of the regression line, and b is defined as the y-intercept of the regression model. It is important to note that a higher sample size will result in greater accuracy within the regression model. The current sample size used that is provided by the manufacturer is 8. The regression lines for the static thrust, current, and torque of a motor are plotted below and are illustrated in Figures 4.1, 4.2, and 4.3, respectively.

Static thrust regression line:

$$y = 0.0161x - 11.0976$$

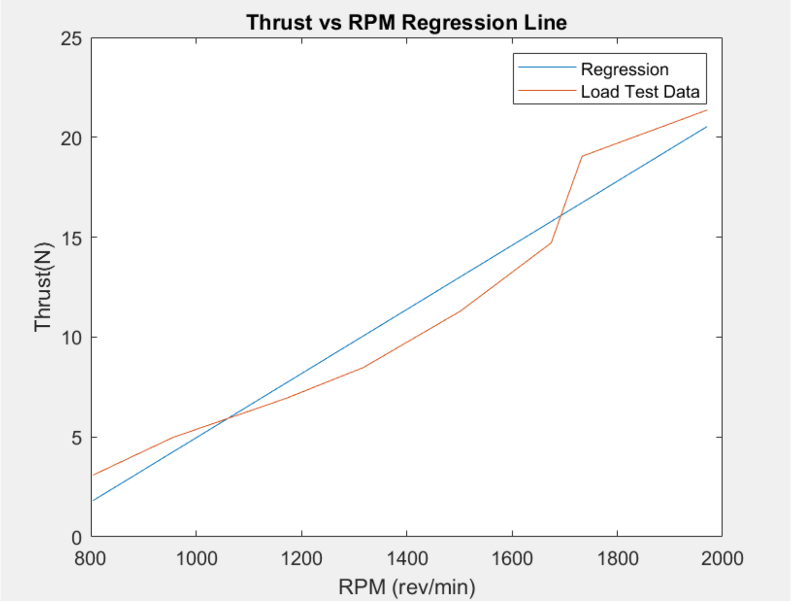

Figure 4.1. Linear regression line for thrust.

Current regression line:

$$y = 0.0356x - 30.9751$$

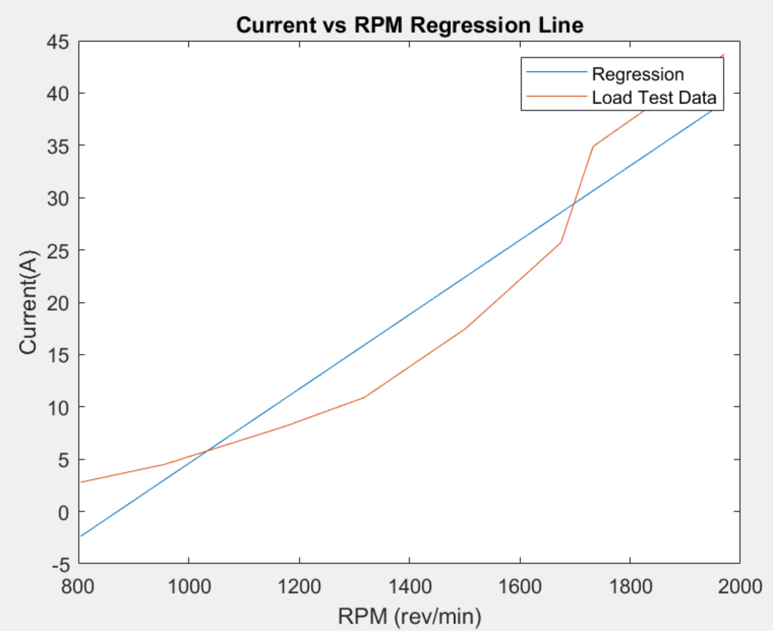

Figure 4.2. Linear regression line for current.

Torque regression line:

$$y = 0.00033417x - 0.2347$$

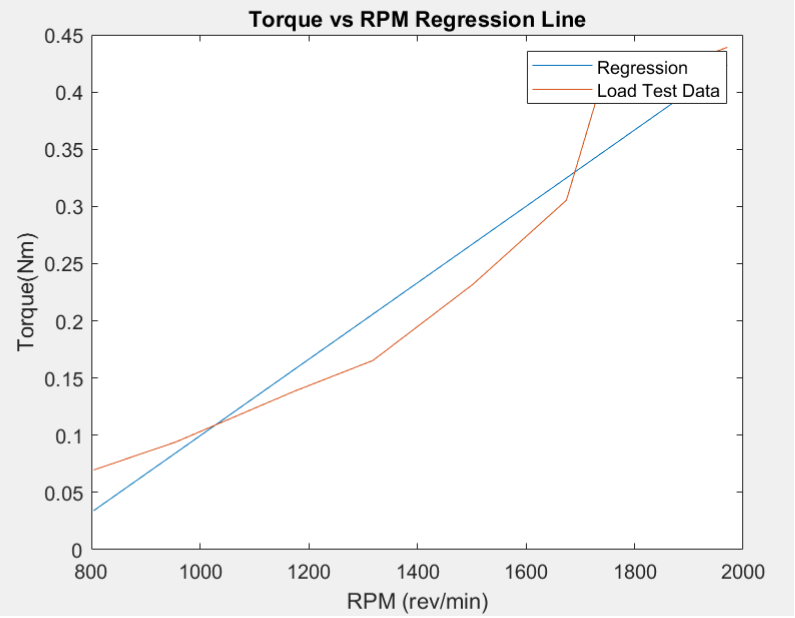

Figure 4.3. Linear regression line for torque.

The regression models shown above provide uncertainties due to linear trends. A different approach which lowers the uncertainty is achieved by introducing a pure quadratic, least-squares curve fit regression model for the load test data of thrust, torque, and current. The aim of these models is to determine the thrust, torque, and current coefficients while utilizing a least squares fit. The model is based on a numerical method to solving a general regression model that incorporates given measurement vectors and predicted equations of the form y = Ax. The design variable x € R^{n+1} minimizes e which is the sum of squared errors:

$$e = \left(y - y_{pred}\right)^T (y - y_{pred}) \tag{4.3}$$

$$e = (y - Ax)^T (y - Ax) \tag{4.4}$$

$$e = (y^T - x^T A^T)(y - Ax) \tag{4.5}$$

$$e = y^T y - y^T Ax - x^T A^T y + x^T A^T Ax \tag{4.6}$$

By taking the derivative of the error equation with respect to the x variable and setting the equation equal zero while using the denominator convention, Equation 4.6 becomes:

$$\frac{de}{dx} = -A^T y - A^T y + 2A^T Ax = 0. \quad 4.7$$

$$A^T Ax = A^T y \quad 4.8$$

$$x = (A^T A)^{-1} A^T y \quad 4.9$$

Equation 4.6 is known as the normal equation which is a convergence of the general regression model of a pure quadratic. The numerical solutions for the thrust, torque, and current coefficients are listed below in Table 4.1.

Table 4.1. List of coefficients for thrust, torque, and current for least squares fit curve.

Coefficient	Value
Kf	1.935969224275217e-07
Ktor	8.392986377371305e-09
Kiw	8.906404760621757e-10

The resulting regression lines are plotted in Figures 4.4 - 4.6 for static thrust, torque, and current, respectively. These regression models are plotted against the load test data and compared to the linear models.

Static thrust least squares, pure quadratic:

$$y = 1.93e^{-07} * RPM^{2.45}$$

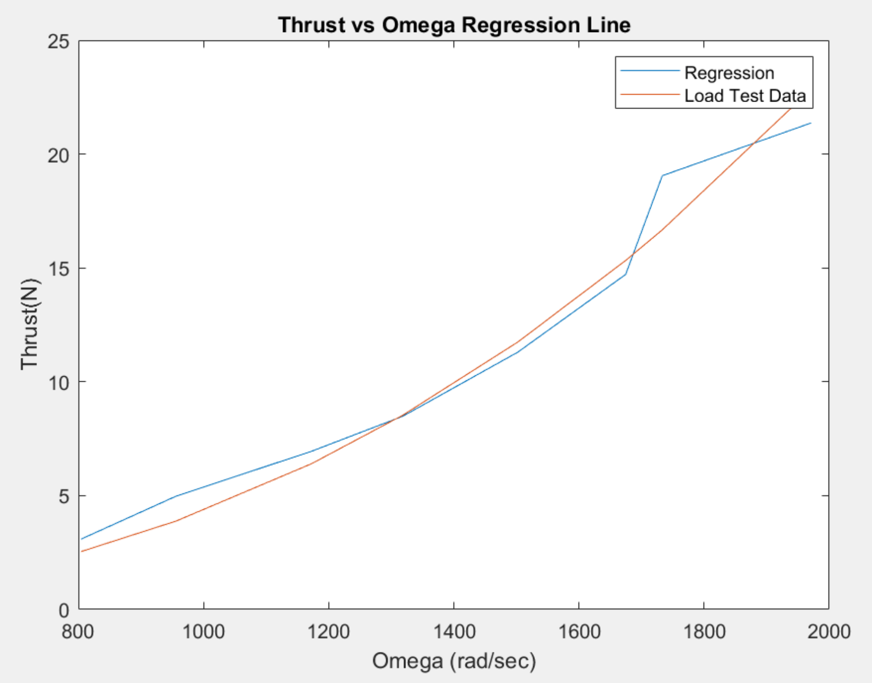

Figure 4.4. Least squares regression line for static thrust.

Torque least squares, pure quadratic:

$$y = 8.39e^{-09} * RPM^{2.35}$$

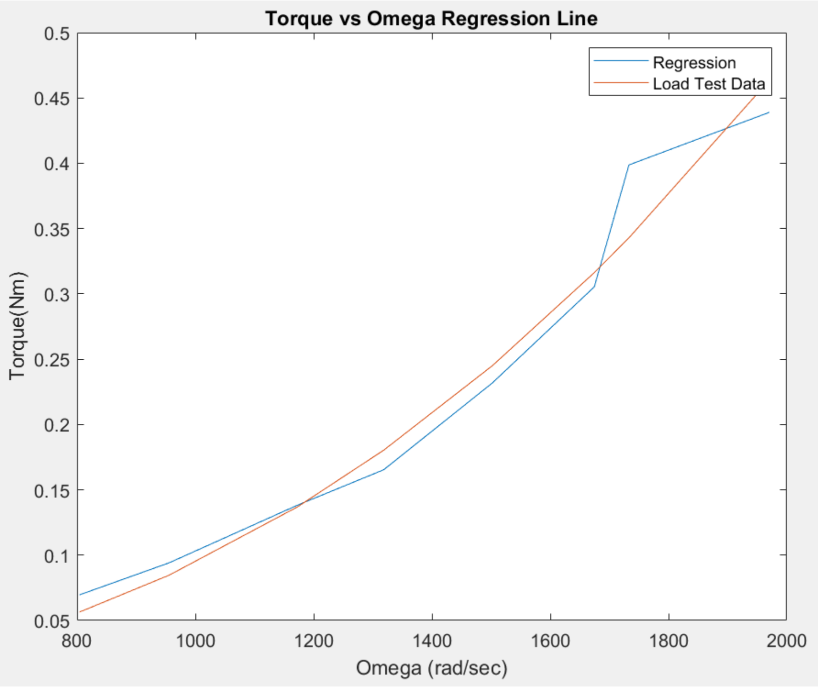

Figure 4.5. Least squares regression line for torque.

Current least squares, pure quadratic:

$$y = 8.91e^{-10} * RPM^{3.25}$$

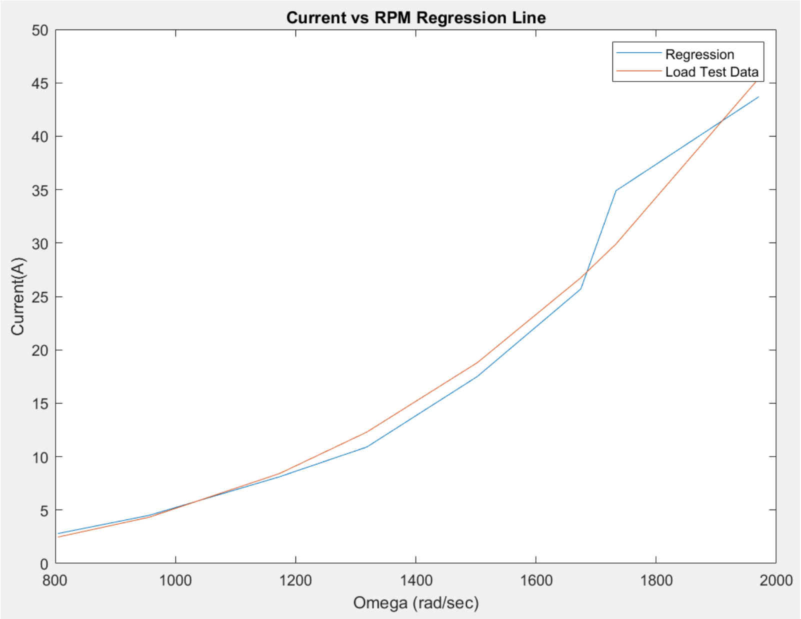

Figure 4.6. Least squares regression line for current.

The following three Figures show all regression models for static thrust, torque, and current used to estimate the load test data provided by the manufactures. The figures include the linear regression lines, polynomial curve fit lines, least square curve fit lines, theoretical models, and the load test data for comparison. The theoretical model curves are derived in Section 3.4 and 3.5.

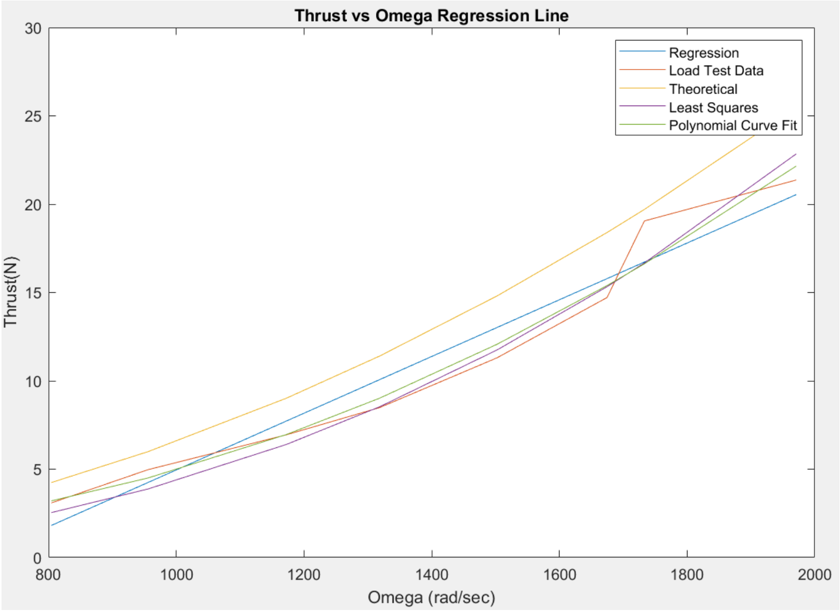

Figure 4.7. Regression lines for static thrust compared to theoretical and load test data.

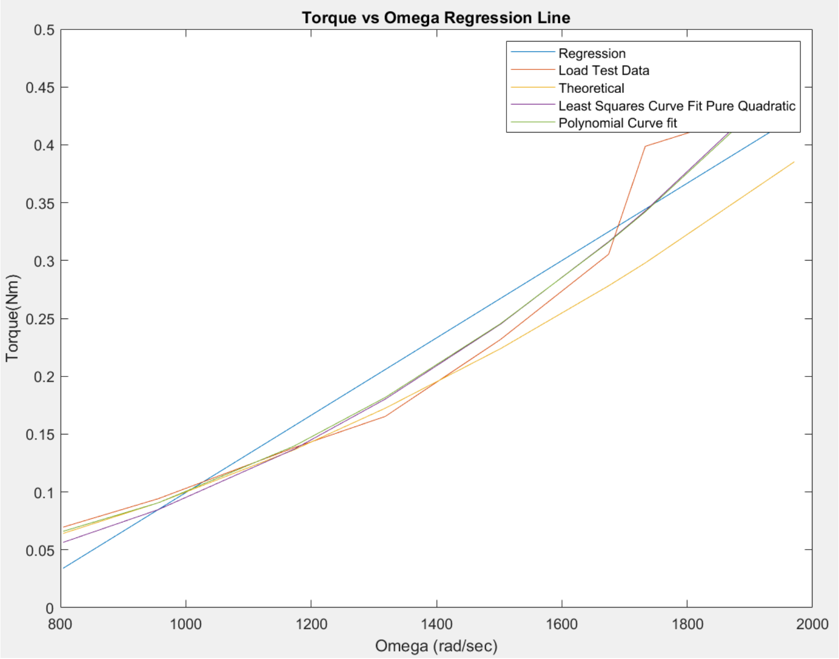

Figure 4.8. Regression lines for torque compared to theoretical and load test data.

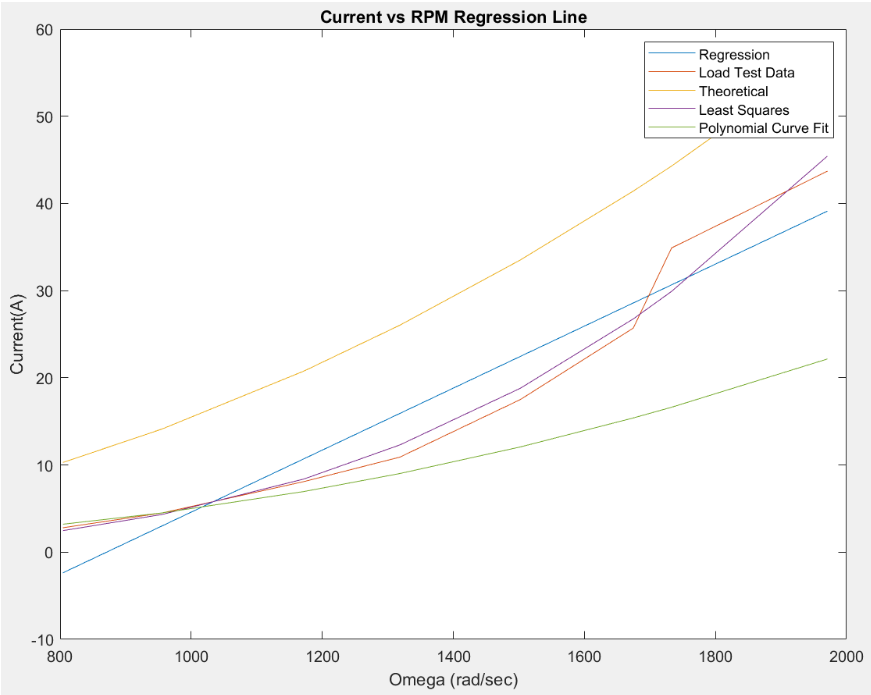

Figure 4.9. Regression lines for current compared to theoretical and load test data.

Upon comparing Figures 4.7, 4.8, and 4.9 to the load test data, it is evident that the least squares regression models are favorable rather than the theoretical calculations. For static thrust, the values are higher due to the lack of mechanical loss modeling for propeller and motor design. Theoretical models for current also lacked modeling for electrical losses throughout the propulsion system design. These unincorporated mechanical and current loss models provide higher values when compared to the load test data. Lastly, although the theoretical model lacks mechanical torque losses it compares closely to the least squares regression model for higher accuracy. An analysis was conducted to show the root mean square errors of all regression models are tabulated in Table 4.2, 4.3, and 4.4.

Table 4.2. Thrust Root Mean Square Analysis (RSME) of predicted models.

Regression Model	**Equation Form**	**RSME**
Linear	$y = mx + b$	1.3894
Polynomial	$y(x) = y_1 x^n + y_2 x^{n-1} + \ldots + y_n x + y_{n+1}.$	1.0052
Least Squares	$y = ax^c$	1.1296
Theoretical	$T = C_T \rho \left(\frac{\omega}{2\pi}\right)^2 D_P^4$	2.6999

Table 4.3. Torque Root Mean Square Analysis (RSME) of predicted models.

Regression Model	**Equation Form**	**RSME**
Linear	$y = mx + b$	0.0319
Polynomial	$y(x) = y_1x^n + y_2x^{n-1} + \ldots + y_nx + y_{n+1}.$	0.0231
Least Squares	$y = ax^c$	0.0237
Theoretical	$Q = C_Q\rho\left(\frac{\omega}{2\pi}\right)^2 D_P^5$	0.0417

Table 4.4. Current Root Mean Square Analysis (RSME) of predicted models.

Regression Model	**Equation Form**	**RSME**
Linear	$y = mx + b$	4.0723
Polynomial	$y(x) = y_1x^n + y_2x^{n-1} + \ldots + y_nx + y_{n+1}.$	10.8334
Least Squares	$y = ax^c$	2.251
Theoretical	$I_m = \frac{MK_{V0}U_{m0}}{9.55(U_{m0} - I_{m0}R_m)} + I_{m0}$	12.7821

4.2 Position Controller

The flight controller for the vehicle includes various sub controllers to incorporate translational and rotational motion. The utilization of cascade P, PD, and PID controllers establish the sub controller design used to linearize the model. It is important to note that the controller is tuned based off the six degree-of-freedom model established for the quadcopter dynamics as shown in Table 3.2. The simulated and theoretical sequence of rotation is yaw-pitch-role or 3-2-1 order. This is one conventional sequence for unmanned aerial vehicles. In addition, reference frames are used to establish these equations. Essentially, two pairs of motors, 1-4 and 2-3 spin in opposite directions: one pair spin clockwise while the other pair spin counterclockwise, respectively. This allows for a reversed pitch on the corresponding propellers to always create thrust in the body frame of the quadcopter in the negative z^b direction for all four propellers. Although this configuration makes the reaction moments on the body frame in the same direction, the signs are opposite to account for the direction of the angular velocity vector of the propellers. For the quadcopter to hover, or hold its position, the overall thrust generated by the motors needs to compensate for earth gravity. In addition, the sum of the torques produced by all four motors must converge to zero to adequately hold its position. To ascend and descend in altitude, the same balanced thrust generation must be equally increased or decreased to vary altitude. It should be noted that while the provided thrust from the propeller increases so does the overall body torques produced by the propellers, but the sum remains zero. If this sum is not equal to zero then the quadcopter is theoretically yawing, or changing its heading, depending on which motor speeds vary. To turn left and right, motor pairs 1-4 and 2-3 need to increase rotational speed while motors 2-3 and 1-4 decrease in angular speed, respectively. This creates an imbalance in torque on the vehicles body frame that effectively rotates the quadcopter along the z^b body frame. The equations used to describe the motion of quadcopter are presented below.

Table 4.5. Summary of quadcopter dynamics.

Translational Kinematic Equations (3.1) $\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} uc_{\varphi}c_{\theta} + v\left(s_{\varphi}c_{\emptyset} + s_{\theta}c_{\varphi}s_{\emptyset}\right) + w(s_{\varphi}s_{\emptyset} - s_{\theta}c_{\varphi}c_{\emptyset}) \\ -us_{\varphi}c_{\theta} + v\left(c_{\varphi}c_{\emptyset} - s_{\theta}s_{\varphi}s_{\emptyset}\right) + w(c_{\varphi}s_{\emptyset} + s_{\theta}s_{\varphi}s_{\emptyset}) \\ us_{\theta} - vc_{\theta}s_{\emptyset} + wc_{\theta}c_{\emptyset} \end{bmatrix}$
Force Equations (3.2) $\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} + \frac{1}{m}\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}^b$
Rotational Kinematic Equations (3.3) $\begin{bmatrix} \dot{\emptyset} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} p - qs_{\varphi} - rc_{\varphi}s_{\theta} \\ qc_{\varphi} + rs_{\varphi}s_{\theta} \\ rc_{\theta} \end{bmatrix}$
Moment Equations (3.4) $\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{L - qr(I_z - I_y)}{I_x} \\ \frac{M + pr(I_z - I_x)}{I_y} \\ \frac{N - pq(I_y - I_z)}{I_z} \end{bmatrix}$

It is important to note that the variables p, q, and r are given and are measured from onboard rate gyros, or IMU. The BNO055 absolute orientation sensor from Adafruit that was used for this project can output three-axis orientation data based on a 360-degree sphere, three-axis of rotational speed in radians per second, and three-axis linear acceleration in meters per second squared. This sensor data is set at a sample rate of 100 Hz while other sensor data such as magnetic field strength vector and temperature sensor are 20 Hz and 1 Hz, respectively. Eulerian angles ψ, θ, and φ are calculated numerically through Simulink using the above equations. The variables I_x, I_y, and I_z correspond to the principal moment of inertia and are gathered from the inertia tensor used in the angular momentum equation. It should be noted that the products of inertia in the inertia tensor are not convergent to zero for this quadcopter configuration due to unequal moment arm lengths for each motor; the inertia tensor is listed below in Table 4.6.

Table 4.6. Inertia tensor used in 6-DOF system gathered from Creo Parametric 7.0.

Inertia Tensor					
I_{xx}	**I_{xy}**	**I_{xz}**	2.907E4	-1.114E4	-6.292E3
Iyx	**I_{yy}**	**I_{yz}**	-1.114E4	4.939E4	-2.285E3
I_{zx}	**I_{zy}**	**I_{zz}**	-6.292E3	-2.285E3	7.230E4

To increase the stability of the flight controller, filtered derivative, output saturation, and anti-windup clamping methods are used. In general, the stability of linear time-invariant dynamic systems may have one or more states of equilibrium. An equilibrium state may be unstable if for small perturbations the system does not return back to equilibrium an diverges away. However, nonlinear systems may also have one or more equilibrium states, but each of them requires a set of local stability properties. It is important to note that the stability of the equilibrium state determines the stability of the overall system in a time-invariant system which has only one. The block diagram for the cascade controller design is shown in Figure 4.10.

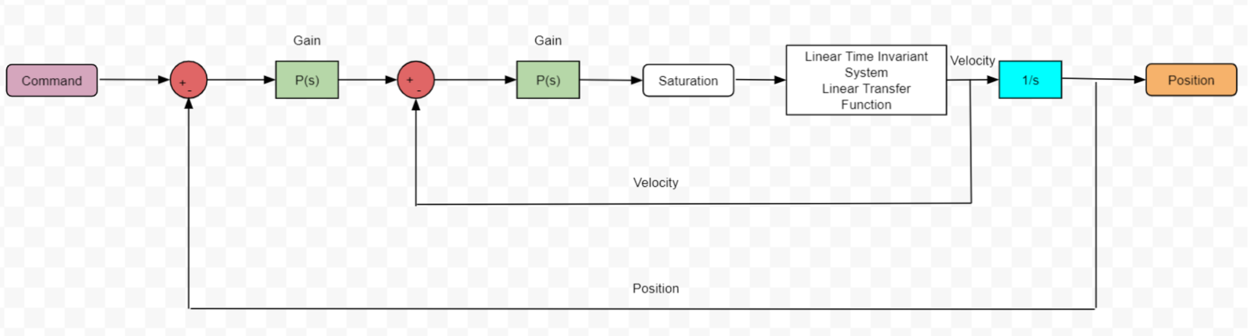

Figure 4.10. Cascade controller block diagram for quadcopter mode.

It is important to note that the developed sub controller model is linearized about the quadcopters hovering conditions. The sub controller has an outer loop of position and an inner loop of velocity. The control task for this specific model is to select the proper gain values for each degree of freedom for each of the two loops. The gain value blocks are P(s) and are denoted by the green colored Simulink blocks. The gain values are tuned using the automated PID tuner provided by Simulink. This design technique is a major drawback due to having a controller linearization about only one operating condition. This is important since for any other flight conditions outside the linearized model will translate into inaccurate performance values. However, for low-speed flights, this technique is useful and acceptable since the quadcopter will be operating close to this condition. The gain values used are listed below in Table 4.7 and 4.8.

Figures 4.11-4.13 show the desired yaw, pitch, and roll plots that were achieved with the tuned gain values shown in Table 4.6, respectively.

Table 4.7. Controller tuned gained values for X, Y, and Z position.

Symbol	Description	X	Y	Z
P_{outer}	Outer loop proportional gain constant	0.2168	0.5225	3.0223
I_{outer}	Outer loop integral gain constant	0.0285	0.0023	1.1817
D_{outer}	Outer loop derivative gain constant	0	-0.2812	0.0
N_{outer}	Outer loop filtered coefficient constant	0	0.1857	0.0
P_{inner}	Inner loop proportional gain constant	-0.0537	0.0643	12.228
I_{inner}	Inner loop integral gain constant	-0.3423	0	0.0
D_{inner}	Inner loop derivative gain constant	-0.2111	0	0.0
N_{inner}	Inner loop filtered coefficient constant	26.84	0	0.0
S_{high}	Outer loop saturation upper limit	2.0	1.0	2.0
S_{low}	Outer loop saturation lower limit	-2.0	-1.0	-2.0
SD_{high}	Inner loop saturation upper limit	0.5950	0.5950	60
SD_{low}	Inner loop saturation lower limit	-0.5950	-0.5950	0.0

Table 4.8. Controller tuned gain values for Pitch, Yaw, and Roll.

Symbol	Description	Pitch	Yaw	Roll
P_{outer}	Outer loop proportional gain constant	2.7498	7.2731	3.7564
I_{outer}	Outer loop integral gain constant	0.0601	0.4407	0.1134
D_{outer}	Outer loop derivative gain constant	-0.4106	-0.1018	-0.3834
N_{outer}	Outer loop filtered coefficient constant	3.4233	6.6355	4.6655
P_{inner}	Inner loop proportional gain constant	1.1404	9.4157	0.6639
I_{inner}	Inner loop integral gain constant	6.1980	0.0	4.4882
D_{inner}	Inner loop derivative gain constant	0.0335	0.0	0.0126
N_{inner}	Inner loop filtered coefficient constant	53.895	0.0	49.268
S_{high}	Outer loop saturation upper limit	0.5950	0.1700	0.5950
S_{low}	Outer loop saturation lower limit	-0.5950	-0.1700	-0.5950
SD_{high}	Inner loop saturation upper limit	5	5	5
SD_{low}	Inner loop saturation lower limit	-5	-5	-5

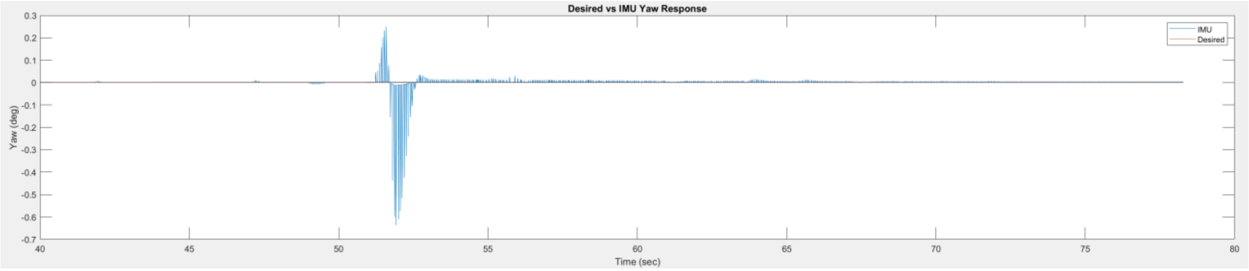

Figure 4.11. Desired yaw response that was achieved with the tuned gain values.

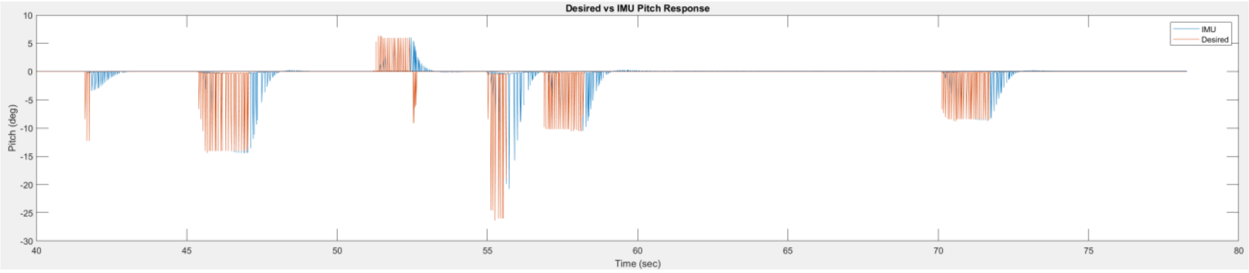

Figure 4.12. Desired pitch response that was achieved with the tuned gain values.

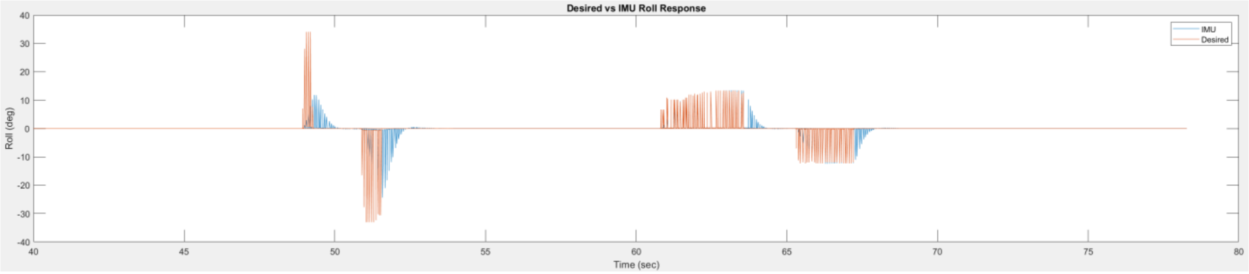

Figure 4.13. Desired roll response that was achieved with the tuned gain values.

4.3 Discussion

The overall performance of the hardware is a limiting factor to the simulation of the flight controller as mentioned in Section 2.2. Hardware limitations for the motor drive system such as the electronic speed controller hinder the output performance of both the single board computers such as the frequency and response times. The ESCs are at a fixed frequency of 50 Hz where reconfiguring is not possible without effectively replacing the component entirely. The highest frequency that the flight controller can operate at for increased accuracy with the current configurations is 100 Hz which is double that what the ESCs are operating at.

4.4 Future Research

The modeled flight controller and experimental results are based on the first iteration of the electric quadcopter vehicle. This vehicle design only incorporates the hardware discussed in section 2.2 whereas future considerations involve higher performing single board computers and a more sophisticated sensor suite. The Nvidia Jetson Nano is to be replaced by an Nvidia Jetson AGX Xavier which has higher processing power and is better optimized for machine learning. When compared, the Jetson AGX Xavier has an enhanced 512-core Volta graphical processing unit (GPU) that includes tensor cores and an 8-core ARM 64-bit CPU for increased processing. In addition, the storage capacity is significantly increased from 8 GB to 32 GB for onboard storage allocation. However, the size of the AGX Xavier is significantly larger at 105 x 105 x 65 millimeters when compared to the Jetson Nano. Nonetheless, the Nvidia Jetson AGX Xavier is essentially an AI computer for autonomous applications where it provides enhanced GPU workstation performance in an embedded hardware module for under 30 Watts of power. In addition to the 30-Watt module, a 15-Watt light detection and ranging (Lidar) system will be incorporated. Essentially, the lidar sensor creates a virtual map of the environment that consists of both static and dynamic objects for a translating vehicle to navigate throughout a variety of environmental and lighting conditions safely and accurately. However, due to the massive amounts of power consumption between these two new modules in future iterations, the addition of a new battery for independent use may be needed. An increased power consumption results in a decreased flight time and power to weight ratio. Nonetheless, a similar analysis that compares theoretical and experimental modeling results through statistical regression will be conducted with the new components.

5 References

[1] D. F. | D., “NVIDIA Jetson AGX Xavier Delivers 32 TeraOps for New Era of AI in Robotics,” NVIDIA Developer Blog, retrieved 5 September 2020.
https://developer.nvidia.com/blog/nvidia-jetson-agx-xavier-32-teraops-ai-robotics/

[2] Foehn, P., Brescianini, D., Kaufmann, E., Cieslewski, T., Gehrig, M., Muglikar, M., and Scaramuzza, D., “AlphaPilot: Autonomous Drone Racing,” *arXiv.org* Database, retrieved 12 September 2020.
https://arxiv.org/abs/2005.12813.

[3] “AlphaPilot – Lockheed Martin AI Drone Racing Innovation Challenge,” HeroX, retrieved 18 September 2020.
https://www.herox.com/alphapilot/updates

[4] Shi, D., Dai, X., Zhang, X., and Quan, Q., “A Practical Performance Evaluation Method for Electric Multicopters,” IEEE/ASME Transactions on Mechatronics, Vol 22, NO.3, June 2017, ResearchGate, retrieved on 13 September 2020.
https://www.researchgate.net/publication/314200017_A_Practical_Performance_Evaluation_Method_for_Electric_Multicopters

[5] Szafranski, G., Czyba, R., and Błachuta, M., “Modeling and Identification of Electric Propulsion System for Multirotor Unmanned Aerial Vehicle Design,” IEEExplore, retrieved 23 September 2020.
https://ieeexplore.ieee.org/document/6842287

[6] Papadimitriou, D., Rosu, V., Naidu, V., Cruz, D., and Skarakis, J., “Reliability-Based Aerodynamic Shape Optimization of a Quadcopter,” ResearchGate, retrieved 14 September 2020.
https://www.researchgate.net/publication/322312767_Reliability_Based_Aerodynamic_Shape_Optimization_of_a_Quadcopter

[7] Wolfram, D., Vogel, F., and Stauder, D., “Condition Monitoring for Flight Performance Estimation of Small Multirotor Unmanned Aerial Vehicles,” IEEExplore, retrieved 10 September 2020.
https://ieeexplore.ieee.org/document/8396471

[8] Harrington, A. M., “Optimal propulsion system design for a micro quad rotor”, Harrington Library, retrieved 15 September 2020.
https://drum.lib.umd.edu/bitstream/handle/1903/12029/Harrington_umd_0117N_12609.pdf?sequence=1&isAllowed=y

[9] Leutenegger S. et al., “Flying Robots,” Springer Handbook of Robotics, retrieved 21 September 2020.
https://doi-org.libaccess.sjlibrary.org/10.1007/978-3-319-32552-1_26

[10] Sørensen J.N., “Blade-Element/Momentum Theory,” General Momentum Theory for Horizontal Axis Wind Turbines, Research Topics in Wind Energy, Vol 4. SpringerLink, retrieved 7 September 2020.
https://doi-org.libaccess.sjlibrary.org/10.1007/978-3-319-22114-4_7

[11] Ehrich, S., Schwarz, C. M., Rahimi, H., Stoevesandt, B., and Peinke, J., “Comparison of the Blade Element Momentum Theory with Computational Fluid Dynamics for wind turbine simulations in turbulent flow,” Applied Sciences, MDPI, retrieved 9 September 2020.
https://www.mdpi.com/2076-3417/8/12/2513/pdf

[12] Rwigema, M. K., “Propeller Blade Element Momentum Theory with Vortex Wake Deflection,” International Congress of the Aeronautical Sciences, retrieved 25 September 2020.
http://www.icas.org/ICAS_ARCHIVE/ICAS2010/PAPERS/434.PDF

[13] Stepniewski, W. Z., “Rotary-Wing Aerodynamics Vol 1 Basic Theories of Rotor Aerodynamics,” Strives Uploads, retrieved 10 September 2020.
https://strivesuploadsprod.s3.usgovwest1.amazonaws.com/19790013868/19790013868.pdf?AWSAccessKeyId=AKIASEVSKC45ZTTM42XZ&Expires=1601453411&Signature=PH0XUrtUxURWI30GUpLltN4Zbow=

[14] Sørensen, J. N., “Optimum Rotor Performance Based on Momentum Theory,” General Momentum Theory for Horizontal Axis Wind Turbines, Cham: Springer International Publishing, 2016, pp. 43–58, retrieved 14 September 2020.
https://link.springer.com/chapter/10.1007/978-3-319-22114-4_5

[15] McCrink, M. H., and Gregory, J. W., “Blade Element Momentum Modeling of Low-Re Small UAS Electric Propulsion Systems,” Aerospace Research Central, retrieved 18 September 2020.
https://arc.aiaa.org/doi/pdf/10.2514/6.2015-3296

[16] “Title 14 CFR Part 107,” Fact Sheet – Small Unmanned Aircraft Systems (UAS) Regulations (Part 107), retrieved 22 October 2020.
https://www.faa.gov/news/fact_sheets/news_story.cfm?newsId=22615

[17] Theys, B., Dimitriadis, G., Hendricks, P., and De Schutter, J., “Influence of propeller configuration on propulsion system efficiency of multi-rotor Unmanned Aerial Vehicles,” ResearchGate, retrieved 15 October 2020. https://www.researchgate.net/publication/304817071_Influence_of_propeller_configuration_on_propulsion_system_efficiency_of_multi-rotor_Unmanned_Aerial_Vehicles

[18] Bershadsky, D., Haviland, S., and Johnson, E. N., “Electric Multirotor Propulsion System Sizing for Performance Prediction and Design Optimization,” Semantics Scholar, retrieved 9 October 2020. https://pdfs.semanticscholar.org/98dc/c5211c6c600a344454bbc67d5835cf8e7031.pdf

[19] Khalid, A. S., “Development and implementation of rotorcraft preliminary design methodology using multidisciplinary design optimization,” Ph.D. Dissertation, Aerospace Engineering Department, Georgia Institute of Technology, December 2006.

[20] Johnson, W., “Blade Element Theory, Helicopter Theory”, New York: Dover Publications, 1980, pp. 45–56, retrieved 2 November 2020.

[21] Gur, O., “Comparison between Blade-Element models of propellers,” ResearchGate Database, retrieved 14 November 2020. https://www.researchgate.net/publication/290308462_Comparison_between_Blade-Element_models_of_propellers

[22] Ghoddoussi, A., “A more comprehensive database for propeller performance validations at low Reynolds numbers,” Ph.D. Dissertation, 2016, retrieved 8 February 2020. https://soar.wichita.edu/bitstream/handle/10057/12635/d16011_Ghoddoussi.pdf?sequence=1&isAllowed=y

[23] Silvagni M., Chiaberge M., Tessari F., “Analysis & Modelling of Powertrain Components for an Efficient UAV Design,” Advances in Service and Industrial Robotics, RAAD 2017. Mechanisms and Machine Science, vol 49. Springer, Cham, retrieved 22 March 2020. https://doi-org.libaccess.sjlibrary.org/10.1007/978-3-319-61276-8_53

[24] D. Shi, X. Dai, X. Zhang and Q. Quan, "A Practical Performance Evaluation Method for Electric Multicopters," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 3, pp. 1337-1348, June 2017, doi: 10.1109/TMECH.2017.2675913, retrieved on 10 April 2020. https://www.researchgate.net/publication/314200017_A_Practical_Performance_Evaluation_Method_for_Electric_Multicopters

6 Appendix A

```
% AE 295B MSAE Project
% Analysis of a Racing Quadcopter
% Description: Script develops output values for theoretical
quadcopter performance flight characteristics.
% Author: Brayan Mendez

close all, clear all, clc

%% Manufacturer Load Test Data

% Throttle values for 1380 Kv motor, percent value
Throttle = [30 40 50 60 70 80 90 100];

% Voltage values for 1380 Kv motor, measured in Volts
Voltage = [20.0 20.0 20.0 20.0 19.9 19.9 19.8 19.8];

% Current values for 1380 Kv motor, measured in Amps
Current = [2.8 4.5 8.1 10.9 17.5 25.7 34.9 43.7];

% Rotor velocity for 1380 Kv motor, measured in rev/min
RPM = [7681 9128 11195 12590 14345 15991 16550
18820]/9.5492965964254;
Rps = RPM/0.10472; % measured in radians per second

% Thrust values for 1380 Kv motor, measued in Newtons
Thrust = [.315 .507 .708 .865 1.152 1.500 1.942 2.178]*9.81;

% Input Power for 1380 Kv motor, measured in Watts
PowerIn = [56.00 90.00 162.00 218.00 348.25 511.43 691.02
865.26];

% Thrust efficieny values for 1380 Kv motor, measured in
grams/watts
ThrustEff = [5.63 5.63 4.37 3.97 3.31 2.93 2.81 2.52];

% Torque values for 1380 Kv motor, measured in N-m
Torque = PowerIn./(RPM);

%% Propeller Modeling

% Constants for propeller modeling obtained from relevant
literature
rho = 1.225; % Density at sea level
gamma = 0.9; % correction coefficient
```

```
alpha0 = 0; % zero lift angle
epsilon = 0.85; % correction factor from downwash
zeta = 0.6; % propeller parameter
K0 = 6.28; % lift gradient
C_fd = 0.02; % zero-lift drag coefficient
A = 8.89; % Aspect Ratio
e = 0.9; % oswald factor

% Propeller Dimensions and Constants
c = 0.02; % chord length in meters
Dp = 0.1778; % diameter in meters
Hp = 0.1016; % propeller pitch in meters
Bp = 3; % number of blades

% Average rotor linear velocity
U = (pi*zeta*Dp.*RPM)/60; % air speed of the blade

% Blade area
S = (Bp/2)*gamma*Dp*c;

% Pitch/blade angle
theta = atan(Hp/(pi*Dp));
```

```
% Absolute angle-of-attack
alpha_ab = epsilon*theta;

% Lift coefficient
cl = (K0*alpha_ab)/(1+(K0/(pi*A)));
%cd = C_fd + ((pi*A*K0^2)/(e))*((epsilon*atan(Hp/(pi*Dp))-
alpha0)/((pi*A)-K0)^2);

% Drag coefficient
cd = C_fd+((cl^2)/(pi*A*e));

% Torque Coefficient
CQ = (1/(8*A))*(pi^2)*cd*(zeta^2)*gamma*(Bp^2);
%Q = (1/4)*rho*Bp*cd.*(U.^2)*S*(Dp);

% Thrust Coefficient
CT =
((1/4)*(pi^3)*gamma*(zeta^2)*Bp*K0)*(((epsilon*atan(Hp/(pi*Dp)))
)/((pi*A)+K0));

% Calculation of Thrust (N) and Troque (N-m)
T = (CT*rho.*((RPM/(2*pi)).^2)*(Dp^4)); % Thrust
Q = CQ*rho.*((RPM./(2*pi)).^2)*(Dp^5);  % Torque

%% Electric Motor Model

% Motor constants obtained from manufacturer specifications
Im0 = 0.95; % amps, no-load current
Um0 = 10; % volts, no-load supply voltage
Rm = 0.06953; % ohms, internal resistancew = RPM;
Kv = 1380;
w = RPM;

% M = Torque, (propeller torque = motor mechanical torque)
M = Q;
```

```
%M = (PowerIn*60)./(2*pi.*RPM);

% Back-electromotive force constant
KE = (Um0-Im0*Rm)/(Kv*Um0);

% Calculating equivalent motor input current and equivalent
supply voltage
Um = Rm*(((M*Kv*Um0)/(9.55*(Um0-Im0*Rm)))+Im0)+((Um0-
Im0*Rm)/(Kv*Um0))*w; % Equivalient supply voltage
%U_m = KE*w+Im*Rm;

Im = ((M*Kv*Um0)/(9.55*(Um0-Im0*Rm)))+Im0; % Equivalent motor
input current

%% ESC Modeling

%ESC constants obtained from manufacturer specifications
Ub = Voltage; % battery voltage measured in volts
Ib = Current; % battery current measured amps
Rb = (Ub-Um)/Im; % battery resistance measured in Ohms
Re = (Ub-Um)/Current; % esc resistance measured in Ohms

% Duty cycle measured in percent
Dc = (Um +Im*Re)/Ub;

% Supplied current measured in Amps
Ie = Dc*Im;

% Supplied voltage measured in Volts
Ue = Ub-(Ie-Im)*Rb;

figure()
plot(RPM,Ie)
hold on
plot(RPM,Ue)

%% Battery Modeling
Cb = 5000; % battery capacity measured in mAh
Cmin = 750; % min battery capacity due to safety margin,
measured in mAh
Kb = 100; % maximum discharge rate, measured in C
% Ib is the battery current measured in Amps

for Ib = Current
    if Ib <= Kb*Cb
        T_end = ((Cb-Cmin)/Ib)*(60/1000); % measured in minutes
    end
end

%% Regression Line Modeling

% Constants Derived from Hardware

kv = 1380; % motor speed constant
k_torque = 60.0/ (2*pi*kv); % torque coefficient
%fprintf("k_torque = %d \n", k_torque) % Nm/A

ThrottlePercent = [30, 40, 50, 60, 70, 80, 90, 100]; % Throttle
Percent/PWM Max
omega = [7681, 9128, 11195, 12590, 14345, 15991, 16550,
18820]*2*pi/60; % Rad/sec
StaticThrust = [0.315, 0.507, 0.708, 0.865, 1.152, 1.500, 1.942,
2.178].*9.81; % N
Current = [2.8, 4.5, 8.1, 10.9, 17.5, 25.7, 34.9, 43.7]; % A

%% Solve for thrust coefficient, curve fit

% Least Squares Curve Fit Pure Quadratic, Thrust

%F = k_f.*omega.^2.45;
factor_kf = 2.45;
k_f = StaticThrust / omega.^factor_kf;

% Polynomial curve fit, Thrust

x1 = omega;
y1 = Thrust;

coefs1 = polyfit(x1,y1,2);
curve_thrust = polyval(coefs1,x1);

% Linear Regression Curve Fit

n = 8;
x = omega;
y = Thrust;

sum_xy = sum(x(:).*y(:));
sum_x = sum(x(:));
sum_y = sum(y(:));
sum_x_squared = sum((x(:)).^2);
squared_sum_x = sum(x(:)).^2;
```

```
m_thrust = (n*sum_xy-(sum_x*sum_y))/(n*sum_x_squared-
squared_sum_x);
b_thrust = (sum_y-m_thrust*sum_x)/n;

y_thrust = m_thrust*omega + b_thrust;

figure
plot(omega,y_thrust)
hold on
plot(omega, Thrust)
hold on
plot(omega,T)
hold on
plot(omega,k_f*omega.^factor_kf)
hold on
plot(x1,curve_thrust)
hold off
```

```
xlabel('Omega (rad/sec)')
ylabel('Thrust(N)')
title('Thrust vs Omega Regression Line')
legend('Regression', 'Load Test Data', 'Theoretical','Least
Squares','Polynomial Curve Fit')
%% Solve for throttle percent coefficient, curve fit

% Least Squares Curve Fit Pure Quadratic, Throttle Percent

% Throttle = k_throttle*omega^1.3
factor_kt = 1.3;
k_throttle = ThrottlePercent/ omega.^factor_kt;

% Polynomial curve fit, Throttle

x2 = omega;
y2 = Throttle;

coefs2 = polyfit(x2,y2,2);
curve_throttle = polyval(coefs2,x2);

figure
plot(x2,y2,'r',x2,curve_throttle)
hold on
plot(omega, k_throttle.*omega.^factor_kt)
hold off

xlabel("Omega (rad/sec)")
ylabel("Throttle Percent (%)")
title('Throttle Percent vs Omega')
legend('Load Test Data','Ploynomial Curve Fit','Least Squares
Curve Fit Pure Quadratic')
```

```
%% Solve for torque coefficient, curve fit

% Least Squares Curve Fit Pure Quadratic, Torque

% Torque = k_torque*omega^1.3
factor_ktor = 2.35;
k_tor = Torque/ omega.^factor_ktor;

% Polynomial curve fit, Torque

x3 = omega;
y3 = Torque;

coefs3 = polyfit(x3,y3,2);
curve_torque = polyval(coefs3,x3);

% Linear Regression Curve Fit, Torque

x2 = omega;
y2 = Torque;

sum_xy2 = sum(x2(:).*y2(:));
sum_x2 = sum(x2(:));
sum_y2 = sum(y2(:));
sum_x_squared2 = sum((x2(:)).^2);
squared_sum_x2 = sum(x2(:)).^2;

m_torque = (n*sum_xy2-(sum_x2*sum_y2))/(n*sum_x_squared2-
squared_sum_x2);
b_torque = (sum_y2-m_torque*sum_x2)/n;

y_torque = m_torque*omega + b_torque;

figure
plot(omega,y_torque)
hold on
plot(omega,Torque)
hold on
plot(omega,Q)
hold on
plot(omega, k_tor.*omega.^factor_ktor)
hold on
plot(x3,curve_torque)
hold off

xlabel('Omega (rad/sec)')
ylabel('Torque(Nm)')
```

```
title('Torque vs Omega Regression Line')
legend('Regression', 'Load Test Data', 'Theoretical','Least
Squares Curve Fit Pure Quadratic','Polynomial Curve fit')

%% Solve for current coefficient, curve fit

% Least Squares Curve Fit Pure Quadratic, Current

%Current = k_iw*omega^1
factor_k_iw = 3.25;
k_iw = Current/(omega.^factor_k_iw);

%fprintf("k_iw = %d \n", k_iw)

% Polynomial curve fit, Current

x4 = omega;
y4 = Thrust;

coefs4 = polyfit(x4,y4,2);
curve_current = polyval(coefs4,x4);

% Linear Regression Curve Fit, Current

x1 = omega;
y1 = Current;

sum_xy1 = sum(x1(:).*y1(:));
sum_x1 = sum(x1(:));
sum_y1 = sum(y1(:));
sum_x_squared1 = sum((x1(:)).^2);
squared_sum_x1 = sum(x1(:)).^2;

m_current = (n*sum_xy1-(sum_x1*sum_y1))/(n*sum_x_squared1-
squared_sum_x1);
b_current = (sum_y1-m_current*sum_x1)/n;

y_current = m_current*omega + b_current;

figure
plot(omega,y_current)
hold on
plot(omega,Current)
hold on
plot(w,Im)
hold on
plot(omega, k_iw.*omega.^factor_k_iw)
```

```
hold on
plot(x4,curve_current)
hold off

xlabel('Omega (rad/sec)')
ylabel('Current(A)')
title('Current vs RPM Regression Line')
legend('Regression', 'Load Test Data', 'Theoretical','Least
Squares','Polynomial Curve Fit')

%% Residuals and Root-Mean-Square-Error (RMSE)

% Residuals

% Thrust
ai_thrust = Thrust; % Actual values from load test data
pi_thrust = T; % predicted values from theoretical results
pj_thrust = y_thrust; % predicted values from regression models
pk_thrust = k_f*omega.^factor_kf; % predicted values from least
squares
pl_thrust = curve_thrust; % predicted values from polynomial

ei_thrust = mean((ai_thrust - pi_thrust).^2); % mean squared
Errors
ej_thrust = mean((ai_thrust - pj_thrust).^2);
ek_thrust = mean((ai_thrust - pk_thrust).^2);
el_thrust = mean((ai_thrust - pl_thrust).^2);

% Current
ai_current = Current; % Actual values from load test data
pi_current = Im; % predicted values from theoretical results
pj_current = y_current; % predicted values from regression
models
pk_current = k_iw*omega.^factor_k_iw; % predicted values from
least squares
pl_current = curve_current; % predicted values from polynomial

ei_current = mean((ai_current - pi_current).^2);
ej_current = mean((ai_current - pj_current).^2);
ek_current = mean((ai_current - pk_current).^2);
el_current = mean((ai_current - pl_current).^2);

% Torque
ai_torque = Torque; % Actual values from load test data
pi_torque = Q; % predicted values from theoretical results
pj_torque = y_torque; % predicted values from regression models
pk_torque = k_tor*omega.^factor_ktor;
```

```
pl_torque = curve_torque; % predicted values from polynomial

ei_torque = mean((ai_torque - pi_torque).^2);
ej_torque = mean((ai_torque - pj_torque).^2);
ek_torque = mean((ai_torque - pk_torque).^2);
el_torque = mean((ai_torque - pl_torque).^2);

% RSME thrust

RMSE_Thrust_theoretical = sqrt(ei_thrust);
RMSE_Thrust_regression = sqrt(ej_thrust);
RMSE_Thrust_leastsquares = sqrt(ek_thrust);
RMSE_Thrust_polynomial = sqrt(el_thrust);

% RSME Current

RMSE_Current_theoretical = sqrt(ei_current);
RMSE_Current_regression = sqrt(ej_current);
RMSE_Current_leastsquares = sqrt(ek_current);
RMSE_Current_polynomial = sqrt(el_current);

% RSME Torque

RMSE_Torque_theoretical = sqrt(ei_torque);
RMSE_Torque_regression = sqrt(ej_torque);
RMSE_Torque_leastsquares = sqrt(ek_torque);
RMSE_Torque_polynomial = sqrt(el_torque);


%% Flight Test Data Log1

%type Log1.txt;

Log1 = readmatrix('Log1.txt');

roll_desired = Log1(:,8);
pitch_desired = Log1(:,9);
yaw_desired = Log1(:,10);

roll_imu = Log1(:,11);
pitch_imu = Log1(:,12);
yaw_imu = Log1(:,13);

x_position_des = Log1(:,14);
y_position_des = Log1(:,15);
z_position_des = Log1(:,16);
```

```
x_position_est = Log1(:,17);
y_position_est = Log1(:,18);
z_position_est = Log1(:,19);

TM1 = Log1(:,20);
TM2 = Log1(:,21);
TM3 = Log1(:,22);
TM4 = Log1(:,23);

RPM1 = Log1(:,24);
RPM2 = Log1(:,25);
RPM3 = Log1(:,26);
RPM4 = Log1(:,27);

I1 = Log1(:,32);
I2 = Log1(:,33);
I3 = Log1(:,34);
I4 = Log1(:,35);

TQ1 = Log1(:,36);
TQ2 = Log1(:,37);
TQ3 = Log1(:,38);
TQ4 = Log1(:,39);

time_recorded = Log1(:,3)/1E9;
timestep =Log1(:,7);

figure()
plot(timestep,roll_imu)
hold on
plot(timestep,roll_desired)
hold off

xlabel('Time (sec)')
ylabel('Roll (deg)')
title('Desired vs IMU Roll Response')
legend('IMU','Desired')
axis([40 80 -40 40])

figure()
plot(timestep,pitch_imu)
hold on
plot(timestep,pitch_desired)
hold off

xlabel('Time (sec)')
ylabel('Pitch (deg)')
```

```
title('Desired vs IMU Pitch Response')
legend('IMU','Desired')
axis([40 80 -40 40])

figure()
plot(timestep,yaw_imu)
hold on
plot(timestep,yaw_desired)
hold off

xlabel('Time (sec)')
ylabel('Yaw (deg)')
title('Desired vs IMU Yaw Response')
legend('IMU','Desired')
axis([40 80 -0.8 0.8])


%% Flight Test Data Log2

%type Log2_HighRPM.txt;
Log2 = readmatrix('Log2_HighRPM.txt');

roll_desired1 = Log2(:,8);
pitch_desired1 = Log2(:,9);
yaw_desired1 = Log2(:,10);

roll_imu1 = Log2(:,11);
pitch_imu1 = Log2(:,12);
yaw_imu1 = Log2(:,13);

x_position_des1 = Log2(:,14);
y_position_des1 = Log2(:,15);
z_position_des1 = Log2(:,16);

x_position_est1 = Log2(:,17);
y_position_est1 = Log2(:,18);
z_position_est1 = Log2(:,19);

TM11 = Log2(:,20);
TM21 = Log2(:,21);
TM31 = Log2(:,22);
TM41 = Log2(:,23);

RPM11 = Log2(:,24);
RPM21 = Log2(:,25);
RPM31 = Log2(:,26);
RPM41 = Log2(:,27);
```

```
I11 = Log2(:,32);
I21 = Log2(:,33);
I31 = Log2(:,34);
I41 = Log2(:,35);

TQ11 = Log2(:,36);
TQ21 = Log2(:,37);
TQ31 = Log2(:,38);
TQ41 = Log2(:,39);

time_recorded1 = Log2(:,3)/1E9;
timestep1 =Log2(:,7);

figure()
plot(timestep1,roll_imu1)
hold on
plot(timestep1,roll_desired1)
hold off

xlabel('Time (sec)')
ylabel('Roll (deg)')
title('Desired vs IMU Roll Response')
legend('IMU','Desired')
%axis([40 80 -40 40])

figure()
plot(timestep1,pitch_imu1)
hold on
plot(timestep1,pitch_desired1)
hold off

xlabel('Time (sec)')
ylabel('Pitch (deg)')
title('Desired vs IMU Pitch Response')
legend('IMU','Desired')
axis([40 80 -3E-4 3E-4])

figure()
plot(timestep1,yaw_imu1)
hold on
plot(timestep1,yaw_desired1)
hold off

xlabel('Time (sec)')
ylabel('Yaw (deg)')
title('Desired vs IMU Yaw Response')
```

```
legend('IMU','Desired')
axis([40 80 -14E-4 14E-4])

figure()
plot(timestep1,TM11)
hold on
plot(timestep1,TM21)
hold on
plot(timestep1,TM31)
hold on
plot(timestep1,TM41)
hold off

xlabel('Time (sec)')
ylabel('Force (N)')
title('Force Response from each Motor')
legend('TM1','TM2','TM3','TM4')

figure()
plot(timestep1,I11)
hold on
plot(timestep1,I21)
hold on
plot(timestep1,I31)
hold on
plot(timestep1,I41)
hold off

xlabel('Time (sec)')
ylabel('Current (A)')
title('Current Response from each Motor')
legend('IM1','IM2','IM3','IM4')
```