Design and Evaluation of a Fixed-Pitch Multirotor UAV with a Nonlinear Control Strategy

Kenneth Edward Kroeger

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science

in

Mechanical Engineering

Kevin B. Kochersberger, Chair

Alexander Leonessa

Craig A. Woolsey

May 6, 2012
Blacksburg, Virginia

Keywords: UAV, HexaCopter, Integral-Backstepping Control, Extended Kalman Filter, IMU, GCS

Design and Evaluation of a Fixed-Pitch Multirotor UAV with a Nonlinear Control Strategy

Kenneth E. Kroeger

(ABSTRACT)

The use and practical applications of small UAV systems has continually grown in the past several years in both the public and private sectors. These UAV systems are used for not only defensive purposes, but for commercial applications such as exterior bridge and home inspections, wildlife/wildfire management and observation, conservation exercises, law-enforcement, radio-repeating operations, and a wide variety of other uses that may not warrant the use, expense, space constraints, or risk of a manned aircraft. This thesis focuses on the design of a fixed pitch multirotor UAV system for use in furthering research projects and facilitating payload data collection from a flying platform without the expense or risk of testing with available larger UAV systems.

The design of a multirotor UAV system with a flight control scheme, communication architecture and hardware, electrical architecture and hardware, and mechanical design is presented. An Extended Kalman Filter (EKF) strategy is implemented aboard a developed Inertial Measurement Unit (IMU) to estimate vehicle state. Experiments then validated the estimates from the EKF through a comparative approach between the developed unit and a commercial unit. A nonlinear flight control system is implemented based on an Integral-Backstepping control strategy. The flight control strategy was then fully simulated and exhaustively tested under a variety of external disturbances and initial conditions from a fully dynamic modeled environment. Parameters about the vehicle were experimentally determined to increase the accuracy of the model which would increase the chances of successful flight operations.

Flight demonstrations were conducted to evaluate the abilities and performance of the control system, along with testing the interface abilities and reliability between a universal ground

control station (UGCS) and the aircraft. Lastly, the model was revisited with the input data from the flight control experiment and the output captured was evaluated against the output of the model system to evaluate effectiveness, reliability, and accuracy of the model. The results of the comparison showed that the computer simulation was accurate in predicting attitude and altitude of the vehicle to that of the realized system.

Acknowledgments

First, I would like to acknowledge and thank my adviser Dr. Kochersberger, without him I would never have been able to have the opportunity to successfully pursue higher education. He had supported me getting accepted into graduate school, and given me numerous opportunities to express my engineering creativity through numerous projects in the lab, the patience for allowing a mechanical engineer to build electrical systems, and the overall support needed to allow an engineer to explore the possibilities of technologies in unmanned systems. I would also like to thank my advising committee Dr. Leonessa and Dr. Woolsey for the valuable insight and knowledge of dynamic modeling and control systems. Our conversations about the expansions of technologies and with it, the capabilities and possibilities of research and new ideas have been enlightening and inspiring.

I am forever grateful for the support and love my parents and family have always given me. Dad, you have always pushed me to strive for greater challenges, constantly push oneself in every way possible, and always push for improvement. Mom, you have always supported and encouraged me through the toughest times of school when in my mind there was no apparent end to the struggle in sight. Brother, hopefully I have shown you the importance and rewards of pursuing higher education and I hope to always help you in pursuing your goals in learning and in life. To my loving wife, I appreciate the patience, support, love, and encouragement you have given me through this endeavor. Hopefully my nerves have calmed at the conclusion of this work and you will no longer have to hear me yelling at inanimate electronics or computer code asking them why won't they work or listen to me. I ensure you the my sanity was only temporarily missing, but I am still here, and I love you. To Rory puppy, you have seen me through the toughest times in this project. You had stayed up with me through the late hours and the early mornings. Even though you were never able to understand what was happening, you had always supported me by being at my side wherever I went. I am sorry about the missed opportunities of a walk or run when the weather was

nice. You will be missed, and you will always be in our hearts and thoughts forever.

Lastly, USLers; it has been a great time with every one of you, including those who have left and those that still remain. You have been there with anything that the time had called for. In any way needed you would provide the technical advice where my knowledge may have lacked, the listening ear when I needed someone to bounce ideas off of, the motivational and competitive support to strive for the best, and the much needed distraction and constant reminder to just have fun.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

This thesis presents a portion of work conducted to further the Unmanned Systems Lab's continuing efforts of exploring and developing UAV technologies. UAV technologies have in the past century grown at rapid rates as MEMS and computational abilities have consistently become more powerful, lighter, and smaller. However, before the technological and digital ages had begun, these technologies gathered their first recorded efforts in the late 1800's with the capture of aerial imagery from released kite and balloon platforms. With the growing global conflicts of WWI, technologies had grown to develop the first military deployed radio controlled platform for reconnaissance efforts, dubbed the Queen Bee. Later, global conflicts had again pushed efforts to remove manned operators from battle with the introduction of Germany's V1 rocket system of WWII. DARPA's efforts in the Cold War had continued pushing UAV developments for reconnaissance purposes that would remove pilots from dangers of both a physical and politically heated environment [1]. Today, UAV technologies have achieved milestones inconceivable to its predecessors with flight platforms such as the Shadow TUAS achieving its 500K hour flight mark in May '10 and achieving the 600K hour flight mark in February '11. Another milestone in operational endurance has been the MQ Predator series accumulating over 1M flight hours. Other deployments of UAV technologies have grown to include further efforts in combat, reconnaissance, conservation, search and rescue, exploration, and research. The ideas in which a UAV can be used are seemingly endless at this current time as their boom in both the private and public sectors

continue to develop.

The primary goal of this research is to explore, develop, and implement a small UAV system with a inner-loop stabilization flight controller. *Note that many in the research community have been using the term MAV to denote their UAV development, however, this author assumes the DARPA definition of MAV to be smaller than* $6in^3$*, therefore, we shall denote this a small UAV.* The inner-loop flight controller governs the internal stabilization of the UAV while maintaining control of the aircraft's translational and angular velocities. Therefore, a fundamentally stable, sound, and reliable inner-loop is required before considerations of an outer-loop can be made for trajectory/waypoint following. This type of nested/hierarchy control structure is commonly deployed as to separate and decouple efforts of each control system. Typically, the inner-loop control system focuses on novel techniques for stability and require information fusion from the vehicle's available sensory network with much higher update rates than its outer-loop counterpart. Thus, inner-loop control demonstration will not only require system architecture development for the vehicles power, communication, and physical construction, but also require that the system can support and collect sensory data as needed to understand and govern the vehicles attitude. The outer-loop typically builds further state information about the aircraft from information obtained in the inner-loop and other peripheral sensors. Thus, as a progressive build-up of a UAV system, this work will start at the fundamental step of system development and inner-loop flight control.

1.2 Motivation

The primary goal of this research is to develop a system with preliminary efforts that could be used in the future to develop a low-cost, useful, and robust UAV system. The preliminary efforts seen here will hopefully continue into a fully developed system and see its fielded usefulness as the author further pursues aircraft technologies and autonomous systems. Currently, UAV deployments from the lab typically consist of a Yamaha RMAX, Aeroscout B1-100, Sig 110, or Bergen Industrial Twin aircraft with payload platforms and technologies that engineers from the lab are testing and developing. The above platforms inherently require large areas of operation and considerable man-power to ensure a safe and successful flight. Also, these platforms require experienced pilot operators for flights and are large monetary risks in the event an operation deviates from as planned. However, benefits include large payload capacities and long flight endurance times. Figure 1.1 below highlights

several of these platforms available from the lab.

Figure 1.1: From left to right: Yamaha RMAX, Aeroscout B1-100, and Bergen Industrial Twin

The motivation of this project is to begin an effort to develop an intuitive UAV system for operations throughout the lab for testing and development of other payload and UAV technologies. Currently, many of the projects fielded in the lab have not required the full capabilities from the aerial vehicle platforms used. Typically, operations have consisted of 15 minute flights, minimal aircraft movement beyond typical flight envelopes (non-aggressive maneuvers), and have carried minimal payload hardware such as camera systems and radios. Therefore, the question posed is, why take such unnecessary costly risks and time when there are smaller and easier to operate flight platforms that could be available?

The answer to this question typically has been because of the availability and reliability of the above aircraft systems. However, this author proposes the usefulness and cost effectiveness of a smaller UAV platform. To date, flight operations have not deviated outside the typical flight envelope for the aircraft, with a majority of flights consisting of a helicopter platform in the hover position or low forward velocities; fixed wing aircraft have been used in a similar manner (less the hover condition). Therefore, the author had proposed researching a multirotor based platform for its ease in mechanical complexity, and ability to both hover and achieve low flight velocities. While there are many UAVs in existence for a variety of uses, a fixed pitch multirotor platform will be the concentration of this work. A typical helicopter with variable pitch rotor platform has been dismissed from this work as the helicopter contains a mechanical swash plate that introduces mechanics and modeling that are not desired for development.

1.3 Description of Specific Work, Objectives, and Goals

It can be seen from the above literature review, that the general conclusion of many of prior works in this field had fairly similar outcomes. Typically, most authors have recommended similar improvements that could be made towards full system performance, such as integration of outer-loop control, integration of GPS and camera systems for situational awareness, better computational power and higher compactness of electrical components, higher lifting capacity for larger payload integration, longer flight durations and improvements of electrical efficiencies, further uncontrolled environmental testing, and typically better implementation and robustness of communication protocols for operators understanding of vehicle performance and decision making.

Therefore, the general goals of this work will be to:

- Design and develop a UAV system. This UAV system will have to be able to handle not only its own weight for flight, but also be able to incorporate necessary additional payloads for practical uses and further flight research. These payloads may include a LIDAR system, scanning IR system, cameras, additional flight computers, and other typical flight research payload platforms. Also, this UAV system will require a reasonable amount of flight endurance so that using this vehicle as a test platform is a viable option.
- Derive/Test/Implement a reliable way in estimating the vehicle state. It can be seen from the literature review that many works have computational complications as a result of trying to estimate far greater states than potentially what is needed for a minimalistic control scheme and inner-loop control.
- Design/Test/Implement necessary flight electronics to reduce weight and increase efficiencies.
- Implement an efficient and robust communication protocol for data transfer between the operator on the ground and the air vehicle. This will include an effective way to present information to the operator so that he/she may understand vehicle status and vehicle information.
- Develop a reliable and accurate mathematical model environment in which to test derived control strategies aboard the vehicle.

- Implement/Test flight control strategies in an untethered environmental setting and compare realized system performance to that of the mathematical model performance.

The research discussed in this thesis will address several key aspects on the road to autonomy for a UAV system:

1. Literature Review (Chapter 2).

2. The design, fabrication, and system architecture development of a UAV system with a focus towards a fixed pitch HexaCopter platform(Chapter 3).

   - Designing a UAV system has many aspects that need to be considered from a holistic point of view. A suitable UAV platform to achieve the goals stated above must have the ability to support requested flight times, carry small payload platforms, and be intuitive for an operator to use and manipulate. Designs will be modeled and created in this work to accommodate foreseen desired functionality. Electrical systems will be designed and integrated to accommodate necessary flight hardware and foreseeable payload systems. Lastly, a communications architecture is presented and implemented aboard the UAV system to communicate information to the operator at the developed UGCS terminal.

3. Coordinate Frame and Attitude Determination (Chapter 4).

   - Coordinate frame (CF) establishment is the basis for mathematically modeling and developing a control system for the aircraft. The CF will establish an architecture through which reference parameters of the vehicle can be understood and communicated throughout the system and to operators. Understanding the vehicle attitude is important for the flight controller to take necessary corrective actions to stabilize the system to the desired point; therefore, a developed IMU is discussed with an implemented attitude estimation filter. The mathematical rigor of the developed estimation filter will be presented along with comparative results of the developed IMU performance to a commercially available unit.

4. Model Development and Simulation (Chapter 5).

   - This section details the model/plant development of a multirotor UAV platform. The design of a reliable and accurate model is fundamental in simulating a flight

controller because this mathematical set of equations describes the flight dynamics of the model in question. This model describes the motion and response of the aircraft as a result of a change in aerodynamic, propulsive, and gravitational forces. This section will discuss assumptions made in model development, and highlight important steps in the process of deriving the differential state equations of the system.

5. Controller Development and Simulation (Chapter 6).

   - Once a model is developed, it is desired that the plant behave in a manner that the operator desires. Therefore, based on sensory feedback, a controller generates inputs to the plant based on reference inputs such that the system achieves performance in the desired manner. This section details the derivation and integration of a Integral-Backstepping flight controller on the model developed system and evaluates the controller's performance and abilities to become a viable flight control solution for a multirotor platform. A short segment introduces the theory of stability and robustness for nonlinear control and evaluates these parameters against the presented controller design.

6. Real-time Implementation and Results (Chapter 7).

   - After the controller had been validated through a simulated model, the complete attitude estimation and flight control system was realized in hardware. This chapter discusses the final performance specifications the system is capable of and details how this performance adheres to the simulated system. Lastly, this section will highlight a final field test of the entire system in both a structured and unstructured environment and present flight results and imagery of the implementation.

7. Future work and Conclusion (Chapter 8)

   - The work will conclude with future recommendations and potential further research avenues for development so that the platform may become a viable UAV replacement for larger aircraft systems in the Unmanned Systems Lab.

Chapter 2

Literature Review

There are many aspects involved in the successful design and implementation of an autonomous UAV system. Aspects include everything from mechanical design of an airframe, motor selection and vibration isolation, electrical design of necessary components for power distribution, signal conditioning, signal processing, and software and system architecture developments to govern and control the actions of the UAV to ensure both a safe and successful flight. This chapter contains an overview of work previously done in the area of fixed multirotor platforms from various public and private domain applications. Since the system architecture itself is large, many varying fields were used in the completion of this work. The next several sections are broken down in a similar fashion. First, commercial platforms are discussed along with the difficulties and challenges of commercialization for use in furthering the research community. Next, sensory fusion is an important central role in UAV stabilization, therefore several papers are reviewed discussing filtering techniques to better understand vehicle attitude. Lastly, highlights of other works contain explanations of modeling and control techniques for a multirotor UAV system and several recommendations that those authors had provided about future developed aerial platforms are presented.

2.1 Commercialization of Platforms and Sensory Systems

Unmanned Aerial Vehicles have been a well studied area for their use in a variety of potential applications. These types of vehicles continue to become smaller, lighter, and less expensive as researchers continue to develop them to their full potentials. Fixed-pitch UAVs have been

booming as research platforms for control development and payload abilities as increases in technologies continue in strides as developments are made in the private and public domains.

2.1.1 Commercially Public Platforms

The commercially available UAVs on the market today have peaked interests of hobbyists, photographers, and many people of the like who enjoy flying a stabilized platform without the difficulties typically associated with R/C flight. The enjoyment and pitch for many is that with the on-board flight control system, anyone can fly these vehicles to accomplish a given task. Thus, their markets have opened to the masses to include hobbyists who enjoy flying new vehicles, engineers who enjoy the "wow factor" and are interested towards new discoveries and uses. Another marketable group has been photographers and cinematographers. Through the use of commercially available UAV systems, people have been able to capture still-frames and footage into areas never thought possible or traversable by typical camera crews.

Draganflyer Innovations Inc. has been a public company inventing varying UAV systems that have included various forms of quadrotors and blimp platforms for the commercial user since 1998. Their most recent commercial designs and company flagships have been the Draganflyer X4, X6, and X8. Their platforms have been adopted into law enforcement for surveillance operations to ensure officer safety when the scenario is unknown and requires an eye in the sky. Draganfly has also entered the research university level working with engineers at MIT and Vanderbilt. Masses for their platforms range from 700g to 1.7kg, while the size of their platforms are approximately 25 inches in diameter without the rotors. Also, flight times consist of anywhere from 20~80 minutes depending on payloads and battery size installed. However, these respectable and sleek platforms come complete with ground station for full autonomous operations with hefty price tags, making them expensive research and development platforms. However, their robustness and development qualities make them desirable to many.

Microdrones had formed more recently in 2005, and markets two primary platforms of UAV systems. The MD4-1000 and the MD4-200 are both quadrotor based platforms each featuring respectable payloads and flight times ranging from 20 to most recently 80 minutes. Their flagship is the large MD4-1000 with a platform diameter of 1030mm without the rotors and a take-off mass without payload of 2.56kg. However, an integrated operating system

with ground station and flight controller on this platform will run approximately $53K. As developed and stable as this platform is for payload integration and potential research options, the high price tag again makes this platform hard to achieve and integrate into a research setting. Although there are many other large commercial companies producing multirotor platforms, these two companies appear to have had the most success at full engineered solutions. For hobbyists, the do-it-yourself movement has exploded in the last few years providing commercially supported platforms with the cost benefit of assembly and open-source software and hardware components. The next section touches on two large open-source movements with commercial backing.

2.1.2 Open Source Commercial Platforms

First, approximately in 2008, Mikrocopter had started production on their quadrotor. Quickly expanding to a HexaCopter platform and from there they developed a niche in the market of film and photography with multirotor UAV systems. All of their vehicles provide modification abilities for pan-tilt camera mechanisms and offer varieties of landing gear options for user customizable capabilities. However, following true to the open-source and DIY mentality, frames come disassembled, sensors uncalibrated, and electronics unsoldered. This allows for cost-reductions, bringing the cost down to $2K for a simple HexaCopter platform without radio and base station. However, the largest and most successful DIY movement in the field of UAVs appears to have been 3DRobotics.

3DRobotics commercially offers fully assembled multirotor platforms for similar price ranges as the Mikrokopter platform and offers unassembled platforms for cheaper. The advantage of this open-source community development is that the project becomes the collaboration of many engineers who typically would do these tasks as a hobby. Thus, by sharing information and the wealth of knowledge, combinations of ideas from engineers and hobbyists alike can come together and create something far greater than the individual alone. This open-source mentality massively reduces the costs of overhead and development and then in turn can sell potential ideas to gain internal revenue. For 3DRobotics, this internal revenue has gained some employees a permanent spot in the company to dedicate their time to continued development and support of the platform. A platform however, is nothing without the inner system architecture of power and processing systems.

2.1.3 Commercial Autopilots and Measurement Units

Many commercial autopilot and sensing systems are available on today's market; each featuring some vantage point over another system. Whether its cost benefits, more resolution, better GPS systems, better filtering systems for estimation of attitude, etc. If a UAV is in consideration for military and law enforcement use, a higher grade system would be chosen with an appropriate price complimenting the abilities of the system. Rather, for amateur hobbyists and photographers, a much lower quality system could be chosen with again a price point to compliment its quality. First, a common autopilot solution implemented by MIT and Stanford is the Kestrel Autopilot from Procerus Technologies. Features include a high-bandwidth controller for VTOL platforms, dead-reckoning for GPS-denied environments, 17-state EKF navigation system, and SD card data logging. However, its price point is comparable for the features at $8,500. Another company called Microstrain produces a successful line of inertial sensor systems. Two commonly used in UAV applications are the 3DM-GX3-35 & 45 models. Each are miniature AHRS systems that have an internal IMU with GPS system. For the hobbyists and DIYers out there, Sparkfun and 3DRobotics offer complete autopilot and sensory data solutions at affordable costs. However, the trade-off becomes these solutions may not be the most robust or contain the best electronics and engineering software. However, they typically perform their job and allow for operators to change and modify the hardware as they deem fit.

2.1.4 Problems with Commercialization & Research

However, as great as the above platforms and open source commercialization communities may be, it is difficult to argue the cost or time benefit in a research and development setting. If platforms and systems are purchased from outside companies, this develops a dependence from the research student on the said company for support. This support can be in the form of replacement parts, software or hardware support, and potentials for unknown electronic or software failure. This required support then dictates a schedule and perhaps forces to delay schedules of further research developments or hinders possible platform potentials. Purchasing available platforms also hinders modeling of particular characteristics for developments. This could be in the form of not being able to access the tuning matrices of filters on measurement units, as well as not gaining access to on-board software that may be needed to control the vehicle in a manner that may be most beneficial to the researcher. An

example of this would be if a researcher needed to interrupt the internal flight controller to perform another action, or perhaps develop more advanced control algorithms. Understanding proprietary software from another researcher typically consumes time and costs. This is a driving motivation for this author in developing all components in-house away from commercialization. Commercial components such as motors, props, and ESCs will be bought, however, power systems, software and system architecture, and mechanical design will all be done by the author to promote a self reliance on one architect and promote stability of the project. This promotes the idea that if problems were to arise with the system, the author will have developed the system to understand how and why the system has failed and can provide solutions and answers to potentially correct the problem in a timely manner. This makes diagnosing issues and problems easier and more effective. Also, this allows for future customization of the platform while understanding how the rest of the system will adhere and perform with the provided changes.

2.2 Data Fusion for Enhanced Attitude Estimation

The fundamental part in any control system is the ability for the controller to understand the current state of the plant. Within UAS, this involves measuring and computing a 12 state vector including, rotation, rotational speed, position, and velocity. To gather this information, a variety of sensors must first be employed in order to collect information about the vehicle. The authors of [2, 3, 4, 5] offer a segment of detailed literature about the sensors that comprise an INS. An INS is comprised of an Inertial Measurement Unit (IMU) and a Global Positioning System (GPS). The system could be further comprised of altimeters, inclinometers, and various other measurement devices that relay information about a vehicle's attitude or position in a relative or an absolute coordinate system. [2, 3] further details the breakdown of an IMU unit into its typical individual components such as the accelerometer, gyroscope, and magnetometer and the current technologies associated with each component. Integration of an IMU typically consists of a strap-down or gimballed configuration [3, 32]. However, the difficulty with these sensors is the inherent errors associated with the inertial sensors that lead to a growing error in the attitude, velocity, and position estimates of the aircraft. A table highlights these errors in [5, 6, 7] while Kumar [8] numerically quantifies the errors experimentally on a sample IMU. It can be seen from here, that in order to correctly develop an attitude estimation system, a sample model of the proposed sensor needs to be

created and simulated to ensure correct state estimation.

Literature [3, 5, 6, 8, 9, 10] has shown that there are various ways to model each of the individual sensor components that comprise an IMU. Each of these literature's had highlighted the importance of accounting for alignment errors, sensor bias/offset, sensor output scaling, non-orthogonality of the gyroscopes and accelerometers, random noise, and lastly gyro drift. Many of these generated errors are constant throughout the duration of use in an IMU and can often be determined and calibrated for prior use. However, most had chosen to use an update equation in their filtering scheme to correct for the gyroscopic drift. The next important step in the process was to determine an effective means of filtering and fusing the information from the sensors chosen to be apart of the INS. It is important to note that many of the filtering processes are very similar in derivation and typically differences only appear at the measurement/update equations of each author's filtering scheme.

There are many filtering techniques to fuse data information that is available to promote better attitude and localization of a UAV system. Simple forms of low pass filters and complimentary filters are available for improved readings of Euler angles for attitude estimation from MEMS systems. More complicated forms of filtering techniques include varying forms of the Kalman Filter. This filter can incorporate host dynamics for better state estimation, and incorporate a seemingly endless amount of sensor information if proper relation equations can be formed between each type of sensor.

Kumar[8] had proposed a Kalman Filter (KF) implementation for fusing both IMU and GPS data. He had implemented a 9 state KF using both feed-forward and feedback techniques for data filtering and correction. The author found that both of his proposed filtering techniques had a high dependence on GPS correctional information. Therefore, at times of GPS satellites not in view of the receiver, the estimated position and velocities were highly inaccurate. He also had sighted that his lack in attention to the IMU system may account for some of the error. He believed that first correcting the IMU measurements further by accounting for scale factors, bias, non-orthogonality, and drift, would aid in improving the results of each KF technique. However, Kumar's results were only simulated based on a first order disturbance input model.

However, imposing a loosely coupled GPS and IMU filtering scheme is what Ronnback[11] had proposed. In his work, he had demonstrated an extensive amount of mathematical modeling of GPS characteristics to incorporate into a proposed KF technique to better fuse

IMU and GPS information. The author had collected flight data on one of the universities aircraft and post-processed the information through the proposed filtering scheme. The author had concluded several important factors that were similar to the author above. First, he had observed that a KF technique should hold states to track the bias and drift rates of the gyroscopes. However, the note here is that with the increased number of states from 9 to 15, would dramatically increase computational operations for the processor. Lastly, the author had concluded that vibration dampening of the motors is highly important as the motor significantly decreases the potential for a desirable SNR.

Literature continues to develop that if a processor can handle the computational growth of estimated states for a filter, the accuracy of a vehicle can improve with additional various sensors. It can also be seen that there is a gradual movement from the base KF to the EKF and UKF variants as processing power increases. Authors [9, 10, 12] begin to build EKF models because of their ability to incorporate the nonlinearities of the system into their state estimate. From here, readers begin to see that ability for an architect to add a variety of sensors for additional state information and refinement of attitude estimation with improved results. Lastly, [13, 14] discuss the importance of and how to optimize processors for state estimation filtering while literature continues to incorporate more states including LIDAR and visual odometry cues as discussed in Nonami[15].

2.3 Modeling & Control Techniques

There are many forms of linear and non-linear modeling and control techniques researched and proposed for UAV systems. Typically each author models the system using either a Newtonian or Lagrangian based approach where they discuss the dynamics of the system and develop a set of EOM for the aircraft. During this process, many authors vary the factors and forces they incorporate into their model based on the desire for modeled accuracy. There are authors who present complex models that account for the aerodynamic drag affects during flight Martinez[16], blade flapping and induced vibration Wierema[17], and some [18, 19] model the DC drive motor if they have added a gear reduction. Other authors [18, 20] argue that some of these effects can be ignored because for them the multirotor platform typically remains around a hover state, therefore, does not introduce many of the effects other authors are referring to. The modeling process typically consists of the authors opinions and assumptions that they desire to make during their mathematical process and

according to this literature search, has seen little effect of whether modeling or unmodeling of these parameters caused performance issues in their simulated or realized system.

The control systems for multirotor UAVs have been presented with varying levels of robustness, and have included both linear and nonlinear control schemes. Many have presented works on PID control [20, 21, 22, 23], Backstepping techniques [20, 24, 25, 26], Sliding mode control [20, 27], Neural Networks [28], and LQR control [18, 19, 20]. Many have implemented the intended control scheme on the aircraft with success, while many others have implemented intended control on the aircraft with failure. Typically, failures have been accounted towards failure in timing sequences of developed hardware and software solutions, failure of performance from the measurement sensory system, and others have had mechanical failures in flight not realizing the stresses that the frame may be loaded with. At the end of each work, typically most authors have recommended similar improvements that could be made towards full system performance, such as integration of outer-loop control, integration of GPS and camera systems for situational awareness, better computational power and higher compactness of electrical components, higher lifting capacity for larger payload integration, longer flight durations and improvements of electrical efficiencies, further uncontrolled environmental testing, and typically better implementation and robustness of communication protocols for operators understanding of vehicle performance and decision making.

Bucholz[29] had developed a custom quadrotor platform to improve sensor performance utilizing a higher efficiency set of brushless motors. The authors had also derived a dynamic model of the platform and had designed, simulated, and implemented an LQR flight controller derived from the linearized dynamic model. The results of the simulation model had shown promise that the designed control system could work if realized correctly on their computing platform aboard the UAV, however, the authors had reported that timing executions of their routines aboard the computer were not executing properly to promote effective testing of their control design. One major design recommendation from the authors was that the motor support arms be stiff and rigid enough to handle potential torques induced by the motors. Their original designs had allowed for too much twisting and induced noise into the IMU and instability into the structure itself. A processing recommendation had included not to use floating point values in the data structure variables of any of the control loops. This caused delays in their command routine and thus only allowed control updates at approximately 25Hz. Using a fixed point precision variable structure had allowed the authors to theoretically provide control updates at 100Hz. Lastly, the authors

had recommended a highly reliable and more powerful communications architecture. The author had implemented a wireless blue-tooth module for communications, however noted that this severely limited range and bandwidth. Ultimately, they had recommended a more reliable and higher bandwidth serial communication system with checksum error detection to determine the validity and completeness of incoming messages.

Bresciani[30] had developed and modeled a quadrotor platform. Differences within this work included extensive modeling of both the vehicle and DC motor model. The DC motor model presented an interesting case as the motors had been gear reduced to decrease RPM but increase the torque capabilities of their motors. Also, body and rotational moments of inertia were calculated through rough approximations of components into common calculable shapes with corresponding look-up tables for calculation. The system was then mathematically derived through the Newton-Euler model approach. Another difference was the incorporation of both an IR and SONAR sensor to aid in altitude approximations. Lastly, this author had simulated a PID based controller for attitude and altitude stability. The author concludes that to improve the quadrotor project, an aerodynamic model should be further studied to evaluate the performance of the vehicle when performing maneuvers outside a hover condition. The author had also recommended the integration of a GPS unit for vehicle location information if available and discusses the usefulness of an integrated camera system. He suggests that the camera system poses a dual purpose for operator situational awareness and for obstacle detection and avoidance. Mechanically, the author recommends that within the initial phases of planning, the architect should alot further weight to be carried aboard the vehicle as this will progressively be the desired goal of the system with the further integration of payloads. Lastly, the author recommends replacing the remote control on the ground with a computer solution to provide feedback to the operator and to enable the operator further control of the vehicle.

The author of this thesis had found in his literature search that Bouabdallah[20] had presented the most extensive test of varying control efforts compiled into a single source. The convenience of this was that with a single developed model of a quadrotor platform, he was able to present comparisons of various control systems and highlight positives and negatives of each type. When comparing an authors control system to another author's control system, it is often hard to evaluate performance metrics because of differences in the way the vehicle was modeled, parameters that were accounted for in the mathematical model, noise levels added to the measurement matrix, and various other differences that make comparisons diffi-

cult in determining what work has been done out there and what is a conclusive result. In his work he had modeled the quadrotor system using the Lagrangian technique, and developed a mathematical model to describe the system for later use in controller development. Bouabdallah continues to simulate and bench test designs of control schemes including Lyapunov, PID, LQR, Backstepping, and Sliding Mode. For several of these works he includes analysis of a tethered free flight of the vehicle and presents both positives and negatives from each of these techniques. He concludes that the best technique would be a merge between PID control and Backstepping control, which yields an Integral Backstepping technique widely discussed in non-linear control theory. He presents the stability from a tethered hover in a controlled test environment. He concludes that the stability of the system shows promising results and presents several other techniques on closing an outer-loop for trajectory/waypoint path following, however, not demonstrated in flight. At his conclusion he recommends that a system be built with the ability to carry higher payloads to allow for further integration of sensors and payloads for localization and obstacle avoidance control schemes. Recommendations also include more powerful computing platforms to allow for further data fusion and to handle complex processing efforts, and to increase flight endurance abilities for testing and practical use purposes. Finally, the author recommends that the control scheme be tested further with larger multirotor platforms in uncontrolled environments with disturbances to evaluate the proposed control scheme further.

Chapter 3

Vehicle Requirements & Design

This chapter addresses the mechanical, electrical, and the system architecture design of a fixed pitch multirotor platform to be used throughout the duration of this thesis for modeling and controller development. The selected design will also be constructed and tested with attitude estimation and control systems presented in later chapters. Key design features will also be discussed and how these features meet the requirements of the system, benefited the design for airworthiness, durability of design, and the potential expansion for future applications. The chapter concludes with a developed communications architecture and UGCS terminal for interacting with the aircraft's flight and payload computers.

3.1 Mechanical Design & Hardware Selection

The design of this multirotor platform is governed by efforts and foreseeable uses of this vehicle for future research projects and developments in the laboratory. First, cost is a high concern when considering the design of a vehicle. Materials and electronics had to be purchasable and available for flight testing. In the event of a vehicle crash, the vehicle had to be cost effective so that parts could be purchased and the platform reassembled to continue testing and development. Crashing also brings the requirement in design that the platform had to be able to withstand and be durable against rough landings, rough handling, and light crashes. Currently, the system design in consideration would need to support multiple data feeds from the vehicle. Data feeds could consist of information from the flight computer or a video feed from an on-board camera. This functionality could aid in operator situational

awareness of the vehicles location, attitude, and status in its operating environment.

Payloads and payload support are dominant requirements in unmanned system designs. This means that if an unmanned system can support and operate multiple payload platforms, the more versatile the UAV becomes. This drives the requirements of payload interchangeability and interoperability. Operating payloads drive two main requirements in design considerations for a UAS. First, adding payloads adds vehicle weight, thus the vehicle needs to have additional payload overhead in order to remain under the maximum lifting capabilities of the motors. Also, when switching between payloads it is desirable that the mounting points for the payload be kept generic and open. Requiring this functionality of the design allows for interchangeability and ease of future developments for payload platforms. Summing the above two concepts together yields a generic platform design that supports access and mounting points throughout the vehicle that can structurally support varying payload weights and sizes.

Flight duration is another key component of system design that dictates much of the functionality and success of a vehicle. Flight times will dictate if the UAV can be used in both primary and supportive operations. Lastly, when considering a design, it must not be impractical in size. This platform is expected to be stored, transported, and operated by a single user. This puts constraints on how large the propeller and rotor system can be, as well as the overall dimensioning and size of the vehicle. Part availability will also be considered as driven by size constraints, because if particular pieces of the vehicle are not of common size, their availability will most likely be minimal. Also, weight and size will play large factors in the flight duration, payload availability, and machinability and reproduction of vehicle parts.

There are currently many vehicle platforms out there that may be commercially or privately available as discussed previously in the literature review section. However, each of those platforms is already designed with constraints and ideas of the designers at the time. By designing a custom solution, the architect has the ability to plan for where he/she wants the design to be currently, and forethought's into the future as to where the vehicle may go. Also, the idea for this system is to drive down the cost in parts so that it may be available and obtainable for a personal level, rather than by a larger agency or business. By designing an in-house system, it is capable for the designer to have control over all parts of the design and system integration. This allows for quicker part development and robustness for the platform.

The first part of the process is to consider the platform on which to design the system and control architecture around. Again, a multirotor fixed pitch platform was chosen over a helicopter platform because this limits the amount of mechanically moving parts, creating system stability and ease of repair. The next sub-section discusses platform considerations as well as the design criteria used when deciding on a platform. Details as to how each of the potential platforms could help in attaining the design goals and requirements from above will also be discussed.

3.1.1 Platform Design Consideration & Selection

To date, many new concepts for fixed pitched multirotor platforms have been developed by innovators and companies from both the public and private sectors. Each design includes beneficial factors from cost of materials and parts, control complexity, and ease of manufacturing. Designs considered plausible include such basic platforms of a Quadrotor, TriCopter, HexaCopter, and OctoCopter UAV systems. These basic forms of multirotor vehicles are shown below in Figure 3.1 for clarity.

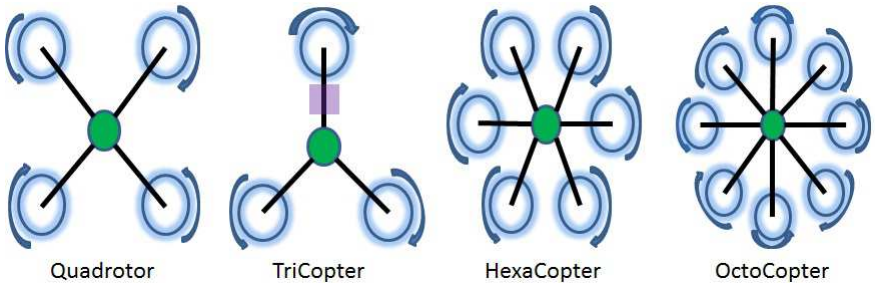


Figure 3.1: This figure highlights typical forms of fixed pitch multirotor UAVs.

Of the above designs, the HexaCopter and OctoCopter platforms are variants of the base modeled design Quadrotor. The change mechanically in each is the addition of a counter-rotating pair of rotors to increase lift capacity and system redundancy. Interestingly, the TriCopter design pictured above is a unique vehicle design because the yawing moment is no longer controlled through the varying torques of the rotors, but rather the mechanical shift of the rear motor through a servo mechanism. This is represented by the purple square in Figure 3.1 on the TriCopter platform. However, the potential designs presented above are merely typical versions of fixed pitch multirotor design platforms. There exists many

forms of variants for the above designs that can decrease yawing moments and mechanical complexity, increase lift potential, and adapt for the potential of large payload platforms that require wide angle viewing areas. Variational design forms that will be considered are shown below in Figure 3.2.

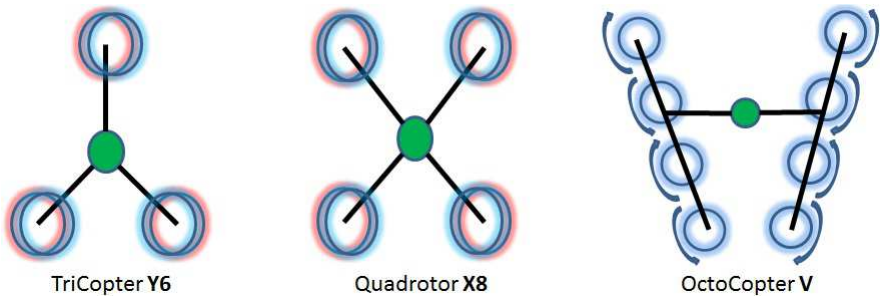


Figure 3.2: This figure shows variational forms of the traditional fixed pitch multirotor vehicles.

The variational forms presented above all present unique mechanical and construction challenges in their design, as well as deviate from typical control basics used in fixed pitch multirotor vehicles. For the TriCopter Y6 and Quadrotor X8 platforms, the red and blue shaded circles represent a pair of stacked counter-rotating rotors. This causes the air to accelerate through two motor systems causing an increase in lift potential while decreasing the required physical area of the vehicle. Also, each motor provides the needed counter-torque to its opposing motor, thus ridding the TriCopter Y6 platform of its mechanical complexity provided in its base form of the TriCopter. Lastly, the OctoCopter V platform provides an interesting design for payload considerations. It can be seen that with its wide support and lift arms clear of the main central area of the vehicle, this platform could be used for wide viewing payloads such as DSLR cameras or LIDAR scanning systems.

The next step of the process was to select a platform in which to implement the theoretical developed control laws on and to physically develop a system for. For this, a basic weighted decision matrix shown in Table 3.1.1 was used to select the best platform when considering some of the design objectives from above. The table weighs the individual components of the design as the author saw fit. The rating scale ranged from -3 to 3 where -3 was worst than the baseline, +3 was better than the baseline, and 0 is equivalent to the baseline model. The baseline model used as a reference for comparison was the Quadrotor platform.

Table 3.1: This table shows the decision matrix that was used in the selection process when considering a design platform.

Various Platforms			Quadrotor	TriCopter	TriCopter (Y6)	HexaCopter	OctoCopter	Quadrotor(X8)	OctoCopter (V)
	Objectives & Design	Weight							
1)	Number of Parts	13%	0	1	-1	-1	-3	-3	-3
2)	Physical Size	5%	0	1	1	-1	-3	-3	-3
3)	Physical Weight	12%	0	1	-1	-1	-3	-3	-3
4)	Power Consumption	5%	0	1	-2	-2	-3	-3	-3
5)	Additional Lift Capability	10%	0	-3	2	2	3	3	3
6)	Payload Adaptation	10%	0	-3	-3	3	3	2	3
7)	Failure Recoverable	30%	0	-3	-3	3	2	2	3
8)	Control Authority	10%	0	-3	-3	2	3	3	3
9)	Mechanical Complexity	5%	0	-3	-2	-1	-1	-3	-2
	Total Weight Score	100%							
	Total Unweighted		0	-11	-12	4	-2	-4	-2
	Weighted Result		0.00	-1.60	-1.70	0.95	0.60	0.50	0.65

The most important design factor in its weighted score was the ability for the vehicle to have failure recovery. This importance weighs highly on the results of the design because the chances of a single motor or ESC combination failure during flight is high when accounting for all of the other potential points of failure in the system. Thus, redundant motors can aid in correcting for pitch, roll, or yawing moments in the event that one motor/ESC combination were to fail. This potential corrective action will be required to keep the vehicle steady enough to provide the ability for the operator to attempt a landing. Also, future goals of the project could consist of an adaptive control scheme that could handle flight failures. This would drive a requirement that the system contain the ability for redundant or powerful lift systems in which the control architecture could leverage to regain control authority over the vehicle. Thus, both forms of the TriCopter and the Quadrotor took large weighted deductions for its lack of system redundancy.

It can be seen from Table 3.1.1 above that vehicle cost, parts, size, and weight drove over 30% of the decision. This was because of the desire to keep the vehicle within the ability to

rebuild the system in the event of a crash. Thus, adding more motors required adding more ESCs; consequently driving the cost up. With additional motors came the added requirement of power consumption and reduction of flight duration. However, the positive trade-off of additional motors included the ability to carry a larger payload platform.

At this point, the decision was made to dismiss the possibilities for a Quadrotor because of its inability to contain system redundancy and the small lifting capabilities of the vehicle when compared to other potential designs. Next, even though the simple and variant forms of the TriCopter provided unique visual aspects and design challenges, the inability again for system redundancy and the required mechanical complexity drove these designs to be undesirable. Lastly, the Quadrotor X8 variant was dismissed for the mechanical complexity of stacking pairs of motors. This left the HexaCopter, OctoCopter, and OctoCopter V platforms for design consideration. Therefore, when considering the additional costs of each motor and ESC pair, this drove the decision towards the development of a HexaCopter platform.

3.1.2 Motor/Propeller/ESC Selection

The difficulty in finding a motor and propeller combination when designing a radio-controlled propulsion system is the lack of documentation present for many of toady's motors on the market. However, even with lift data available, a design cycle is usually difficult because of the numerous unknowns at each stage of the design process. Thus, when trying to narrow down the design, there are always tradeoffs and constraints that need to be in place in order to begin the design process. An example of the process used throughout the design of this UAV system is shown below in Figure 3.3.

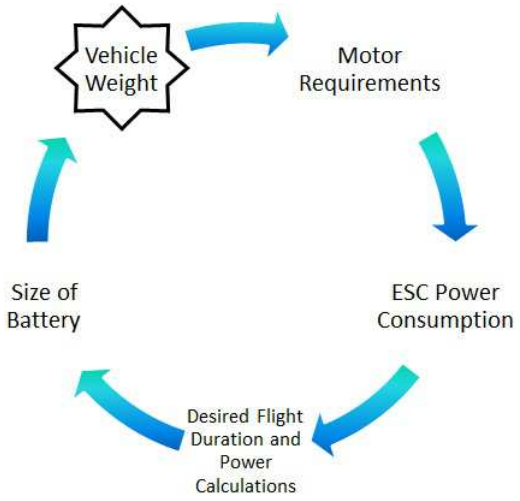


Figure 3.3: This figure details the design cycle that was used to aid in the development of this UAV platform.

It can be seen from the figure above that vehicle weight has been starred. This star represents the unknown in the design process because all other factors of the cycle are typically chosen first. Then with those considerations in mind, a platform is designed around each of the desired subsystems. With the mechanical design of the UAV in place, and vehicle weight produced; the cycle is then repeated to see if the design can be validated and meets the originally imposed constraints/requirements. Examining the propulsion system first allows the architect to impose further constraints later in the design process in which the vehicle will mechanically meet.

Today, propulsion systems are available from all around the world and vary in quality and capabilities. Decisions for the propulsion system were narrowed down by commercial availability and wide-spread use. Thus, three specific motors were selected for consideration along with an open potential for other motors from an R/C distributor. Table 3.2 below describes the weighting decision matrix used when selecting the motor for this vehicle platform.

Table 3.2: The table below details a decision matrix used in selecting a motor for the HexaCopter Platform.

Various Platforms			DIYDrones (880kV)	Values	DIYDrones (850kV)	Values	Exceed Optima 450	Values	Hobbyking Various	Values
	Objectives & Design	Weight								
1)	Availability	10%	0	Al	0	Al	-1	U	-3	S
2)	Cost	20%	0		1		2		3	
3)	Shipping Location	5%	0	USA	0	USA	0	USA	-3	China
4)	Physical Weight	15%	0	72g	1	62g	-3	N/A	0	Light
5)	Power Consumption (3S)	10%	0	20A	1	12.2A	-3	N/A	-3	???
6)	Lift Data	15%	0	Av	0	Av	-3	N/A	-3	???
7)	Lift Capacity	25%	0		-3		1		0	
	Total Weight Score	100%								
	Total Unweighted		0		0		-7		-9	
	Weighted Result		0.00		-0.3		-0.65		-0.6	

1) Al-> Always U->Usually S-> Sometimes 5) 3S refers to a 3 cell LiPo with 11.1V nominal
6) Av->Available N/A-> Not Available
The ??? represent mixed results based on the possible motor selection. While N/A in any category represents Not Available

Major considerations of the motor selection as seen from the table above were cost and availability of published lift data. Also, motor lift data availability is important because this allows for the expected performance characteristics to be planned for in advance and the design to continue. Another factor for motor selection included the availability of the motor. This is important because in the event of a crash or needed repairs, replacements would be needed to be available for purchase to continue testing and operation. The shipping location weighs in for the same basic principles. Power consumption is important as this will dictate the size of ESC system required. Typically the larger current capable ESCs directly effect commercial cost. The larger the power consumption from a Motor/ESC assembly also imposes requirements that the system contain higher gauge wire and thicker traces to properly distribute the current. Thus, with all of the design considerations chosen and appropriately weighted, the choice concluded that the two motors in consideration would

be the motor sets from DIYDrones. These have been commercially sold to many hobbyists and went through numerous hours of beta testing for their abilities and longevity of use. Therefore, the decision was made to design for the larger 880kV motors, while accommodating future potential designs for the lower rated 850kV motors. Also, it should be noted that the planned vehicle weight of 5~7 pounds achieves the desired goal of the system operating at or under 50% of the motor's lifting capacity. This allows for the most efficiency in the system and leaves a sizable portion of bandwidth in the motors for control authority.

Lastly, 30A ESCs were chosen as this current rating satisfies both motor systems in consideration and 11"$x7^o$ propellers will be used to satisfy the propulsion capabilities and requirements of either motor. However, planning will take place assuming the recommended 12" propeller size from the motor specification sheet. The smaller propeller size was chosen because of the wider availability of the necessary rotating and counter-rotating pairs of propellers. The next sections will detail how the motors will be mounted to the platform, as well as the central assembly to the vehicle.

3.1.3 Arm & Central Platform Design

An important design aspect of any UAV system is the frame, as this will handle both the static and dynamic loads imposed on itself and from the environment in which it is to operate. For a HexaCopter platform, the arm structures that hold the motors connected by a central hub are of the highest importance of the system. First, it should be noted that the arm support assembly has to be rigidly mounted as to not introduce large movements and fluctuations in the frame itself. It must also be able to take the moment and force applied by the motors and provide a rigid mount in the main structure to lift the body. Given these lightweight yet strong material constraints, commercial availability, and durability, aluminum was chosen as the material for the arm structure.

A 1/2" extruded aluminum square was chosen so that cabling could be run through the inner core of the arm to protect the cabling from coming loose during flight and incurring a prop strike. The next step was to choose a design constraint to estimate the length required of the arm. As shown in Figure 3.4 below, each arm needed to be distant from the center of the body for each propeller to clear each neighboring prop and to clear the central hub that would house all of the flight and power electronics.

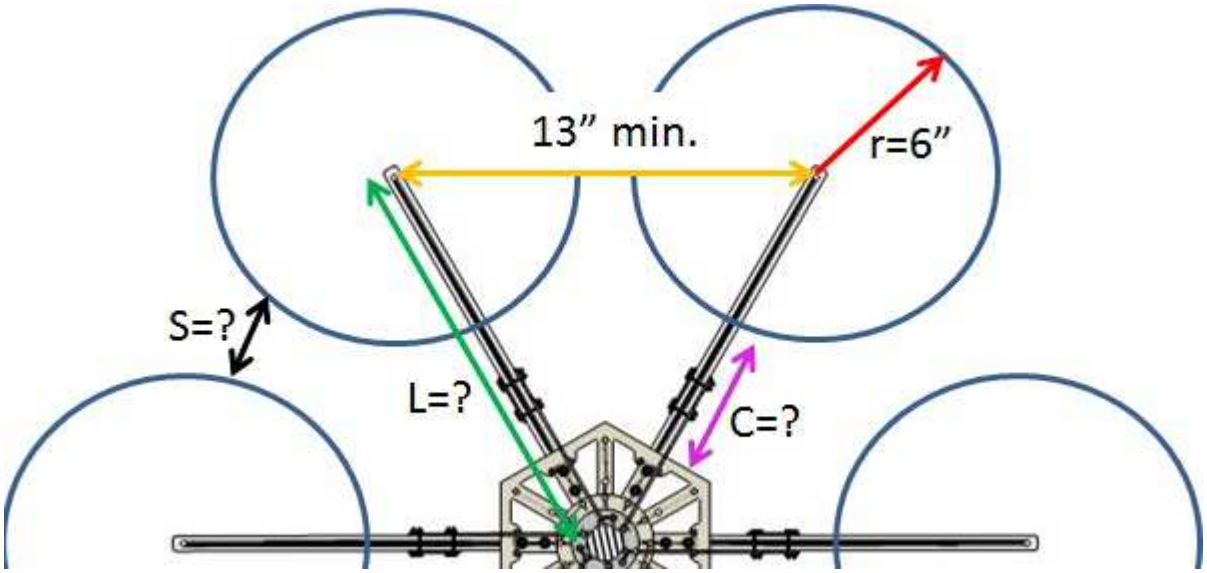


Figure 3.4: Dimensioning constraints in consideration for motor mount arm length.

As shown in Figure 3.4 above, there are several open design parameters such as S, L, and C. Each of these variables represent the distances for prop-to-prop clearance, arm length, and prop to body clearance respectively. First, there are several maximum constraints that can be placed upon the vehicle to help the design process. To make the system portable, it is desired that the system be able to fit through typical 36" doorways and narrower hallways without having to be turned on its side. Also, since this vehicle is to be operated by one individual, we desire to keep the overall size of the vehicle small. Therefore, a constraint can be placed on L to be less than 18 inches. Also, longer wires running from the ESC position to the motor will cause voltage drops because of the increasing distance of the wire and greater inductance loops caused because of the switching frequency of the ESC signal.

Another interesting insight is the relation between the length of the arms and the control authority for the pitch and roll controllers. It can be thought of that the command signals to the ESCs which control the rotor RPM and accordingly affect the moments and forces about the vehicle, have a finite discretized bandwidth. Therefore, as one increases the length of the arm, a flight controller can effectively create larger moments and forces about the axis of the vehicle, however, the control efforts will be coarse because of the discretized resolution of the ESC. However, if the arm length is short, one will not be able to create as much moment and force about the axis of the vehicle, however, will gain the advantage of more fine and precise control. From experience, the author has seen that to create the flight envelope desired from

this vehicle, it will not be necessary to consider large moments, but rather desire precise control and therefore tighten the arms to the center as possible.

Examining Figure 3.4, it can be seen that the 13 inch minimum placed between each pair of motors would allow 1 inch prop clearance. Adding an extra half an inch would allow for additional clearance room and thus constraining S to be 1.5 inches. Since each pair of arms will form two legs of an equilateral triangle, the arm length could be chosen to be 13.5 based on the previously decided information. The body had been sized to be approximately 12 inches in diameter in order to house all of the necessary electronics; therefore, C would be defined to be 1.5 inches. Since the motor arms do not have to mount to the entire central hub, and the motor wires would need to be able to be manipulated to ensure proper connections, the arm length L was decided to be 11.25 inches. Two mounting holes were provided at one end of the motor arm to allow for the the arm to mount to the main central hub of the vehicle while another single hole was provided at the outermost portion of the arm to allow for the motor mount configuration. Design for the motor mount will be discussed in the next section.

Since the arms were made of aluminum, these were the most controllable weight components of the system. Therefore, it was desired to minimize the weight of the arms and determine the amount of material that could be removed from the arm effectively and efficiently. Also, removing material from the arms would provide access holes throughout the entirety of the arm structure and would aid in feeding motor wires and other peripheral cabling through the arm assembly. Thus, a truss like counter weave design was formed on opposing sides of the arm. Finite Element Analysis was performed to determine the amount of material that could be removed before significant increases in loading deflection and stress would be observed. Figure 3.5 below shows the deflection of the arm assembly. The left image highlights the deflection of the beam with material removal, while the right image shows the deflection of the beam without material removal. A summary of the constraints and loads is below the image, as well as tabulated results of the tests in Table 3.3.

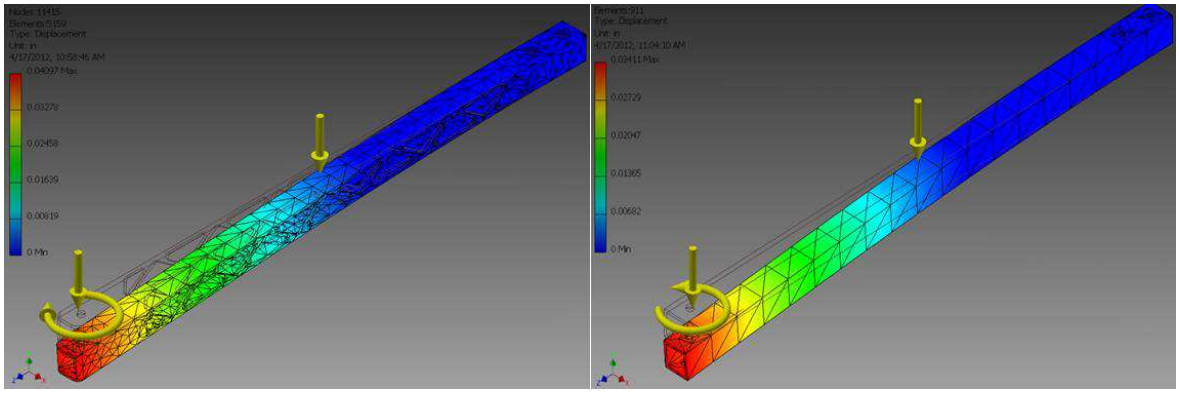


Figure 3.5: FEA analysis of the arm support beam. LEFT: FEA of the arm support beam with material removal. RIGHT: FEA of the arm support beam without material removal.

The FEA analysis had consisted of applying the same conditions to each sample beam set. Considerations for modeling included a fixed boundary constraint at the far end of the beam representing the central hub that would be holding the arm in place. A force for the lifting load of each motor and a torque induced by the motor was added (These values were determined through a test later described in this work). Lastly, a distributed load representing the gravity vector was placed over the beam. The tests had shown that material could be effectively removed for ease of assembly and reducing weight while sacrificing little integrity of the beam for deflection, stress, and factor of safety. Table 3.3 below details the results from the FEA modeling.

Table 3.3: Summary of FEA analysis on arm assembly.

Design	# of Nodes	Deflection	Von Mises Stresses	FOS	Mass
Solid	1819	.03411 in	28.3497 ksi	15 ul	43.893g
Truss	5159	.04097 in	27.191 ksi	15 ul	32.337g

Before assuming the FEA analysis presented by Autodesk is correct, several hand calculations were to be performed on the solid design to determine if the beam had been modeled and simulated correctly. First, the moment of inertia for a beam can be calculated as

$$I_X = I_Y = \left(\frac{bh^3}{12}\right)_{outer} - \left(\frac{bh^3}{12}\right)_{inner} = \left(\frac{(0.5)^4}{12}\right)_{outer} - \left(\frac{(0.4)^4}{12}\right)_{inner} = 0.0031\, in^4 \quad (3.1)$$

Substituting the moment of inertia for the beam and other parameters into Equation 3.2

below yields an approximate max deflection of the solid beam.

$$\delta_{max} = \frac{Pl^3}{3EI} = \frac{(3lb)(10.25in)^3}{3(10.4E6^{lbs/in^2})(0.0031in^4)} \approx 0.0334in\ estimated\ deflection \tag{3.2}$$

It can be seen that from Equation 3.2 above, the FEA model presented from Autodesk is accurate. Therefore, the final design was chosen based on material removal, access for wire feeds, wire protection, and for strength and durability. The chosen design is shown below in Figure 3.6.

Figure 3.6: This figure shows a CAD rendering for each support arm of the HexaCopter.

Next, a central platform hub would need to be designed to connect and hold each of the six arms in place. It was decided that two platforms would be used and compressed around the arm assembly. This would allow the hub size to grow if needed without affecting the required arm length. Also, this would allow for a sense of platform stacking to occur to allow for multiple layers to be created and connected providing additional room and rigid mounting surfaces. The central hub assembly was chosen to be constructed and machined out of carbon fiber for its high strength to weight ratio. Design considerations included additional mounting holes strategically placed to accommodate future landing gear designs and additional stacks, and a large central hole allowing both signal and power wires to feed throughout the entirety of the vehicle. Figure 3.7 below shows the design of the central hub with the joining arm assembly for the vehicle. Each of the six arms is constrained in the assembly through aluminum machine screws, while two additional nylon machine screws were placed tightly on the outside of the beam to prevent any horizontal movement.

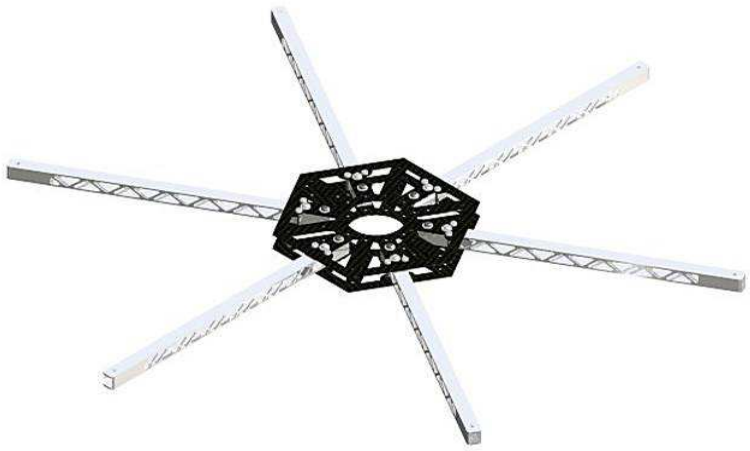


Figure 3.7: CAD rendering of the central hub assembly with the six motor support arms attached.

3.1.4 Motor Mount Assembly

Next, a motor mount assembly needed to be designed and fabricated that would affix one motor to each of the six arms. Design features had to include a safe and effective way to ensure that the motor assembly would not move during flight operations. Typically, hobby motors come with an aluminum "+" shaped mount and associated hardware. However, this would not work as the configuration of the arm assembly from above is a $\frac{1}{2}$inch compared to that of the $1\frac{1}{10}$inch diameter of the motor. Thus, custom motor mounts were designed to hold the motor in place and reduce vibrations induced from the motor/propeller assembly. The design rendering and assembly is shown to the left of Figure 3.8 below. The motor assembly seats atop ribbed grommets which isolate the motor from the frame. Machine screws secure the motor assembly to the top motor mount through a spacer preventing the grommet from being overly compressed and is secured by lock-nuts at the base. A single machine screw at the rear holds the upper motor mount, passes through the arm, and then secures the base of the motor assembly with another lock-nut. Two front machine screws are added around a flared portion of the mount to allow for the entire assembly to compress around the motor arm. Also, the top and bottom motor mount assemblies feature a dado-groove so that the mount seats over and around the arm assembly ensuring that torques caused by the motor do not cause the motor to pivot. Lastly, a 5mm hole was left in the bottom assembly for an LED status light. This status LED can be seen in many forms of the commercial

UAV systems. The material chosen was polycarbonate for its low cost and rigidity. Also, polycarbonate would aide in the dispersal of the LED indicator status. Final assembly is shown to the right of Figure 3.8 below.

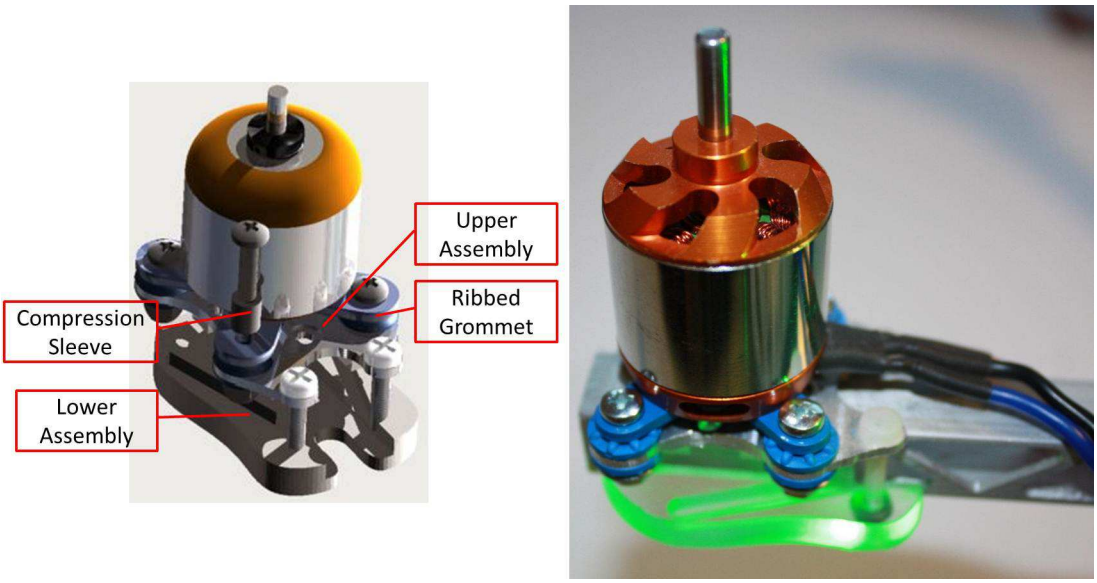


Figure 3.8: LEFT: CAD rendering of the motor mount assembly. RIGHT: Motor mount assembly affixed to a single arm of the HexaCopter with status LED indicator lit.

The LED indicators at the end of the motor and arm assembly serve two primary functions. The first is that they assist in defining orientation of the vehicle for a ground operator as this vehicle is symmetric about its yaw axis. Two green LEDs are contained in the front of the aircraft while two red LEDs are at the rear of the aircraft in an effort to aid in the operator's awareness of vehicle orientation. Also, each LED serves as an indication about specific functions on board the vehicle. Table 3.4 below gives a simple description into what each of the LEDs mean during operation. This is useful information for the operator to quickly evaluate the status of the vehicle before/during/after flight operations.

Table 3.4: Description of Motor LED Indicators

LED Position #	Color	Purpose
1	Green	Battery Voltage
2	Green	Regulator Voltage
3	White	Radio Signal
4	Red	Motors On/Off
5	Red	GCS Comms
6	Blue	GPS Lock

3.1.5 Landing Gear & Undercarriage

It is important that the undercarriage and landing gear assembly be able to support varying loads that may be exerted from the frame. Also, design considerations for the landing and undercarriage assembly must account for its ability to remain generic so that future iterations or design modifications could easily be mounted to existing hardware points. First, the design was based around a landing skid design similar to that of a helicopter. R/C helicopter skid braces were used with tapped holes added to allow for set screws from the frame to hold the skids in place. Next, it was decided that the batteries would be housed in the undercarriage and be supported by the skids. This gave a height and clearance constraint to fix the battery tray at a minimum of two inches below the central hub assembly.Constraining the battery tray as well onto the undercarriage system itself would help further distribute weight.

In design, since this undercarriage and landing skid configuration will be taking most of the loads for takeoff and landing, it was desired that the material be lightweight. Thus, ABS plastic was chosen for the landing gear assembly as it has high tensile strength qualities that mesh well with its machinability and elastic properties, making it desirable to use for a landing skid framework. Lastly, it was desired that the majority load of the structure not be distributed entirely through the standoff supports that also held the undercarriage tray in place. Thus, a unique keyed feature was designed on top of the landing skid assembly to allow the structure to slide in between the two main layers that support the arms and constrain itself between the 60 degree angles formed by the arms. This then locks the landing skid configuration in place and is held further by one final machine screw to ensure that the structure cannot slide away from its assembly. Figure 3.9 below highlights the design of the landing skid and battery tray assembly.

Figure 3.9: CAD rendering of the undercarriage and landing gear assembly of the HexaCopter platform.

3.1.6 Roll Cage & Canopy Assembly

The central power and processing electronics are the most costly and vital pieces of equipment housed on-board a UAV system. Therefore, it is necessary to protect vital components on board the vehicle to ensure their survival and useful longevity. To protect the central processing and power hub, a roll cage was designed so that in the event that the UAV were to crash or roll over during or after flight, everything housed in the center would be protected. Each aluminum arm mounted to the central hub was used as a support structure for each roll cage arch. The six arches were then met at the top by two rings keyed appropriately to accept the key position of the arch. The bottom of the arch assemblies were chosen to again be compressed by polycarbonate around each arm through four nylon machine screws. Figure 3.10 below on the left details a side profile of a single arch, while the right image details the rollover sub-assembly for the vehicle.

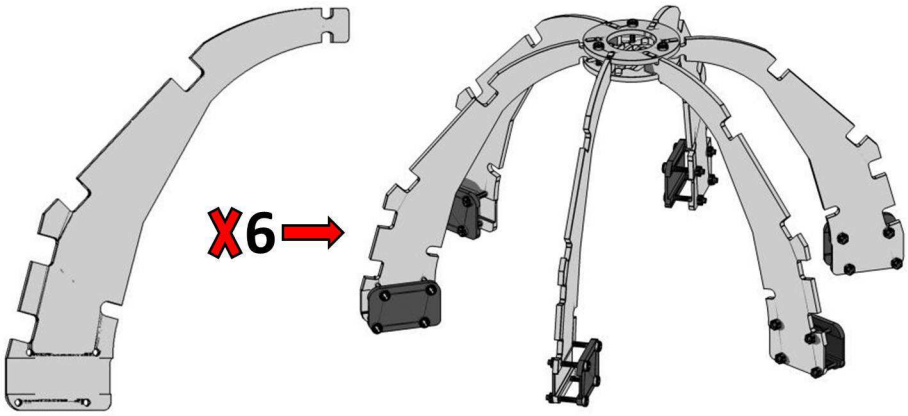


Figure 3.10: This figure shows how six of the individual rollover arms on the left form a complete rollover cage on the right.

3.2 Required Electrical Components

The next several sections will go into an overview showcasing the custom electrical component designs that were required from this UAV system. Several Eagle schematics are provided for further reference in Appendix A, while others are available upon request.

3.2.1 R/C Safety MUX Development

An important function in an autopilot system is the ability for the operator to regain control authority over the flight commands of the aircraft. Reasons for this functionality include unauthorized aircraft operating within the airspace of the test system, unforeseen/unwanted environmental disturbances, aircraft drifting towards inanimate objects and collision avoidance is required, or the autonomous controller may fail or operate incorrectly. An R/C MUX figuratively operates by allowing a command signal on a switch control gates between multiple sources of information and the outputs which will cause the aircraft to actuate surfaces or control thrusters. The governing architecture for this R/C MUX was a customized design to allow for extensive testing while providing safety for both the vehicle and operator. A generic system level architecture for the MUX is presented in Figure 3.11 below.

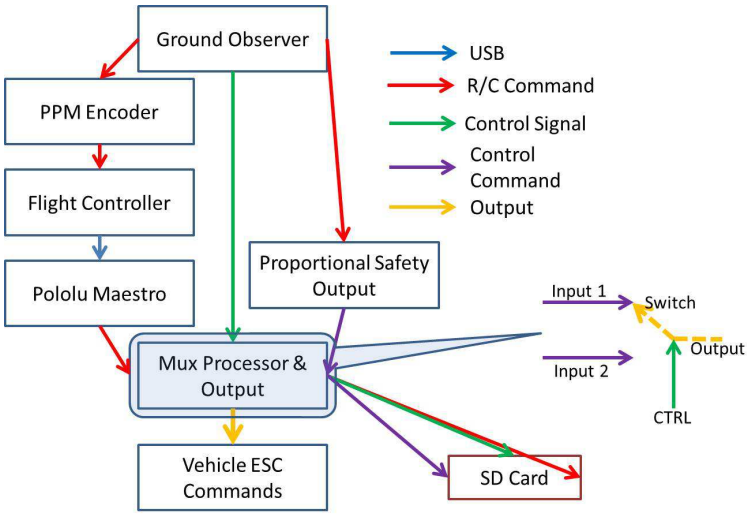


Figure 3.11: A system level view of information flow on the R/C MUX.

The ground observer or pilot generates five commands with a TX that govern actions of the flight control system aboard the aircraft. The R/C commands consist of roll, pitch, yaw, and throttle signals as well as a control signal operated from a switch on the TX. The RX on the aircraft then receives the ground commands from the remote and splits the signal to the flight controller as well as a proportional safety controller. Each controller generates an output command based on the given input signal as well as any extra peripheral information available. This flows to an IC MUX chip and based on the control signal, a decision is made to allow one of the control outputs through to the ESCs. Also, on-board is a micro SD card that logs the control signal command and the proportional safety output command that is time stamped into a log relative to boot up time of the chip. The mux system developed for this thesis had also been implemented in several other flight control systems from the lab to ensure safety and proper operation of unmanned aircraft. The testing and implementation in other groups was necessary to ensure successful operation and endurance of this integral piece of hardware. Algorithm 3.1 documents a general pseudo code of how the processor on the mux works and the logic as to how it determines when to switch between signals. Appendix 1 contains a schematic of V2.0 of the created MUX board.

Algorithm 3.1 Processor R/C MUX Pseudo-code.

```
1: Receive control and R/C commands from ground operator
2: Parse and send signal to flight controller and onboard linear controller for processing
3: Process linear interpolations based on flight commands
4: Send linear commands to one side of mux channel
5: Interpret control signal
6:    if (Control Signal > Flight Control Threshold) && (Event Trouble==TRUE)
7:         ∀ Linear Control Signals do
8:             if(Flight Control(i) ∈ Linear Flight Control Command(i) +/- Tolerance)
9:                Event Trouble=FALSE
10:            end if
11:             else if(Flight Control(i) ∉ Linear Flight Control Command(i) +/- Tolerance)
12:                    Event Trouble=TRUE
13:                 break
14:            end if
15:            else Event Trouble=TRUE
16:                    break
17:         end for all
18:    end if
19:    if (Control Signal > Flight Control Threshold) && (Event Trouble==FALSE)
20:        hold mux channel high to enable flight control output
21:   end if
22:     if (Control Signal< Flight Control Threshold)||(Event Trouble==TRUE)
23:        hold mux channel low to enable linear flight control output
24:         Event Trouble=TRUE
25:   end if
26:  Write linear output commands, Control Signal, Event Trouble value to SD card
```

Figure 3.12 below shows the final PCB design of the MUX selector and pinout/connector board developed for this work, along with the connecting radio module. Not featured in the image is the SD card expansion board.

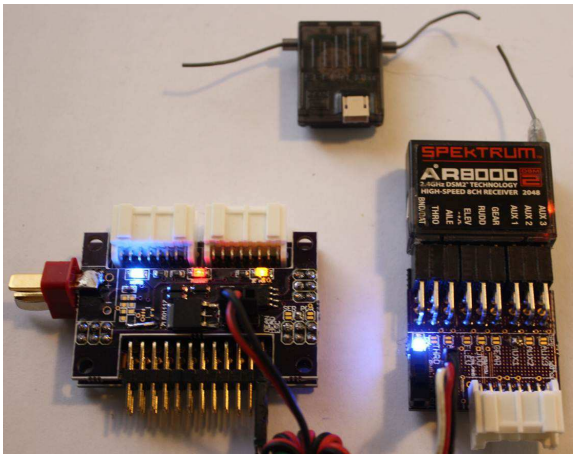


Figure 3.12: Left: V2.0 of an 12-channel R/C MUX system developed for UAV operations. Upper and lower right: Spektrum 8 channel receiver and satellite with RX docking board.

3.2.2 Power Systems

Each power system in consideration efficiently and effectively delivers power to the necessary subsystems for operation. There are three main power distribution and regulation boards in the system that deliver either battery level voltage directly to the subsystem, or a series of voltage regulators that reduce the battery level voltage down to the desired operating voltage of the subsystem. Figure 3.13 below details the proposed electrical system architecture incorporating the three power distribution systems above. Further details of how each electrical subsystem is contained in the following sections below with schematics documented in the appropriate Appendix sections of this work.

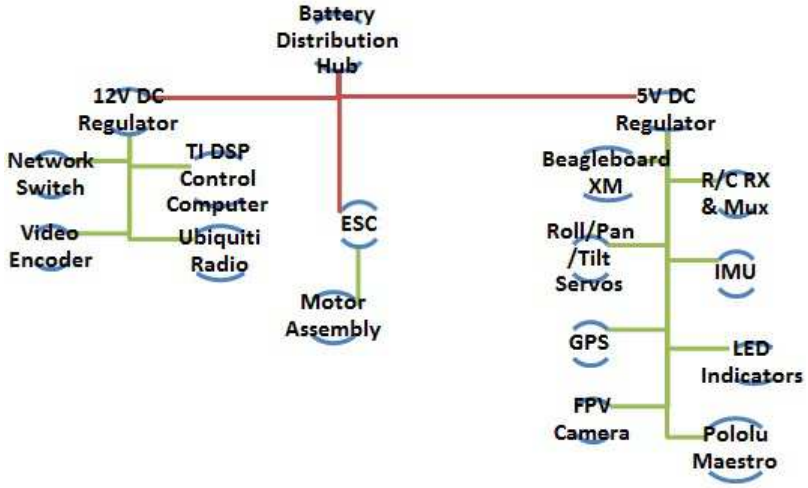


Figure 3.13: Proposed electrical system architecture implemented on board the UAV platform.

3.2.2.1 Primary Power Distribution Hub

First, the primary electrical source for the system needed to be distributed throughout the entirety of the vehicle while providing terminals for potential future subsystems. A deca-sided distribution hub, as shown in Figure 3.14, contains ten generic output connectors that subsidiary systems could use to gain access to the primary power source. It can be seen from Figure 3.14 that six terminals were going to be used to power each of the ESC/Motor assemblies, leaving four generic output terminals for remaining peripherals. Next, power needed to be regulated and conditioned from the input power source to the required 12VDC and 5VDC for the remaining subsystems.

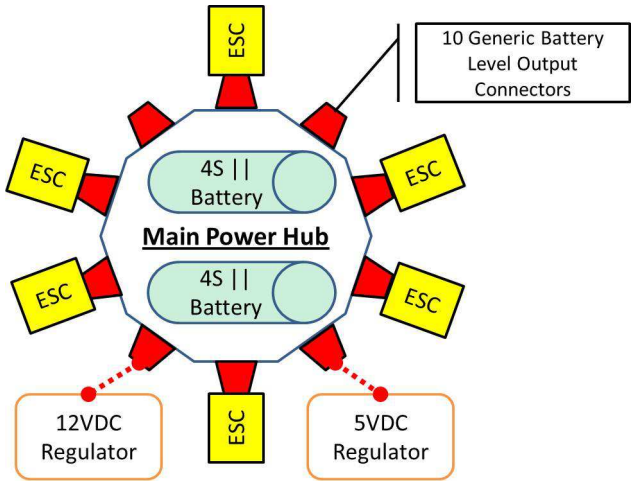


Figure 3.14: This graphic shows the central hub where battery power is distributed to generic connector terminals.

3.2.2.2 Accessory Power Regulation

Two primary voltages were required from system peripherals as shown from Figure 3.13 above. Therefore, custom circuit designs were developed for their efficiency and the operators understanding of the internal circuitry on-board the vehicle. The 12VDC regulated power from the system is integral for the flight processing computer on-board the vehicle. Therefore, it is desired that 12V be regulated without concern to what the input voltage to the regulator may be. Thus, the 12VDC system was designed around the principles of a buck-boost regulator. The buck-boost regulator makes for an elegant and reliable solution when a specific output voltage is required for a wide unknown input range. A further detailed schematic can be found in Appendix 1 while Figure 3.15 below shows the assembled PCB.

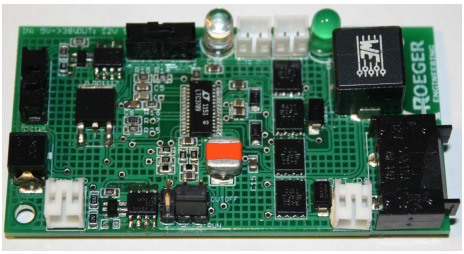


Figure 3.15: V1.0 of 12VDC buck-boost regulator designed for system peripherals on the HexaCopter platform.

Regulating 5VDC from an unknown but consistently higher voltage source is easily done through a buck converter. Many hobbyists would use the internal BEC provided with many ESCs, however, it is possible to overdraw such BECs and cause a burnout on the ESC powering the motor. The power systems relying on 5VDC will get their power from another custom PCB design as shown in Figure 3.16 below. A further detailed schematic can be found in Appendix 1.

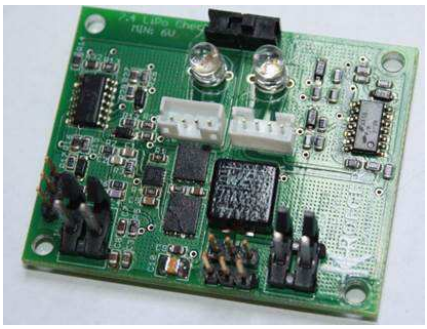


Figure 3.16: V1.0 of 5VDC buck regulator designed for system peripherals on the HexaCopter platform.

3.2.2.3 Primary Power Source

The motor specifications require an operating voltage range of 9-17.2 VDC, which implies a 3-4S LiPo battery. The voltage rating on a motor is produced by the manufacturer and is based on the coil windings inside a brushless motor which dictate the motor's RPM rating and its current handling capabilities. A rating of 880kV implies 880 revolutions per applied volt. This implies that with a 4S battery (nominal voltage of 14.8V), the motors could achieve approximately 13,000 RPM with no loading. Sizing the batteries from this point

becomes a constant trade between expenses, charge time, total vehicle weight, and flight endurance. The goal is to maximize flight endurance within reasonable performance specs of the motors. A design characteristic as stated at the beginning of the mechanical design of the vehicle was to achieve a vehicle hover weight while using approximately 50% of the motor's capability.

First, the estimated vehicle weight based on the CAD model is approximately 4.5 pounds. Based on the performance results from testing (test described in Section 5.4), a weight of 4.5 pounds will cause each motor to consume approximately 32.5 watts of power. A 4S battery's nominal voltage of 14.8 volts would yield a current consumption of 2.2 amperes per motor, totaling at approximately 13.2 amperes for the vehicle. With this amount of power consumption, a 3300mAh battery could ideally operate the vehicle for approximately 15 minutes. However, this flight time does not include the weight of the battery into its calculation. The process here refers back to the process as discussed from Figure 3.3 above where additional batteries are added in parallel to see how the flight time increases at the cost of carrying additional weight. Figure 3.17 below highlights the predicted flight time per additional battery on-board the vehicle.

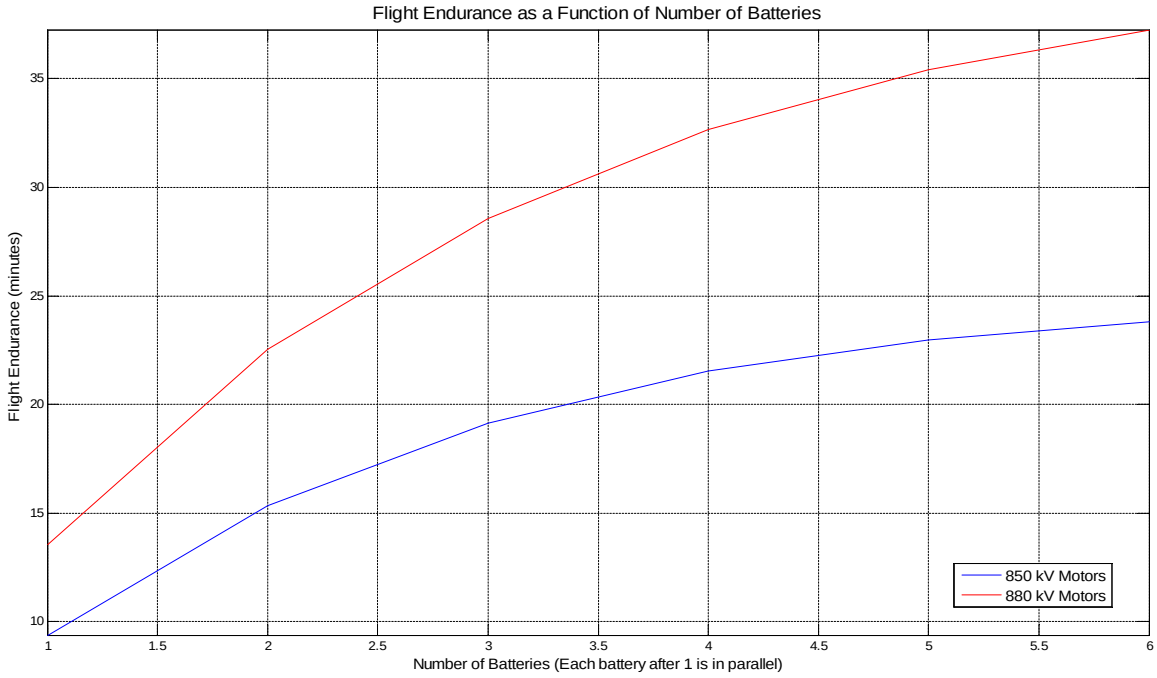


Figure 3.17: This figure details the limited return of flight time through the addition of batteries in parallel.

It can be seen from Figure 3.17 above that there are marginal returns for each additional battery because of the implied additional vehicle weight. It can also be seen that the derivative of the slope is highest through the addition of 1 battery in parallel. An advantage of a minimum of 2 batteries in parallel is this allows for power redundancy in the event that one battery should fail during flight. Therefore, after evaluating the improvement of flight time, cost per battery, and redundancy of power systems; two 3300mAh batteries were chosen to be run in parallel for the power system of this vehicle. This would provide a flight endurance of approximately 23 minutes in a hover condition. An important test here is to recognize that the calculations of the figure above were done with tests using the recommended propellers for each motor. The difference seen is logical as if we consider the smaller motor a baseline performance, we have increased the number of revolutions and blade size when testing with the larger motor and have not increased the load on the motor. Thus, the difference in performance results between the two motors. Another important examination as shown in Figure 3.18 below, is how additional payload weight will effect flight endurance.

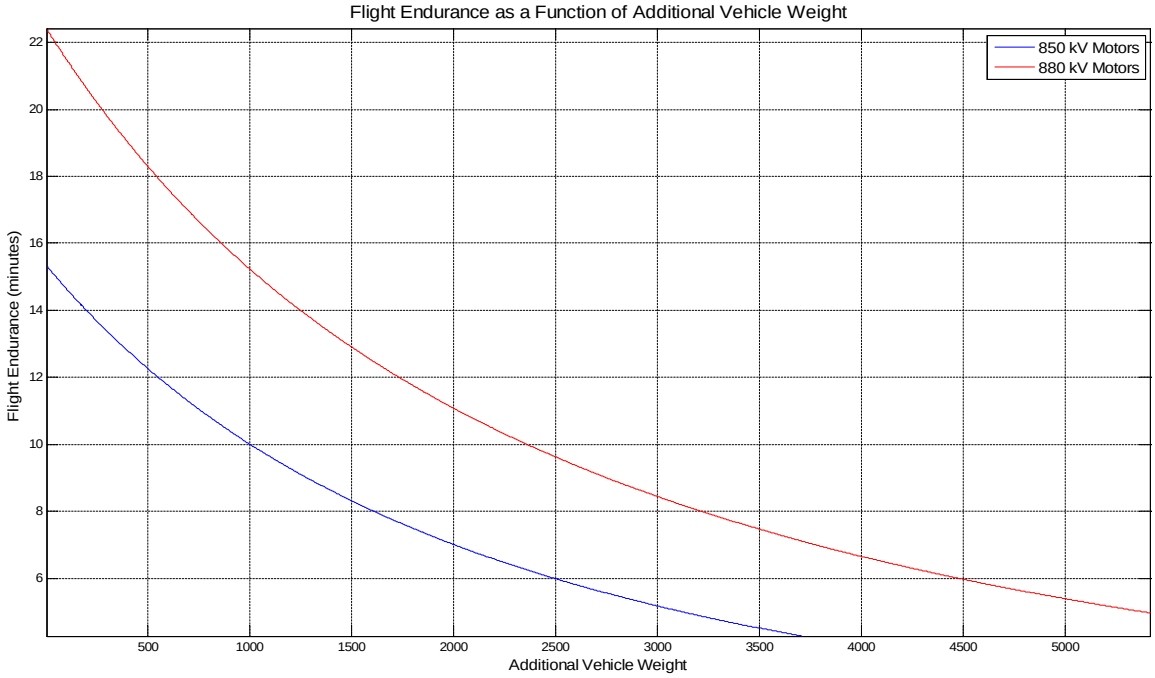


Figure 3.18: Analysis plot of flight duration as a function of additional payload mass when comparing two different KV rated motors.

3.3 Peripherals: Data Stream & Video Feed

Situational awareness for an operator is important when operating an unmanned vehicle, therefore, considerations for communications of both data streaming and video feeds would be required. A single 5.8GHz radio system, five port network switch, and video encoder were acquired and integrated for the task. The next three sections will briefly discuss the integration of a roll/pan/tilt camera system and data transmission system implemented with the UAV.

3.3.1 Roll/Pan/Tilt Camera Gimbal & Mounting

It was deemed necessary that visual feedback be present in the event of beyond LOS operations and when considering the future developments of the aircraft. To improve the operator's situational awareness, a camera system was designed for mounting on the frame of the HexaCopter platform.

A roll/pan/tilt camera assembly was designed and mounted to the two front arm assemblies allowing for the widest unobstructed viewing area. The camera affixed to the mount can then be operated through the UGCS with communications handled onboard the vehicle through the Beagleboard XM SBC. The assembly allows for $\pm 15^o$ in roll, $\pm 45^o$when panning, and $0^o - 90^o$ while tilting. Figure 3.19 to the right shows a rendered sketch of the roll/pan/tilt camera assembly.

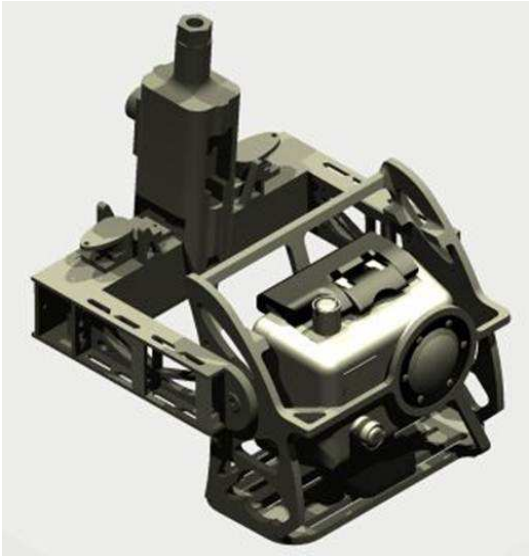


Figure 3.19: CAD rendering of the Roll/Pan/Tilt camera assembly used to manipulate the on-board camera.

An important note when selecting a camera for UAV operations is that a majority of low-cost camera systems, including the one depicted, contain an analog video out. Therefore, a video encoder is required to convert the analog video stream to a digital stream with an associated IP address in order to relay

the video to the operator on the ground. There exists analog video transmitters that could accomplish the task, however, this would not take advantage of the full bandwidth capable from the network based radios. By providing a networked video stream, the images can be accessed by multiple programs in future efforts to accommodate flight trajectory generation for both avoidance and tracking maneuvers, 3D reconstruction through structure from motion, and various other vision based technologies to further aid in operator situational awareness.

3.3.2 Network Radio System

Figure 3.20: This figure shows the 5.8GHz Ubiquiti Network Radios used for this work. Left: Ubiquiti Rocket used for the ground side communications. Right: Ubiquiti Bullet used for the airside communications.

Data streaming is a valuable peripheral when testing a new software system. This allows a developer to see what the CPU is thinking, how developed algorithms are behaving, and decisions being generated from both the flight and accessory computers. A network radio system also offers a remote accessible environment to aid in an embedded computer solution. Important flight details that are directed to the operator contain information such as vehicle location, vehicle attitude, self monitoring, and power level system checks. This data is invaluable for the operator to understand the conditions on-board the vehicle, and further support situational awareness for a user. Therefore, a data transmission system was devised with the implementation of a Ubiquiti 5.8GHz radio system as shown in Figure 3.20 to the above left.

3.4 Final UAV Design

The additional communication architecture and vision system as discussed from above were added to the HexaCopter platform at a cost of 1.75 additional pounds which reduces the

initial estimated flight time from 23 minutes to 17 minutes. The final mechanical design of the HexaCopter platform (including peripheral mechanical hardware but excluding electrical components and connectors)had consisted of approximately 600 parts and is shown in the CAD rendering to the left of Figure 3.21 below. The final developed platform including all necessary flight electronics and peripherals is shown at the right of Figure 3.21 below. Appendix B contains hardware and software schematics that further detail the low-level implementations of power distribution and communications of the inner systems aboard the HexaCopter.

Figure 3.21: LEFT: Final CAD rendering of the developed HexaCopter UAV platform. RIGHT: Final design of HexaCopter platform including all necessary flight electronics and peripherals.

3.5 Universal Ground Control Station (UGCS)

Supporting the data streaming architecture from above is a UGCS which allows the operator to interact with the vehicle through various flight and payload commands. A large issue in developing a communication architecture between vehicle and operator is the interoperability between varying aircraft and ground terminals. Therefore, examining the community for an open communication architecture, this author chose to support MAVLINK [49]. MAVLINK is currently being implemented through various commercial autopilot solutions and universities. The advantage MAVLINK provides is that it is an open-source implementation that

standardizes message sets for common messages of interest when operating a UAV. Therefore, developing both a UAV and a UGCS that supports this protocol is beneficial in that the interoperability and usefulness for both systems can grow as the system develops.

The architecture design of a UGCS needs to consider holistically what subsystems need access to what pieces of information and more importantly how to effectively deliver the required information in a timely manner. Examining the potential future operations of the project, a distributed network of UDP and TCP network calls was created for passing and handling of information between programs. The advantage of this network based architecture program development is that this allows for future programs to gain access to flight data, without interrupting primary processes that have been developed, tested, and integrated. It should be noted that the internal software development of the UGCS (excluding the TCP Groundstation architecture to support uplink) occupies 5 main programs with 45 subprograms and 52 different MAVLINK protocol functions. Figure 3.22 below overviews a high level SW/HW architecture for several of the primary components within the UGCS. Appendix A contains detailed Eagle schematics for the internal PCB boards developed for the UGCS. Appendix B contains communication and power distribution architecture schematics for the UGCS.

SW in the following sections refers to software stored on the internal computer of the UGCS. HW in the following sections refers to hardware/microprocessors attached to the associated device. It should be noted that each microprocessor contains its own software onboard to correctly function and perform required actions.

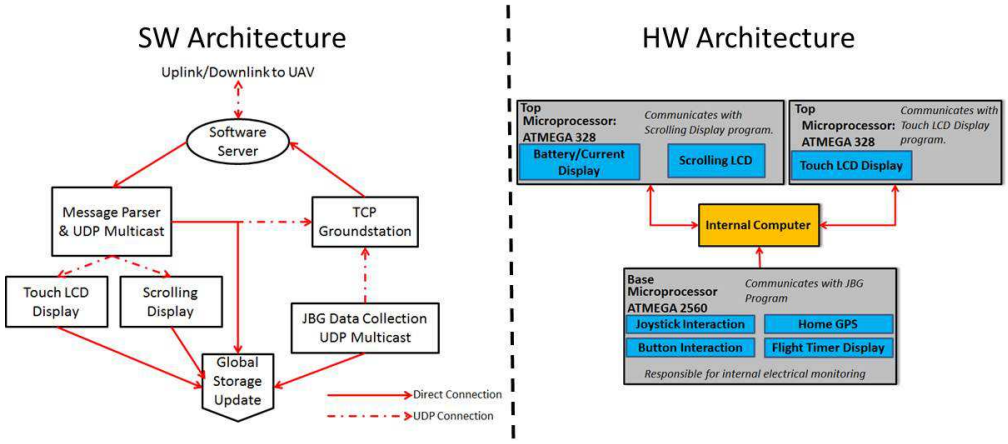


Figure 3.22: This figure presents a high-level architecture of the SW/HW contained within the UGCS.

1. *Software Server:*The purpose of this program is to establish and maintain communications with the UAV system. Information contained in the downlink is first validated through a checksum procedure and if correct, sent to the message parser for organization and distribution among subprograms. Data packets queued for uplink are padded with a start byte, counter, and closed with a two byte checksum.

2. *Message Parser and Distributor:* This program handles the distribution of data to/from subsystems. Data packets passed from the downlink of the software server are organized into their respective message types (attitude, heading, GPS, etc) and updates its respective data cluster within a global storage program. In parallel, the message is then distributed through its coordinating UDP multicast socket for other subprograms usage.

3. *Touch LCD SW/HW:* The purpose of the touch LCD is to provide a user interaction layer between the operator and internal computer, while displaying important flight information from the vehicle. The user first requests information options from the menu on items he/she would like displayed. This information is then sent from the HW to the internal SW, where the SW subscribes to the required UDP multicast addresses needed to gather said data. The information collected is then sent to the HW where the micro-controller renders the information on the display.

4. *Scrolling Display SW/HW:* The purpose of the scrolling display is to alert the operator of changes aboard the UAV. This is pre-programmed to contain messages of regulator voltages, battery voltages, amperage draw, current flight mode, commanded operation of joysticks (ie. camera or flight), and radio information (RSSI, REMRSSI). The SW subscribes to the necessary UDP multicast addresses and sends the information to the scrolling display HW. The software aboard the microprocessor interprets and displays the information aboard the LCD or the 7-segment display. The microprocessor collects information from the analog sliders that control Google Earth's camera display (latitude, longitude, and altitude) and the analog potentiometer that determine the brightness for the LCDs.

5. *JBG Data Collection SW/HW:* The HW associated with this SW first collects information from the required peripherals such as the joysticks, buttons, internal electrical monitors, and internal GPS. This information is then sent to the SW aboard the internal computer where it updates respective data clusters within the global program and

distributes the information over a UDP multicast. A standalone aspect of the HW is the associated flight timer display. The user can begin/pause/clear the timer display from a 3 position switch that the SW aboard the HW interprets.

6. *TCP Groundstation:* The purpose of this program is to allow the operator to visualize the information from the vehicle or visualize the vehicle's GPS position and flight path through Google Earth. This program also generates the flight commands from the joystick inputs, creates gimbal camera commands from joystick inputs, and overall sends information and updates to either the flight or payload computer aboard the UAV. This program connects to the mission server through a TCP connection for reliability of message structure and transmission, and is the only program that has uplink capabilities to the UAV. Sample screen captures from this interface are shown in Figure 3.23 below.

Figure 3.23: This image highlights two screens from the developed TCP Groundstation. The left image shows the joystick interface screen that relays flight commands to the UAV. The right image shows a sample flight path and UAV position in an embedded Google Earth window within LabVIEW.

The UGCS was housed in a Pelican case for durability during transport and operation. The UGCS weighs approximately 25 pounds and supports AC or DC power input options for remote operations. Figures 3.24 & 3.25 below show a sample operation of the UGCS and call out functional items within the system to allow the reader to understand the UGCS interface.

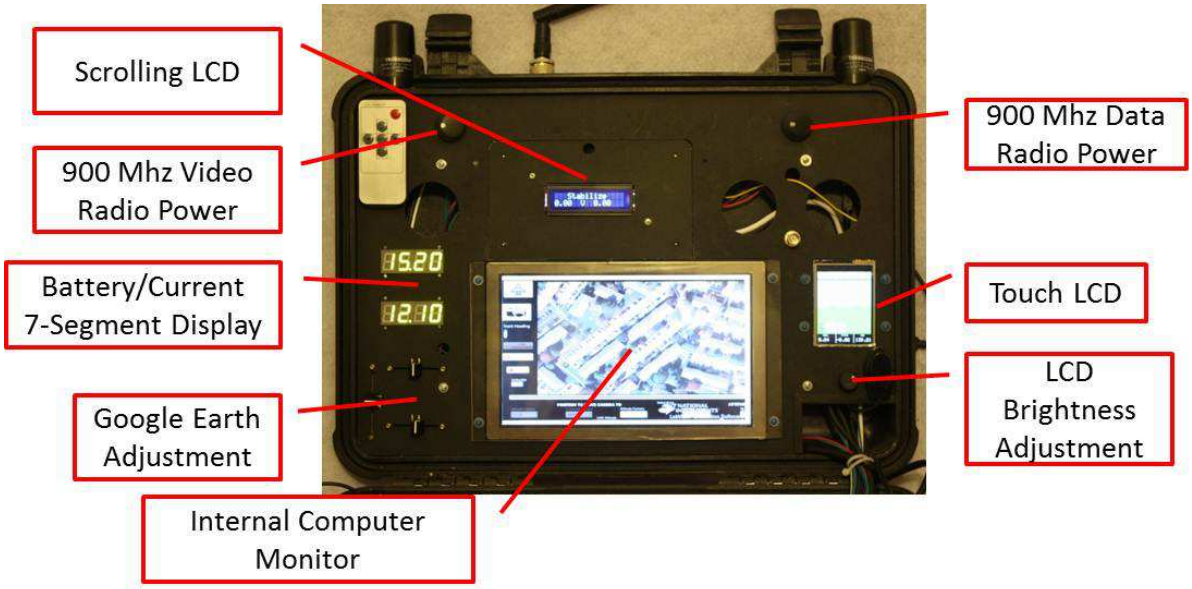


Figure 3.24: Top half of the UGCS.

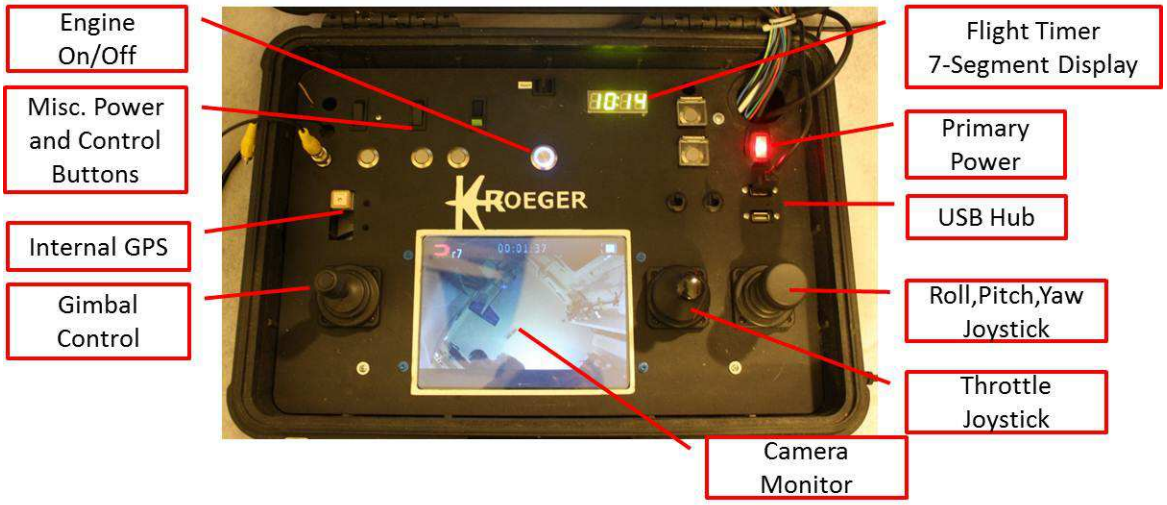


Figure 3.25: Bottom half of the UGCS.

3.6 Future Recommendations and Conclusions

Changes to address for future operations would be to reexamine the landing gear configuration. It would be desired that the landing gear configuration increase in height so that the

peripheral camera assembly could have further clearance from the ground. Also, it would be recommended to reexamine the location of the network switch and encoder which are currently located within the battery tray assembly. Moving each of these electronics would allow more room in the battery tray for larger or oddly shaped batteries. Lastly, it would be desired to consolidate the flight and peripheral computer onto one single processor. However, it should be noted that by keeping the computers separate, each can function and operate independently regardless of problems or software glitches contained in the other. Consolidating each computer would require that the final computer have larger computational capabilities to accommodate both systems operating, without having to increase the allotted mechanical footprint. Currently, the author is examining potential integration options with a sbRIO platform from National Instruments to accommodate both flight and payload operations.

In conclusion, this chapter has addressed the mechanical and electrical design, as well as the system architecture of a fixed pitch multirotor platform. The HexaCopter platform that has been developed will be used throughout the remaining duration of this thesis for modeling and controller development. Key design features have been discussed and how these features help meet the requirements of the system, benefited the design for airworthiness, durability of design, and the potential expansion for future applications have been addressed. The chapter had concluded discussing a developed communications architecture and UGCS terminal for interacting with the aircraft's flight and payload computers.

Chapter 4

Coordinate/Rotational Frames & Attitude Determination

4.1 Coordinate Reference Frames

Coordinate frames are widely used in many fields of engineering dynamic systems. Typically, when defining a coordinate system, they adhere to be right-handed and orthogonal. First, the definition of a coordinate system needs to define the location of the origin along with the definition of at least two axes. The coordinate system definition is completed by defining the third axis orthogonal to the other two established axes.

For UAV systems, there are many coordinate frames that are established to aid in the development of attitude and navigation control algorithms. A summary for reference of the coordinate frames used in the development of a UAV will be discussed in this section, as well as the transformations between coordinate frames. The necessary coordinate frames used in general UAV systems are:

1. The Geodetic coordinate system.
2. The Earth-Centered Earth-Fixed coordinate system.
3. The UAV carried North-East-Down coordinate system.
4. The Local North-East-Down coordinate system.

5. The Body coordinate system.

It is important to recognize that these five general coordinate frame definitions are used often in referring to UAV navigation and attitude. The first three coordinate frames are not used in this thesis because their focus is on the global positioning of the UAV rather than local. They are mentioned here merely for completeness and future research in the event that global positioning of the UAV is desired. Figure 4.1 below highlights the last two coordinate frames in reference to the UAV, and further details defining the coordinate frames are discussed below.

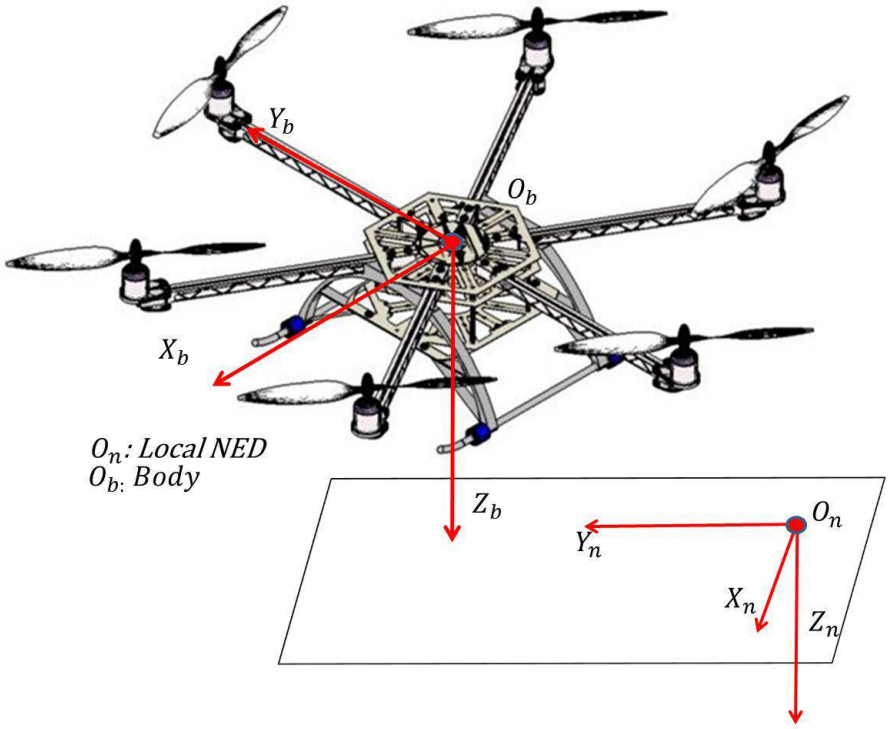


Figure 4.1: This figure highlights the two coordinate systems that are used throughout this work.

4.1.1 Local NED Coordinate System

The local NED coordinate is known as the ground navigational coordinate system. This coordinate frame is fixed to the Earth's surface and is often referenced at the takeoff point or initial sensor calibration point of the UAV start-up before flight. The origin and coordinate

frame are defined in Figure 4.1 and detailed below as:

1. The origin (O_n) is fixed to the earth's surface and originates at the initial start up and calibration of the aircraft.
2. The X-axis (X_n) points to ellipsoid north.
3. The Y-axis (Y_n) points to ellipsoid east.
4. The Z-axis (Z_n) points downward according to the right hand rule along the ellipsoid normal.

The local NED system usually establishes navigation commands for the aircraft based on the aircraft's body position, velocity, and acceleration vectors relative to the local NED origin. The position vector, velocity vector, and acceleration vector are defined respectively as followed

$$\underline{P_n} = \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} \qquad \underline{V_n} = \begin{bmatrix} u_n \\ v_n \\ w_n \end{bmatrix} \qquad \underline{A_n} = \begin{bmatrix} a_{xn} \\ a_{yn} \\ a_{zn} \end{bmatrix} \tag{4.1}$$

This coordinate frame will now be referred to as CF_n.

4.1.2 UAV Body Coordinate System

The UAV body coordinate system is the vehicle carried coordinate system that remains fixed to the body regardless of vehicle attitude. The body fixed frame aligns with the axis of the IMU attached to the UAV. An assumption made here is that the IMU will be at the center of gravity for the vehicle, allowing for ease of coordinate frame transformations. The origin and coordinate frame are defined in 4.1 and detailed below as:

1. The origin (O_b) is located at the center of gravity for the vehicle..
2. The X-axis (X_b) points to the front of vehicle.
3. The Y-axis (Y_b) points to the right of the vehicle.

4. The Z-axis (Z_b) points downward adhering to the right hand rule.

Acceleration and velocity vectors in the body frames can be expressed as

$$\underline{v_b} = \begin{bmatrix} u_b \\ v_b \\ w_b \end{bmatrix} \qquad \underline{a_b} = \begin{bmatrix} a_{xb} \\ a_{yb} \\ a_{zb} \end{bmatrix} \tag{4.2}$$

This coordinate frame will now be referred to as CF_b.

4.2 Rotation Euler Transformations

After defining several coordinate frames, it is often of interest to determine how each of the coordinate frames relate to each-other. The relationship between two coordinate frames is often determined through a transformation that will take a vector in one coordinate frame and rotate the information to be a vector in another coordinate frame. This section will discuss quickly how a rotation matrix works mathematically and some useful properties of rotation matrices. From here, Euler angle rotations are highlighted with details of how this form of rotation relates to a UAV.

4.2.1 Properties of a Rotation Matrix

Coordinate frame transformations occur when it is desired to characterize the relationship between a vector defined in one coordinate system and understand its representation in another coordinate system. For example, $\vec{v_b}$ is the components of a generic vector $\vec{v}$ projected onto the b reference frame which is described by the unit base vector $\left\{\hat{b}\right\}$. It is desired to understand its relation and its components in the i reference frame described by the unit base vector $\left\{\hat{i}\right\}$. If a transformation were to exist, it could be noted that

$$\vec{v} = \vec{v_i}^T \left\{\hat{i}\right\} = \vec{v_b}^T \left\{\hat{b}\right\} \tag{4.3}$$

and therefore we seek a linear transformation $\mathcal{R}$ such that

$$\mathcal{R}\left\{\hat{b}\right\} = \left\{\hat{i}\right\} \tag{4.4}$$

Substituting Equation 4.4 into 4.3 yields

$$\vec{v}_i^T \mathcal{R}\left\{\hat{b}\right\} = \vec{v}_b^T \left\{\hat{b}\right\} \tag{4.5}$$

We can use the definition of orthogonality such that $\left\{\hat{b}\right\} \odot \left\{\hat{b}\right\} = 1$ in Equation 4.5 and conclude that the coefficients of the terms must be equal. Thus,

$$\vec{v}_i^T \mathcal{R} = \vec{v}_b^T \tag{4.6}$$

Therefore, $\mathcal{R}$ is an orthogonal matrix that could satisfy

$$\mathcal{R}\left\{\hat{b}\right\} = \left\{\hat{i}\right\} \; or \; \mathcal{R}^T \left\{\hat{i}\right\} = \left\{\hat{b}\right\} \tag{4.7}$$

4.2.2 Euler Angle Rotations

As described above, one Cartesian coordinate system can be described in respects of another through a rotation matrix. The rotation from one frame to another can be visualized as a sequence of three rotations about a base vector. Each rotation about a specific axis through a specific angle is denoted as a Euler angle. Although all relative orientations between Cartesian frames can be described through Euler angles, the focus of Euler angles for the purpose of this work is to describe the transformation angle between the UAV CF_b and the UAV CF_{nv}. Euler angles for each transformation from the body coordinate system to the local NED system are denoted as roll ϕ, pitch θ, and yaw ψ. For the rest of this work well denote $\Theta \, \epsilon \, \mathbb{R}^{3x1}$ where $\Theta = [\phi, \theta, \psi]^T$. A description of each is denoted below along with Figure 4.2. For each equation c and s represent the trigonometric functions cosine and sine respectively.

- The roll angle denoted as phi ϕ, is the angle from either the Y_{nv} or the Z_{nv} axis to the

respective body axis. The rotation takes place about the X-axis.

$$\mathcal{R}_{\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & s\phi \\ 0 & -s\phi & c\phi \end{bmatrix} \tag{4.8}$$

- The pitch angle denoted as theta θ, is the angle from either the X_{nv} or the Z_{nv} axis to the respective body axis. The rotation takes place about the Y-axis.

$$\mathcal{R}_{\theta} = \begin{bmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\theta \end{bmatrix} \tag{4.9}$$

- The yaw angle denoted as psi ψ, is the angle from either the X_{nv} or the Y_{nv} axis to the respective body axis. The rotation takes place about the Z-axis.

$$\mathcal{R}_{\psi} = \begin{bmatrix} c\psi & s\psi & 0 \\ -s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.10}$$

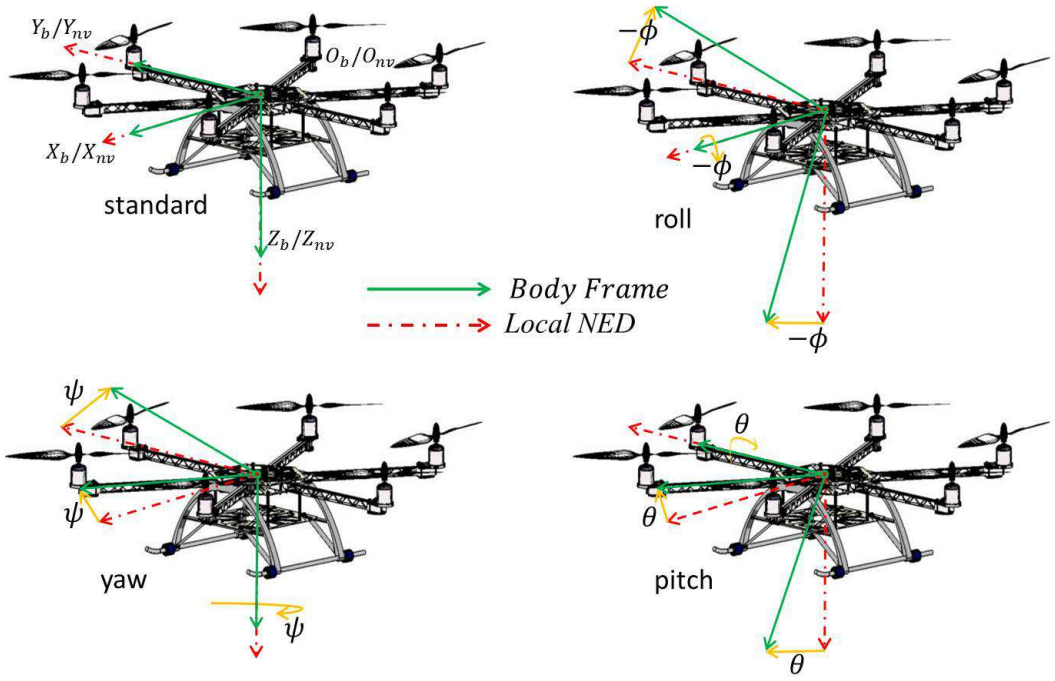


Figure 4.2: This figure represents how Euler angles can describe vehicle attitude. The images walk through a roll, pitch, and yaw sequence definition aboard the HexaCopter platform.

The complete transformation from the CF_b to the CF_n is often referred to as the Direction Cosine Matrix (DCM), and is detailed in Equation 4.11 below.

$$\mathbf{R_b^n} = \mathbf{R}_\phi \mathbf{R}_\theta \mathbf{R}_\psi = \begin{bmatrix} c\theta c\psi & -c\phi s\psi + s\phi s\theta c\psi & s\phi s\psi + c\phi s\theta c\psi \\ c\theta s\psi & c\phi c\psi + s\phi s\theta s\psi & -s\phi c\psi + c\phi s\theta s\psi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \tag{4.11}$$

4.3 Attitude Estimation & Development

It can be seen that as the vehicle attitude becomes detailed mathematically, how useful the information above could be used in order to control and stabilize any UAV system. However, the next step is to understand how the above information can be collected. Today, the Inertial Measurement Unit (IMU) and Inertial Navigation System (INS) are leading the way in sensing and describing vehicle attitudes and position. Each continue to get smaller and cheaper as Micro-Mechanical-Electrical Systems (MEMS) technologies continue to advance

and develop more accurate and robust sensing capabilities. At the core of an IMU and INS are the sensor and data fusion packages that estimate the attitude of the system. The focus of the next several sections will be the build-up of a strap-down IMU system including the mathematical process modeling of its sensors and the presentation of a data-fusion algorithm known as the Extended Kalman Filter (EKF).

4.3.1 Attitude & Position Measurement Devices

The next several sections detail necessary sensors for the build up an IMU, and with the addition of a GPS, an INS.

4.3.1.1 Gyroscopes

A gyroscope will measure angular rates of roll, pitch, and yaw in the body frame denoted by the column vector $\omega \, \epsilon \, \mathbb{R}^{3x1}$ where $\omega = [p, q, r]^T$ and as shown in Figure 4.3 below. The state definition of the gyroscope's readings are denoted in Equation 4.12 below.

$$\underline{Gyro}_{readings} \, \epsilon \, \mathbb{R}^{3x1} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{4.12}$$

Since gyroscopes measure angular rates in the body frame, a transformation relationship is formed as shown in Equation 4.13 to transform the body frame measurements into Euler rates.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & sin(\phi)tan(\theta) & cos(\phi)tan(\theta) \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & \frac{sin(\phi)}{cos(\theta)} & \frac{cos(\phi)}{cos(\theta)} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{4.13}$$

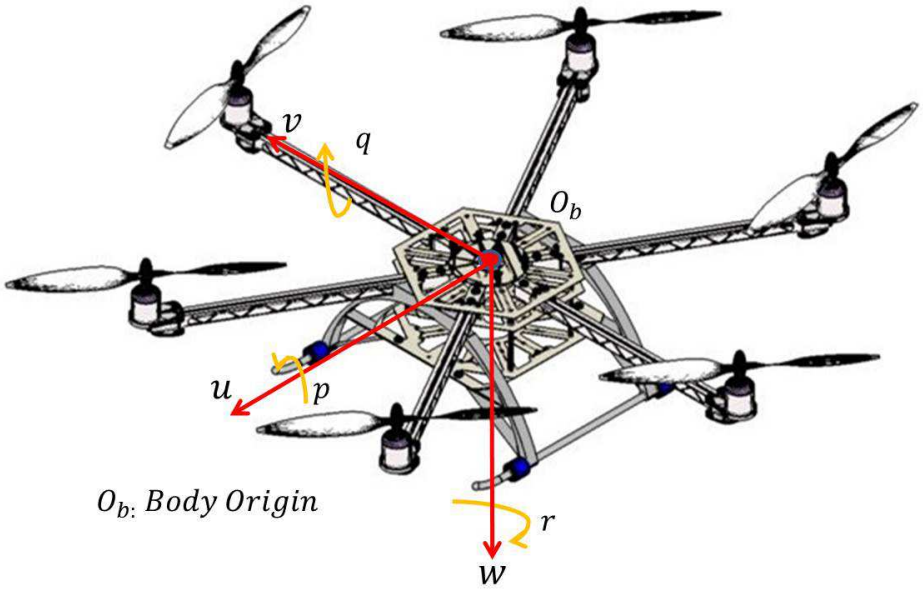


Figure 4.3: This figure details how the body velocities and body angular rates are defined on the HexaCopter platform.

If gyroscopes were true to their readings and contained no error, the Euler rates formed in Equation 4.13 above could be integrated and yield results for vehicle attitude. However, since sensors contain noise and error [10], the definition for the gyroscope reading vector can be redefined as

$$\underline{Gyro}_{readings} = \underline{Gyro}_{true} + \triangle\underline{Gyro} \tag{4.14}$$

where $\triangle Gyro$ contains errors of the gyroscope measurement. These errors include bias error $(\underline{Gyro}_{bias})$, scale error $(\underline{Gyro}_{scale})$, misalignment error $(\mathbf{MA_{Direction}})$, and noise $(\underline{Gyro}_{noise})$ as shown below in Equation 4.15.

$$\triangle\underline{Gyro} = \underline{Gyro}_{bias} + I(3,3) * \underline{Gyro}_{scale} * \underline{Gyro}_{true} + \begin{bmatrix} 0 & MA_{xy} & MA_{xz} \\ MA_{yx} & 0 & MA_{yz} \\ MA_{zx} & MA_{zy} & 0 \end{bmatrix} * \underline{Gyro}_{true} + \underline{Gyro}_{noise} \tag{4.15}$$

Terms such as the misalignment error and scale factor are constant and can be preprocessed

and set when acquiring data from the gyroscope. These terms usually are related to the placement of the gyroscope on the PCB, and the output voltage scaling from the chosen gyroscope. Thus, Equation 4.15 can be simplified to be

$$\triangle \underline{Gyro} = \underline{Gyro}_{bias} + \underline{Gyro}_{noise} \tag{4.16}$$

It is known that since the noise cannot be estimated in the system, we can can consider the remaining case to be $\triangle \underline{Gyro} \approxeq \underline{Gyro}_{bias}$. Therefore, it can be concluded that once these terms have been accounted for, the true gyro measurement is

$$\underline{Gyro}_{true} = \underline{Gyro}_{readings} - \underline{Gyro}_{bias} \tag{4.17}$$

where $\underline{Gyro}_{bias} \epsilon \, \mathbb{R}^{3x1}$ defined as

$$\underline{Gyro}_{bias} = \begin{bmatrix} Gyro_{xbias} & Gyro_{ybias} & Gyro_{zbias} \end{bmatrix}^T \tag{4.18}$$

Thus, having estimates of the gyro bias over time would be the best solution in developing true gyroscopic readings. Therefore, other sensors will be used to compliment the gyroscope and develop an ability for tracking the gyroscopic bias.

4.3.1.2 Accelerometers

An accelerometer is important in an IMU system because accelerations can be integrated once to achieve vehicle velocities and integrated again to achieve vehicle position. A 3 axis accelerometer was chosen for IMU development, therefore we can define the accelerometer reading vector as

$$\underline{Accel}_{readings} = \begin{bmatrix} Accel_{read_x} & Accel_{read_y} & Accel_{read_z} \end{bmatrix}^T \tag{4.19}$$

The next step in modeling the accelerometer is to consider the forces the accelerometer will measure in a dynamic environment. The output of a 3-axis accelerometer that is fixed on the body of a UAV measures the gravitational acceleration as well as the dynamic linear acceleration that occur due to the movements of the rigid body [31]. We will denote the

dynamic linear acceleration as

$$\triangle \underline{Accel} = \begin{bmatrix} \delta Accel_x & \delta Accel_y & \delta Accel_z \end{bmatrix} \tag{4.20}$$

Accelerometers are also very susceptible to noise and vibrations of the subject that they are to measure. The noise levels of these readings will contain both high and low frequency content and assumed to be Gaussian distributed white noise. For the purposes of completeness we will describe the noise of the accelerometer to be of two different variables for both high and low frequency content noise.

$$\underline{Accel}_{H-noise} = \begin{bmatrix} Accel_{HX-noise} \\ Accel_{HY-noise} \\ Accel_{HZ-noise} \end{bmatrix} \;\&\; \underline{Accel}_{L-noise} = \begin{bmatrix} Accel_{LX-noise} \\ Accel_{LY-noise} \\ Accel_{LZ-noise} \end{bmatrix} \tag{4.21}$$

Therefore the accelerometer reading can be redefined through implementing Equations 4.20 and 4.21 as

$$\underline{Accel}_{readings} = \triangle \underline{Accel} \pm \mathbf{R}_{\mathbf{n}}^{\mathbf{b}} \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \underline{Accel}_{H-noise} + \underline{Accel}_{L-noise} \tag{4.22}$$

If the dynamic linear acceleration is considered small in comparison to the components of the gravity vector one can neglect $\triangle Accel$. Also, a low-pass filter can be implemented on the processor of the IMU to ensure that the high frequency noise content of the accelerometer reading does not affect the attitude estimate. This relies on the assumption that changes in attitude are relatively slow compared to the frequency content capable in the readings of the IMU. These assumptions are a fairly large leap if the accelerometer was used in estimating vehicle attitude alone. However, when coupled with a secondary source for measurements, the readings from the accelerometer will merely complement the readings from the gyroscope above. Therefore, the assumed definition for the accelerometer readings will be

$$\underline{Accel}_{readings} = \underline{Accel}_{L-noise} \pm \mathbf{R}_{\mathbf{n}}^{\mathbf{b}} \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{4.23}$$

4.3.1.3 Magnetometers

A magnetic compass measures the heading of an object in which the compass is attached to in relation to Earth's magnetic field with respect to magnetic north [31]. Digital compasses are often referred to as magnetometers. For example, if one were to hold a basic two axis magnetometer on a local horizontal plane normal to the Earth's gravity vector, one would expect results similar to those shown in Figure 4.4 below. In this case, the observer rotated the compass CCW to produce these results. Normalized readings from both axes would follow the graph on the top. To convert these readings into a heading, Equation 4.24 can be applied yielding the results of the graph as shown in the figure at the bottom.

$$Heading = \arctan\left(\frac{-Y_h}{X_h}\right) \tag{4.24}$$

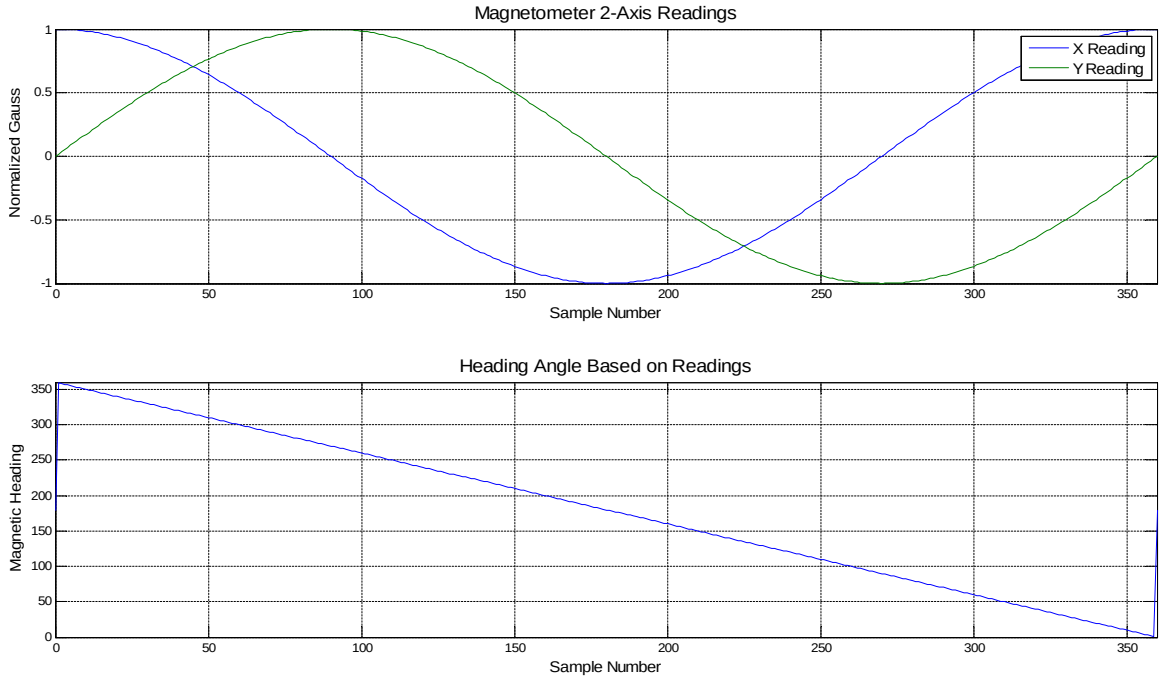


Figure 4.4: Simulated 2-axis compass readings and computed heading angle.

Often, the vehicle's heading which is to be determined is not on a level horizontal plane normal with Earth's gravity vector. Assuming that this fact had been neglected during heading calculations, heading error can accumulate and vary based on the magnetometers attitude relative to the horizontal plane [32]. Figure 4.5 below highlights this fact through

varying pitch angles of the vehicle. The y-axis shows the reading error from what the true heading measurement and the potential actual reading; while the x-axis highlights the true reading. Again, the readings would take place as the compass or vehicle is rotated through yaw while maintaining the pitch angle in question.

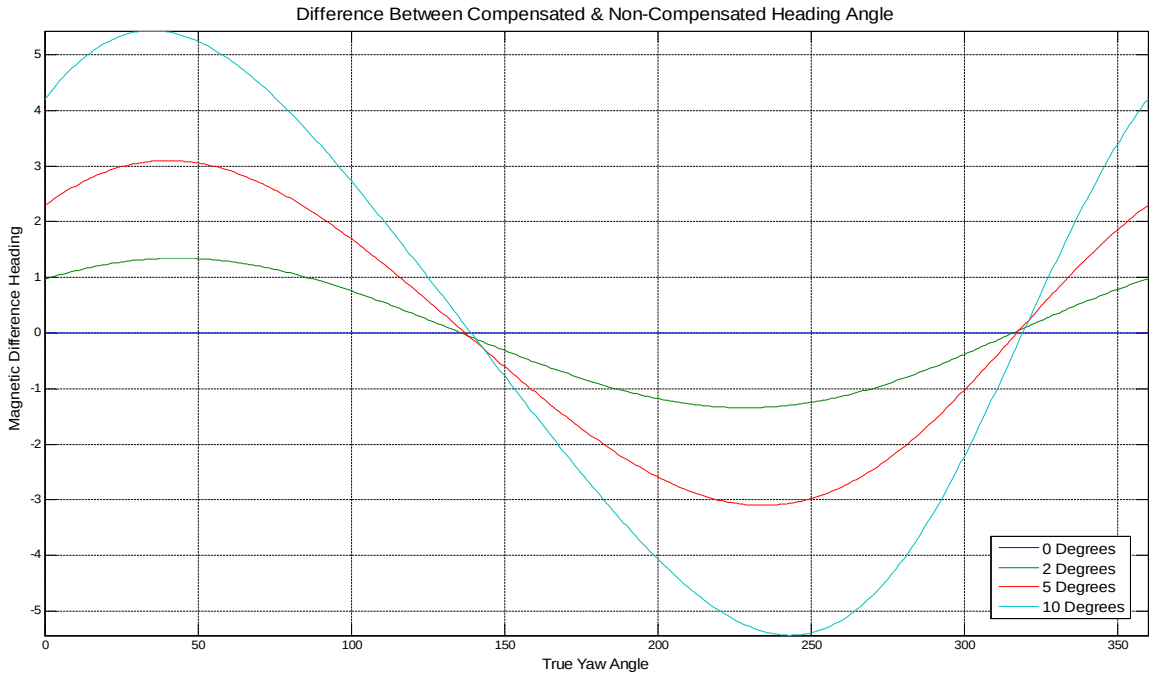


Figure 4.5: Simulated error in heading if magnetometer readings are not properly tilt compensated.

Assuming that the vehicle's roll and pitch attitudes are known, heading measurements can be corrected by transforming the measurements of the magnetometer into the horizontal plane. The easiest method for attitude based correction is to combine other attitude estimates with the 3-axis magnetometer measurements. This allows for prior rotation matrices to be applied for the heading measurement corrections. For this experiment the Honeywell HMC5883L was chosen to be the 3-axis digital magnetometer unit used to complement vehicle heading estimation. The readings from the magnetometer will be defined in the state vector shown in Equation 4.25 below.

$$\underline{Mag}_{reading} \, \epsilon \, \mathbb{R}^{3x1} = \begin{bmatrix} Mag_{read_x} \\ Mag_{read_y} \\ Mag_{read_z} \end{bmatrix} \tag{4.25}$$

The three values of $\underline{Mag_{reading}}$ can be transformed into the horizontal plane of X_h and Y_h by applying the rotations shown below in Equations 4.26 and 4.27 [32].

$$X_h = Mag_{read_x} cos(\theta) + Mag_{read_y} sin(\theta) sin(\phi) + Mag_{read_z} cos(\phi) sin(\theta) \tag{4.26}$$

$$Y_h = Mag_{read_y} cos(\theta) - Mag_{read_z} sin(\phi) \tag{4.27}$$

From here Equation 4.24 can be applied to calculate the heading. However, this heading will not account for the magnetic declination which causes variation between magnetic north and true north. This can be accounted for locally, such as Blacksburg, Virginia currently has a declination of $8^o10'$ West, which implies that the magnetic declination would be subtracted from the calculated heading. Therefore, Equation 4.24 can be updated to reflect the local magnetic declination as shown in Equation 4.28 below.

$$Heading = \arctan\left(\frac{-Y_h}{X_h}\right) \pm Mag_{declination} \tag{4.28}$$

4.3.1.4 Global Positioning System

GPS is a navigation system maintained and operated by the U.S. DoD. Currently there are 31 operating orbital satellites that circle the Earth twice a day and transmit information to GPS receivers. The GPS receivers then use triangulation by comparing time signals sent from the satellite to the time signal received to calculate the position of the receiver. Thus, with three satellites 2D position localization can be achieved, and with four or more satellites 3D position can be achieved. GPS technologies are continuing to push forward in better estimations of vehicle velocity and location. The addition of a GPS system that can provide another estimate of position states would be useful to take the IMU system and form an INS system. However, for the context of this thesis, a GPS was bought and used simply as a tracking device so that a ground station operator could see approximate vehicle location information overlaid onto Google Earth.

Figure 4.6: GS407 GPS Receiver unit used for gathering position information of UAV.

The GPS used for this thesis is a GS407 Helical GPS receiver. A helical GPS antenna receiver was chosen for this vehicle because typical GPS antenna require upward facing orientations with complete unobstructed views of the sky in order to achieve GPS lock. However, ceramic helical antennas sacrifice some antenna gain for the ability to achieve GPS lock in a wide variety of attitudes and through minor obstructing obstacles. Figure 4.6 above shows the GPS unit used in this experiment. Features that will have to be considered when implementing a GPS unit into a system architecture is the GPS update rates and start-up to fix times. This unit features a 4Hz output rate along with a 30 second start-up time.

4.3.2 MEMS IMU Development

The next step of the process was to create a unit that could collect information from all of the above sensor platforms. Therefore, a PCB was designed and fabricated to contain the above sensors, along with a processor, voltage regulator, and a serial port. The serial port was created so that the IMU could have communications with the host controller computer to relay the vehicle attitude information. An image of the developed IMU is pictured below in Figure 4.7. The dimensions of the IMU are 1.2”x1”x0.3”.

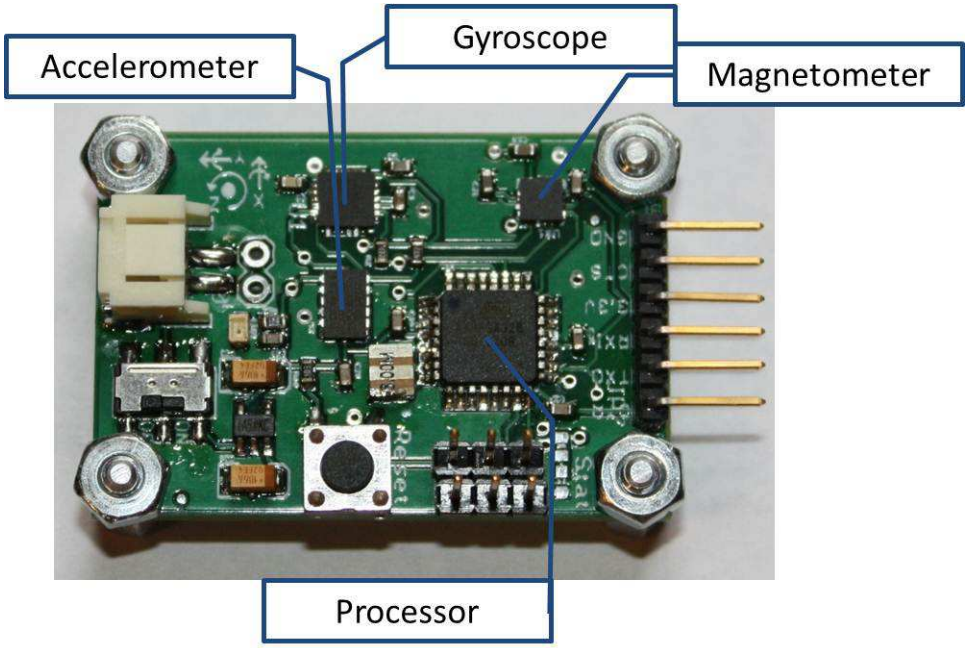


Figure 4.7: This figure highlights the IMU unit created to aid in sensory data collection to better estimate vehicle attitude.

4.3.3 Attitude Estimation Through EKF Derivation

The most difficult process in attitude estimation is determining a way to complement the potential measurement sets in such a manner to better estimate attitude. A variety of filters exist such as low-pass filters [33], complementary filters [34, 35, 33], and extensive amounts or variants of the Kalman Filter [36, 37, 9, 38, 10]. This work will focus on applying a KF with the developed IMU system since most literature suggests many potential benefits from applying a KF. The reason this technique for attitude estimation has been chosen is because of the wide application of the KF and the procedure in which the KF takes into account the process model of the system, measurement devices, system noise, measurement errors, and the uncertainty in the process model to produce an optimal state estimate for the system.

A KF is a recursive based observer which attempts to better estimate the states $\underline{x} \epsilon \mathbb{R}^n$ of a discrete time process governed by a linear stochastic difference equation [36] such as the discrete case given below by

$$\underline{x}_k = \mathbf{A}\underline{x}_{k-1} + \mathbf{B}\underline{u}_{k-1} + \underline{w}_{k-1} \tag{4.29}$$

with measurements $\underline{z} \epsilon \mathbb{R}^m$ governed by

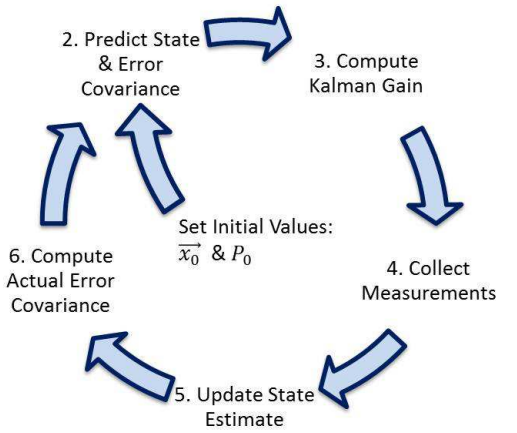
$$\underline{z}_k = \mathbf{H}\underline{x}_k + \underline{v}_k \tag{4.30}$$

Where variables $\underline{w}_k$ and $\underline{v}_k$ represent the process and measurement noise respectively and usually are represented by matrices of $\mathbf{Q}$ and $\mathbf{R}$. Assumptions for these processes are that they are assumed independent of one another, have zero mean distribution, and have normal probability distributions [39] defined as

$$\underline{w}_k \sim N(0, \mathbf{Q_{cov}}) \;\&\; \underline{v}_k \sim N(0, \mathbf{R_{cov}}) \tag{4.31}$$

However, the dynamics of a UAV system are highly nonlinear leaving several potential options when applying a Kalman Filter. First, if a linear KF process is desired, the system dynamics developed could be linearized about a predetermined condition in which the aircraft spent most of its time. For this type of UAV system, the UAV would spend the duration of its flight in a hover state. However, this would create inaccurate estimates if the UAV performed maneuvers outside this state. Next, Unscented Kalman Filter (UKF) research shows promising results and potential benefits. Yet, [33] shows that when implementing an UKF, processing time is substantially greater than that of other KF variants. Lastly, an Extended Kalman Filter (EKF) could be implemented by accounting for the nonlinear dynamics through linearized state equations and evaluating the linearization at the prior estimated state [10, 33, 36, 39]. Even though the state equations were linearized, this technique forms a better estimate than the standard linear KF since the next state estimate will be based closer on the prior state. This rids the requirement of choosing the state through which the vehicle could spend most of its time in. Using these decisions, an EKF was chosen to be implemented. Algorithm 4.1 highlights an overview of a generic KF process with the next sections detailing the process as it applies an EKF.

Algorithm 4.1 Generic KF Pseudo-Algorithm



1. $\overline{x_0}$, P_0
2. $\vec{x}_t^-$,P_t^-
3. K_t
4. z_t
5. $\overline{x_t}$
6. P_t

4.3.3.1 Moving Towards Quaternions

The first part of the process would be to develop a process model for the system. Therefore we can reexamine Equation 4.13 from above and restated below for completeness.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & sin(\phi)tan(\theta) & cos(\phi)tan(\theta) \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & \frac{sin(\phi)}{cos(\theta)} & \frac{cos(\phi)}{cos(\theta)} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{4.32}$$

From Equation 4.32 above, it can be seen that at pitch angles of $\pm 90^o$ several terms will tend towards infinity. This singularity is commonly referred to as gimbal lock. Also, it can be seen that this equation will not meet the requirements for a state definition. If we are defining a function that will yield Euler rates, the equation must consist of the state vector for Euler angles. Therefore, a change of state variables is required and literature searches [33, 39, 40, 41] suggest moving towards quaternions to alleviate both of these known problems.

Quaternions represent a 4-parameter set of variables based on the observation that any two coordinate systems can be related by a single rotation through an axis that has the same representation in each system. The four components of the quaternion can be defined as

$$q_0 = cos\left(\frac{\Phi}{2}\right) \tag{4.33}$$

$$\underline{q}_{1-3} = \vec{a} * sin\left(\frac{\Phi}{2}\right) \tag{4.34}$$

where Φ is the Euler angle and $\vec{a}$ is the Euler axis. The 3x1 vector component, $\underline{q}_{1-3}$, forms the Euler axis component of the quaternion, while q_0 forms the scalar component. The quaternion vector definition is defined in Equation 4.35 below.

$$\underline{q} = \begin{bmatrix} q_0 & \underline{q}_{1-3} \end{bmatrix}^T \tag{4.35}$$

Representing the quaternion vector as a function of Euler angles is presented in [41], and has been studied and stated extensively throughout other works of literature. This useful transformation is reiterated in Equation 4.36 below since this equation will be referenced and called out in later sections.

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} cos\left(\frac{\phi}{2}\right) cos\left(\frac{\theta}{2}\right) cos\left(\frac{\varphi}{2}\right) + sin\left(\frac{\phi}{2}\right) sin\left(\frac{\theta}{2}\right) sin\left(\frac{\varphi}{2}\right) \\ sin\left(\frac{\phi}{2}\right) cos\left(\frac{\theta}{2}\right) cos\left(\frac{\varphi}{2}\right) - cos\left(\frac{\phi}{2}\right) sin\left(\frac{\theta}{2}\right) sin\left(\frac{\varphi}{2}\right) \\ cos\left(\frac{\phi}{2}\right) sin\left(\frac{\theta}{2}\right) cos\left(\frac{\varphi}{2}\right) + sin\left(\frac{\phi}{2}\right) cos\left(\frac{\theta}{2}\right) sin\left(\frac{\varphi}{2}\right) \\ cos\left(\frac{\phi}{2}\right) cos\left(\frac{\theta}{2}\right) sin\left(\frac{\varphi}{2}\right) - sin\left(\frac{\phi}{2}\right) sin\left(\frac{\theta}{2}\right) cos\left(\frac{\varphi}{2}\right) \end{bmatrix} \tag{4.36}$$

Also, the DCM in terms of a unit quaternion is defined [37, 9, 38, 39, 40, 10, 41] as

$$\mathbf{RQ}_{\mathbf{b}}^{\mathbf{n}} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2\left(q_1 q_2 - q_0 q_3\right) & 2\left(q_0 q_2 + q_1 q_3\right) \\ 2\left(q_0 q_3 + q_1 q_2\right) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2\left(q_2 q_3 - q_0 q_1\right) \\ 2\left(q_1 q_3 - q_0 q_2\right) & 2\left(q_0 q_1 + q_2 q_3\right) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{4.37}$$

Therefore, the relationship between the readings from the gyroscopes as body turn rates to quaternions can be defined as

$$\underline{\dot{q}} = \begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{4.38}$$

4.3.3.2 Process Modeling & State Vector Definition

First, any variant form of the Kalman filter requires a process model. This process model contains information that is desired to be measured as well as the noise associated with the measurement. Therefore, since the system has been converted to quaternions and the gyro bias is to be estimated we can conclude that the nonlinear process model is denoted as

$$\underline{x} = f(\underline{x}, \underline{Gyro}_{readings}) + \underline{W} \tag{4.39}$$

where the state vector can be defined as $\underline{x} = \begin{bmatrix} \underline{Gyro}_{bias} & \overline{q} \end{bmatrix}^T \epsilon \mathbb{R}^{7x1}$ where $\overline{q}$ and $\underline{Gyro}_{bias}$ were defined in Equations 4.35 and 4.18 respectively above. Therefore, substituting $\overline{q}$ and $\underline{Gyro}_{bias}$ into the state vector definition yields

$$\underline{x} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \end{bmatrix}^T = \begin{bmatrix} Gyro_{xbias} & Gyro_{ybias} & Gyro_{zbias} & q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T \tag{4.40}$$

Analyzing Equation 4.39 further, it can be seen that the nonlinear function $f(\underline{x}, \underline{Gyro}_{readings})$ contains both a state vector and an input. This nonlinear differential equation will describe the propagation of the quaternion vector as shown in Equation 4.38. It should be noted that the bias on each axis of the gyro must be subtracted from the gyroscopic reading as noted in Equation 4.17. Thus, the nonlinear function $f(\underline{x}, \underline{Gyro}_{readings})$ can be defined as

$$f(\underline{x}, \underline{Gyro}_{readings}) = {}^1\!/_2 \begin{bmatrix} zeros(3,1) \\ \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} p - Gyro_{xbias} \\ q - Gyro_{ybias} \\ r - Gyro_{zbias} \end{bmatrix} \end{bmatrix} \tag{4.41}$$

Discretizing Equation 4.41 through the Euler method yields

$$\underline{x}_t = \underline{x}_{t-1} + f(\underline{x}_{t-1}, \underline{Gyro}_{readings}) \triangle t \tag{4.42}$$

4.3.3.3 Prediction Stage

Now that a process model has been formed, the first two steps of Algorithm 4.1 can be examined. The first step is to predict the next state of the system based on the prior estimated state and the error covariance. During the first time of execution, an initial estimate is provided for the initial state vector and the error covariance matrix. We can use the initial conditions of the vehicle attitude to calculate the initial state of $\underline{q}$ found in the state vector $\underline{x}$. Therefore, if the vehicle is flat on a level surface it can be seen from Equation 4.36 that the only remaining factor is the heading angle of the vehicle as shown in Equation 4.43.

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} 1 * cos\left(\frac{\psi}{2}\right) + 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.43}$$

Therefore, the initial state vector and the error covariance matrix can be defined as shown in Equations 4.44 and 4.45 below.

$$\underline{x}_0 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^T \; when \, \psi = 0 \tag{4.44}$$

$$\mathbf{P_0} = \mathbf{I}(7,7) \tag{4.45}$$

Based on the prior estimate, a prediction needs to be made about the state and error covariance at the current time step. Current time steps will be denoted as subscript (t), while previous time steps will be denoted as subscript (t-1). Predictions will also be denoted by superscript (-). Using the process model as derived in Equation 4.42, the initial state vector will provide the information for the estimate based on the current gyro information.

$$\underline{x}_t^- = \underline{x}_{t-1} + f(\underline{x}_{t-1}, \underline{Gyro}_{readings}) \triangle t \tag{4.46}$$

Substitution of the state variables and gyroscopic measurements into the nonlinear function will yield

$$f(\underline{x}_{t-1}, \underline{Gyro}_{readings(t)}) = {}^1\!/_2 \begin{bmatrix} zeros(3,1) \\ \begin{bmatrix} -x_5 & -x_6 & -x_7 \\ x_4 & -x_7 & x_6 \\ x_7 & x_4 & -x_5 \\ -x_6 & x_5 & x_4 \end{bmatrix} \begin{bmatrix} p - x_1 \\ q - x_2 \\ r - x_3 \end{bmatrix} \end{bmatrix} \tag{4.47}$$

Next, the error covariance for the current time step must be predicted as shown in Equation 4.48 below.

$$\mathbf{P}_{\mathbf{t}}^{-} = \mathbf{F}_t \mathbf{P}_{t-1} \mathbf{F}_t^T + \mathbf{Q} \tag{4.48}$$

where $\mathbf{F}_t$ is the state transition matrix defined as $\mathbf{F}_t \epsilon \mathbb{R}^{nxn}$ and $\mathbf{Q}$ is the covariance matrix of w_k defined as $\mathbf{Q} \epsilon \, diag\,(\mathbb{R}^{nxn})$. $\mathbf{Q}$ is established prior to implementation of the Kalman filter and is determined based on the characteristics of the system. $\mathbf{F}_t$ is calculated as the Jacobian matrix of the partial differentiation of the function $f(\underline{x}_{t-1}, \underline{Gyro}_{readings(t)})$ evaluated at the prior estimated state.

$$\mathbf{F}_t = \left(\frac{\partial f(\underline{x}, \underline{Gyro}_{readings(t)})}{\partial \underline{x}} \right)_{\underline{x}_{t-1}} = \begin{bmatrix} zeros(3,7) \\ x_5 & x_6 & x_7 & 0 & -(p-x_1) & -(q-x_2) & -(r-x_3) \\ -x_4 & x_7 & -x_6 & p-x_1 & 0 & r-x_3 & -(q-x_2) \\ -x_7 & -x_4 & x_5 & q-x_2 & -(r-x_3) & 0 & p-x_1 \\ x_6 & -x_5 & -x_4 & r-x_3 & q-x_2 & -(p-x_1) & 0 \end{bmatrix}_{\underline{x}_{t-}} \tag{4.49}$$

4.3.3.4 Correction Stage

At the correction stage, the predicted estimate $\underline{x}_t^-$ can be fused with observation information about the state to refine the state estimate. First, the Kalman gain needs to be computed. This allows a variable weighting system to be applied to the prediction based on the measurements collected. It can be seen from Equation 4.50 below that a new variable $\mathbf{H}_t$ is introduced as the state to measurement transition matrix where $\mathbf{H}_t \epsilon \mathbb{R}^{mxn}$.

$$\mathbf{K}_t = \mathbf{P}_t^- \mathbf{H}_t^T \left(\mathbf{H}_t \mathbf{P}_t^- \mathbf{H}_t^T + \mathbf{R} \right)^{-1} \tag{4.50}$$

where $\mathbf{P}_t^-$ is the predicted error covariance from Equation 4.48 above, $\mathbf{R}$ is the covariance matrix of v_k defined as $\mathbf{R} \, \epsilon \, diag\,(\mathbb{R}^{mxm})$, and $\mathbf{H}_t$ is computed as the Jacobian matrix of the partial differentiation of the function $h(\underline{x}_t^-)$, where $h(\underline{x}_t^-)$ is defined as a state to measurement column vector. It is important at this point to define our measurement vector. There are many different measurement update techniques available, however this thesis will focus simply on the accelerometer and magnetometer measurement directly for update information. Therefore, the measurement vector $\underline{z}_t$ can be defined as

$$\underline{z}_t \epsilon \mathbb{R}^{6x1} = \begin{bmatrix} Accel_{read_x} \\ Accel_{read_y} \\ Accel_{read_z} \\ Mag_{read_x} \\ Mag_{read_y} \\ Mag_{read_z} \end{bmatrix} \tag{4.51}$$

Keeping our measurement vector in mind, we look ahead to Equation 4.52 below where we analyze how our final state is determined from the EKF to determine how to form $h(\underline{x}_t^-)$.

$$\underline{x}_t = \underline{x}_t^- + \mathbf{K_t} \left(\underline{z}_t - h(\underline{x}_t) \right) \tag{4.52}$$

It can be seen that in parenthesis we subtract $h(\underline{x}_t^-)$ from our measurement vector $\underline{z}_t$. During this process, it is important to realize that this equation is forming our new state estimate based on the Kalman gain and the difference between the two prior vectors. Thus, it can be seen that we are trying to relate our measured states to predicted body frame states. Therefore, our state to measurement matrix $h(\underline{x}_t^-)$ is transforming our predicted state into the body frame in order to relate these measurements. Examining Equation 4.23 for the accelerometer measurement transformation into body frame, it can be seen that if the noise has been filtered (assuming ideal case), we can write the remaining measurement relation as

$$Accel_{readings} = \pm \mathbf{RQ_n^b} \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{4.53}$$

where $\mathbf{RQ_n^b}$ is defined in Equation 4.37. Thus, this defines our state transformation from quaternions to acceleration. Lastly, we need to examine the magnetometer transformations into the body frame and in relation to the quaternion state vector. The measurement of the magnetometer in the body frame can be related to the earth magnetic field vector $\underline{H}_{mag}$ as

$$\underline{Mag}_{readings} = \mathbf{RQ_n^b} \underline{H}_{mag} \tag{4.54}$$

where $\underline{H}_{mag}$ can be defined as

$$\underline{H}_{mag} = \begin{bmatrix} H_x & H_y & H_z \end{bmatrix}^T = \|H\| \odot \begin{bmatrix} cos(i) & 0 & sin(i) \end{bmatrix}^T \tag{4.55}$$

where $\left\|\underline{H}_{mag}\right\| \approx 0.515766G$ for the magnetic field strength in Blacksburg VA and the inclination is $i = 65^o 31'$ [32, ?]. With the estimated state vector related to the body coordinate system for the 6 measurements collected, $h(\underline{x}_t^-)$ can be written as

$$h(\underline{x}_t^-) = \begin{bmatrix} 2g\left(x_5x_7 - x_4x_6\right) \\ 2g\left(x_4x_5 + x_6x_7\right) \\ g\left(x_4^2 - x_5^2 - x_6^2 + x_7^2\right) \\ \left(1 - 2\left(x_6^2 + x_7^2\right)\right)H_x + 2\left(x_5x_7 - x_4x_6\right)H_z \\ 2\left(x_5x_6 - x_4x_7\right)H_x + 2\left(x_4x_5 + x_6x_7\right)H_z \\ 2\left(x_4x_6 + x_5x_7\right)H_x + \left(1 - 2\left(x_5^2 + x_6^2\right)\right)H_z \end{bmatrix} \tag{4.56}$$

Therefore, the Jacobian of Equation 4.56 can be computed as seen below.

$$\mathbf{H}_t = \left(\frac{\partial h(\underline{x}_t)}{\partial \underline{x}}\right)_{\underline{x}_t^-} =$$

$$\begin{bmatrix} & -2gx_6 & 2gx_7 & -2gx_4 & 2gx_5 \\ & 2gx_5 & 2gx_4 & 2gx_7 & 2gx_6 \\ & 2gx_4 & -2gx_5 & -2gx_6 & 2gx_7 \\ zeros(6,3) & -2x_6H_z & 2x_7H_z & -4x_6H_x - 2x_4H_z & -4x_7H_x + 2x_5H_z \\ & -2x_7H_x + 2x_5H_z & 2x_6H_x + 2x_4H_z & 2x_5H_x + 2x_7H_z & -2x_4H_x + 2x_6H_z \\ & 2x_6H_x & 2x_7H_x - 4x_5H_z & 2x_4H_x - 4x_6H_z & 2x_5H_x \end{bmatrix} \tag{4.57}$$

4.3.3.5 State Estimate Update

Substituting the result of Equation 4.57 into Equation 4.50 solves for the updated Kalman gain of the system at the current time step. Next, measurement data at time t is collected and denoted as the vector $\underline{z}_t$ as shown above in Equation 4.51. Next, the Kalman gain from above applies a weighting scheme across the error between the predicted measurement and the actual measurement. Therefore, a new estimated state at the current time is computed through Equation 4.58.

$$\underline{x}_t = \underline{x}_t^- + \mathbf{K_t}\left(\underline{z}_t - h(\underline{x}_t)\right) \tag{4.58}$$

To convert the computed quaternion vector back into the intuitive Euler angles, one can apply Equation 4.59 below.

$$\begin{aligned} \phi &= \arctan\left(2 * \frac{q_2q_3 + q_0q_1}{q_0^2 - q_1^2 - q_2^2 - q_3^2}\right) \\ \theta &= -\arcsin\left(2 * (q_1q_3 - q_0q_2)\right) \\ \psi &= \arctan\left(2 * \frac{q_1q_2 + q_0q_3}{q_0^2 + q_1^2 - q_2^2 - q_3^2}\right) \end{aligned} \tag{4.59}$$

Lastly, update the predicted error covariance as shown in Equation 4.60 below.

$$\mathbf{P}_t = \mathbf{P}_t^- - \mathbf{K}_t\mathbf{H}_t\mathbf{P}_t^- \tag{4.60}$$

It should be noted that if $\mathbf{P}_t$ is increasing or large, this means that the error is increasing implying that the error of the estimate is also large. Also, generic to all variants of Kalman

Filters is the requirement of $\mathbf{P}_t$ and $\underline{x}_t$ below.

$$\underline{x}_{true(t)} \sim N(\underline{x}_t\,,\,\mathbf{P}_t) \tag{4.61}$$

Equation 4.61 implies that $\underline{x}_{true(t)}$ follows a normal distribution with a mean of mean($\underline{x}_t$) and covariance $\mathbf{P}_t$ [33]. Therefore, if one were to examine a normal distribution curve, the peak would occur at the mean of the state and the width of the possible states would be determined by $\mathbf{P}_t$. Figure 4.8 below elaborates this fact further by showing that with a wider curve and high probability distributed across potential state estimates, the range of potential values for the state increases while increasing potential error as well. However, if the probability is high and tight around the mean, the state estimate will produce little error around the true value.

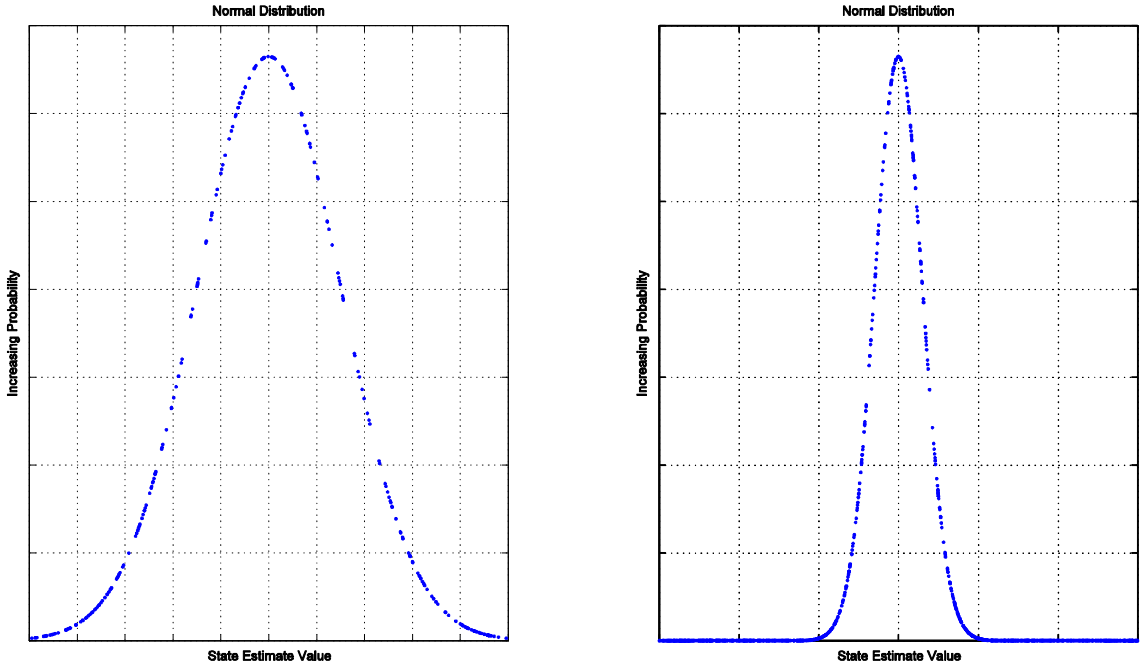


Figure 4.8: It can be seen that by the figure on the left, a larger covariance distributes the potential state estimated values away from the mean of the state. But rather on the right, a lower covariance implies that the state estimate is tighter around the mean value of the state.

A mathematical description of this is that error covariance can be defined as the mean square of the error estimate, which implies that the error covariance is proportional to the error of

the estimate [42].

$$P_t = E\left\{\left(x_{true} - x_{estimate}\right)\left(x_{true} - x_{estimate}\right)^T\right\} \tag{4.62}$$

4.3.4 Attitude Estimation Through EKF Implementation

Testing an IMU system is an in-depth and thorough task in order to best gather estimates of vehicle attitude. Data is collected and compared to data sheet statistics on the noise and variance levels of the sensors on-board the IMU. Data is also captured on a flat and level surface to collect any bias and offset information seen in the sensors. An important note to consider when tuning the $\mathbf{Q}$ and $\mathbf{R}$ matrices in the KF is the effect these matrices have on the predicted state of the system. It can be seen that from Equations 4.58 and 4.60 that the Kalman gain plays a large role in determining how the system performs. Therefore, a quick intuitive examination of the Kalman gain is required to understand how to better estimate current attitude states. Reexamining the Kalman gain equation from above as

$$\mathbf{K}_t = \mathbf{P}_t^- \mathbf{H}_t^T \left(\mathbf{H}_t \mathbf{P}_t^- \mathbf{H}_t^T + \mathbf{R}\right)^{-1} \tag{4.63}$$

it can be seen that $\mathbf{R}$ has a direct effect on the Kalman gain, while $\mathbf{Q}$ has an effect through $\mathbf{P}_t^-$ from Equation 4.48. It can be seen that if $\mathbf{Q}$ were to increase, the predicted error covariance would increase as well. Therefore, since the denominator of Equation 4.63 has an additional $\mathbf{R}$ term, it can be said that the Kalman gain would increase if $\mathbf{Q}$ increases. Thus, since the Kalman gain increases the contribution of the current measurement increases. From this understanding, $\mathbf{R}$ has an opposite effect than that of $\mathbf{Q}$.

Another practical implementation issue to consider is the initialization of the state of the system. For the derivation, we had assumed an ideal case in which the IMU was on a level surface, and the heading that it was facing was taken to be zero. However, this will practically never be true and therefore some forethought into the initialization sequence must be done. The purpose of implementing a state estimating filter is because of the inabilities of any one sensor to give accurate information consistently overtime. Ideally, the gyroscope information would be used and integrated to determine attitude. Yet, the whole reason for implementing the EKF above was because of the drift in the gyroscope measurements. In actuality, upon initialization, the gyroscope measurements can be taken as true as the bias

term hasn't had time to grow to a significant amount yet, and yaw can be determined based on the magnetometer information directly and tilt compensated from the gyroscopic reading. To quickly compute the quaternion vector, a small angle approximation can be made on 4.36 to simplify and improve start-up times. Other works, [9], suggest using the accelerometers for initialization of the state vector is also a viable solution for initialization.

A last practical note in implementing an IMU system is to highly consider the computational requirements of the math involved for each equation above. Terms that have multiplication and division consume many clock cycles within a processor and can severely dampen performance of the state estimation from the IMU. Another processing element that causes severe lags in computation time is processing trigonometric functions. A way to avoid computing trigonometric functions is to use their inherent properties of symmetry and create lookup tables for computations, however, it should be noted that a programmer should be mindful of floating point precision when using trigonometric processes in a microprocessor. Also, the table created could potentially consume a fair portion of memory depending on the precision and interpolation required. For this work, the output of the IMU is in quaternions and the conversion takes place on the main computer where there is a larger availability of memory and computational space.

A sample set of data was collected and is shown in Figures 4.9-4.10 below. The data collected was during IMU movement through roll, pitch, and yaw rotations in no specific order or magnitude. First, Figure 4.9 details the raw collected data from IMU sensors. Lastly, Figure 4.10 shows estimated attitude of the IMU represented in Euler angles after the estimated quaternion vector had been converted.

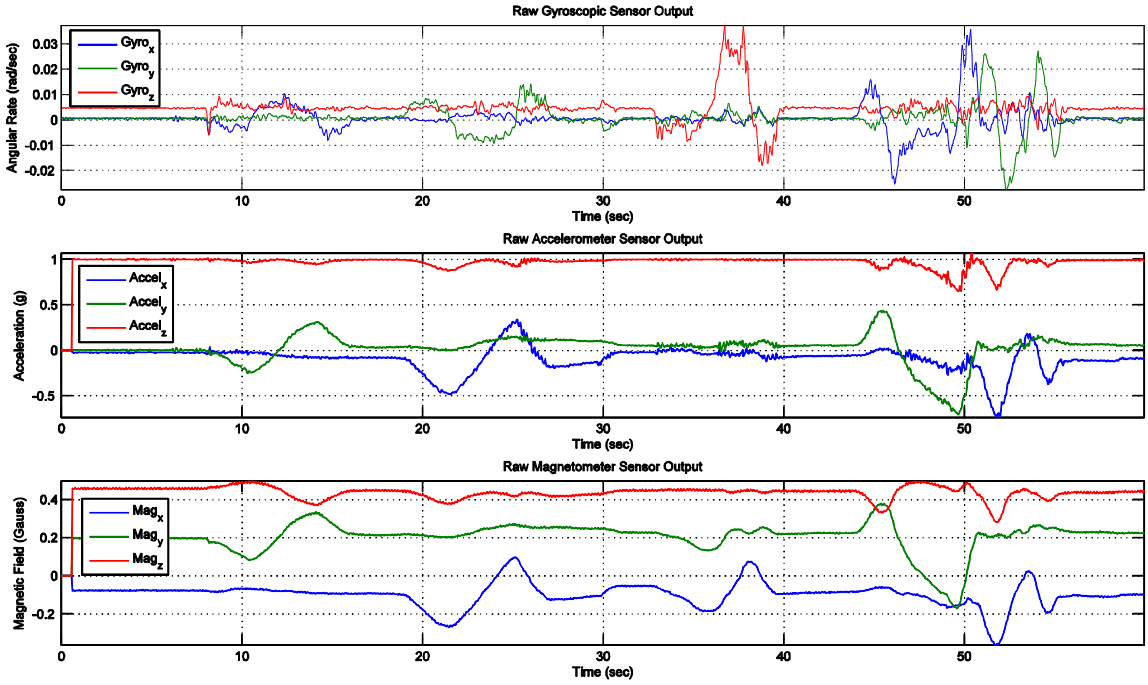


Figure 4.9: Raw data from sensors on-board the IMU during sample testing.

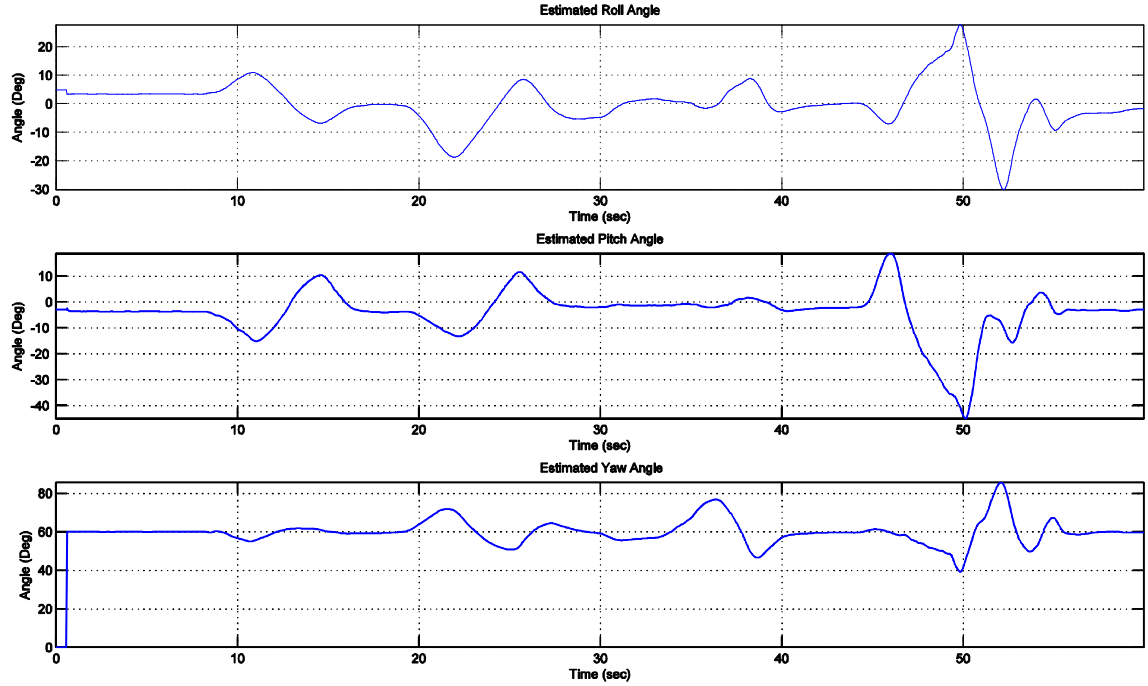


Figure 4.10: Initial performance results from EKF algorithm while estimating vehicle attitude in roll, pitch, and yaw.

Even though it seems that the IMU is performing as expected, the accuracy of the estimated vehicle attitude is unknown. Therefore, it is desired that this IMU be compared to a COTS IMU of known quality. Both the IMU developed in this thesis and a Microstrain 3DM-GX2 were aligned on a platform to conduct a simple test. The purpose of the test was to see how the IMU for this thesis compared in magnitude and rate of the estimated Euler state. Data was collected as the platform was moved through a roll, pitch, and yaw sequence and then post-processed for analysis. The sequence of rations is shown below in Figure 4.11.

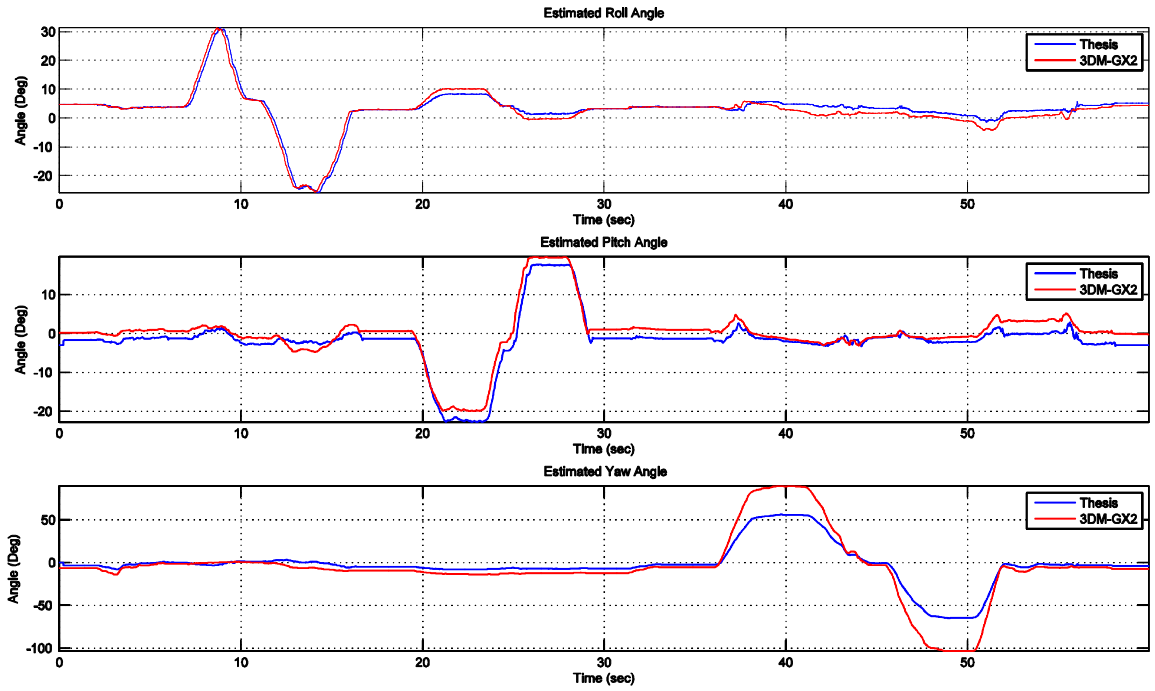


Figure 4.11: Comparative data set between thesis developed IMU and a 3DM-GX2 unit.

It can be seen from Figure 4.11 that the IMU developed for this thesis performs well when compared to the 3DM-GX2. Trends in all three attitudes are similar as well as in magnitude. However, the noticeable difference occurs during the yaw maneuver where there is considerable difference in magnitude of the turn. The surface on which the platform rested contained marks at $\pm 90^o$. The attempt was to maneuver yaw close to this rotation so that some qualitative analysis could be done. It appears that while the IMU developed for this thesis had underestimated the rotation, the 3DM-GX2 had overestimated the actual value. Given the scope and breadth of this thesis already, further testing will not be performed to determine better yaw attitude estimation.

4.3.5 Chapter Conclusion & Future Recommendations

In conclusion, this chapter has established and covered the coordinate frames necessary for inner-loop control of a UAV system. After discussing rotation matrices, Euler angles were presented as a representation of vehicle attitude. Understanding vehicle attitude is important in the development of control loops for vehicle stability. Therefore, the sensors needed to correctly estimate vehicle attitude were modeled and discussed. Finally, an EKF was presented that could run on-board a micro-processor that would collect gyroscopic measurements and augment the state prediction of vehicle attitude with measurements from both accelerometers and magnetometers.

With great success presented at the end of this chapter, there are still many recommendations that would be suggested for further improvements of this IMU system. First, other filtering techniques such as the UKF and Complementary filters could be implemented and explored to realize potential estimation or computational benefits from varying techniques. Further testing could be done to calibrate the IMU with known precision instruments for gyroscopic and accelerometer data. Magnetic field strength and magnetic field calibrations could also be performed on the magnetometer so that better yaw corrections could be made. Lastly, misalignment errors on the PCB were not accounted for as the space is "small" and the proximity of one sensor to another was close. However, if further completeness of the system would be desired, calculation and transformations would be implemented to account for the sensor misalignment from axis to axis. Lastly, the system contains a GPS unit that could close the loop on the system, allowing the vehicle to follow way-point trajectories and create an INS rather than a IMU. This would be useful because rather than estimating position based on integrated calculations, one could develop another staged Kalman filter that could incorporate the GPS information as a corrective estimate for calculations of the position on a global scale of the vehicle.

Chapter 5

Multirotor Mathematical and Dynamic Development

Once the vehicle had been developed, it was required that a mathematical model of the system be developed in order to simulate controller designs and evaluate performance before implementation into flight. This chapter first defines the propeller direction and reference for dynamic modeling while presenting how the HexaCopter translates and ascends or descends with changes in attitude or throttle. Next, this chapter derives the equations of motion presented as a derivation of the 6 DOF rigid-body equations derived using the Newton Euler method where forces and moments about the vehicle are analyzed. Third, the measurement model is quickly presented as it relates to the dynamics that are measured by the IMU presented above. Fourth, this chapter presents analysis of the motor model. The motor model incorporates testing to analyze the response of the rotor, inertia calculations of the assembly, and drag and thrust factors relating to the propeller. Lastly, this chapter presents how the above information is incorporated into a SIMULINK model for testing and control development.

5.1 Achieving Rotational and Translational Motion

Traditionally, a rotary wing platform achieves rotation and translational motion through a swashplate mechanism that creates a varying thrust vector along the body of the vehicle and thus causes respective attitude changes which result in translations. A fixed pitch platform alleviates the mechanical complexity by causing changes in moment/torque about a specific axis of the vehicle through relative changes of propeller's angular speed. Figure 5.1 at the right highlights the rotor definition that will be referred to throughout the remainder of this work. The arrows refer to the direction of the rotor, while the bold, standard, or dotted outline of the arrow refers to the relative increased, no change, or decreased RPM respectively of the rotor in relation to the other rotors. It can be seen from this image that an increase in all of rotor's angular speeds will result in the vehicle to increase thrust and climb in altitude, while a decrease in all of the rotor's speeds will decrease thrust and descend in altitude. Similarly, increasing thrust while in translational motion will increase the velocity vector of translation. Pure rotational maneuvers causing changes about a single axis are discussed below.

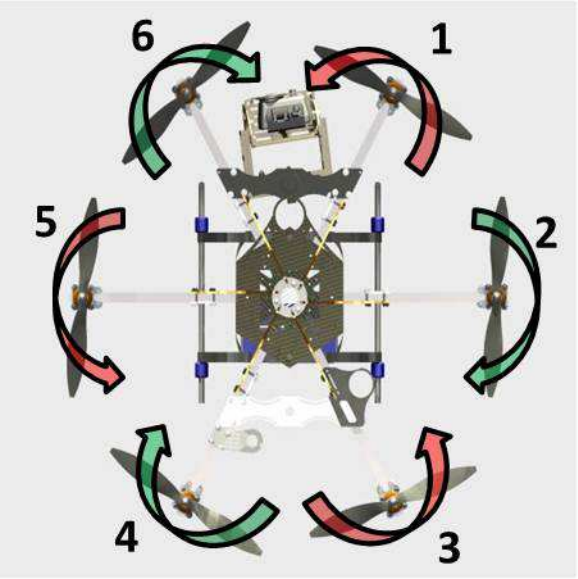


Figure 5.1: Motor numerical assignment for future reference and default direction of rotation for assigned propeller.

Roll Rotation: A pure roll maneuver of the vehicle (about x-axis) is achieved by increasing the angular speed of $[\Omega_1, \Omega_2, \Omega_3]$ relative to the decrease in angular speed of $[\Omega_4, \Omega_5, \Omega_6]$ causing a roll of the vehicle counter-clockwise and thus translating the vehicle in the -y-direction. If the angular speed of $[\Omega_1, \Omega_2, \Omega_3]$ is decreased relative to the increase in angular speed of $[\Omega_4, \Omega_5, \Omega_6]$, this will cause a roll of the vehicle in the clockwise direction and thus translating the vehicle in the +y-direction. This is highlighted in Figure 5.2 below. An interesting note about roll maneuvers on a HexaCopter platform in this 'X' configuration is that this maneuver would cause a yawing rotation about the z-axis. However, later experimentation

and results show that the relative change in angular speed of the propellers for a roll maneuver is at a minimum of magnitude less than that required for a yaw maneuver. Therefore, a roll maneuver will likely not be able to overcome the inertia and cause a yawing maneuver.

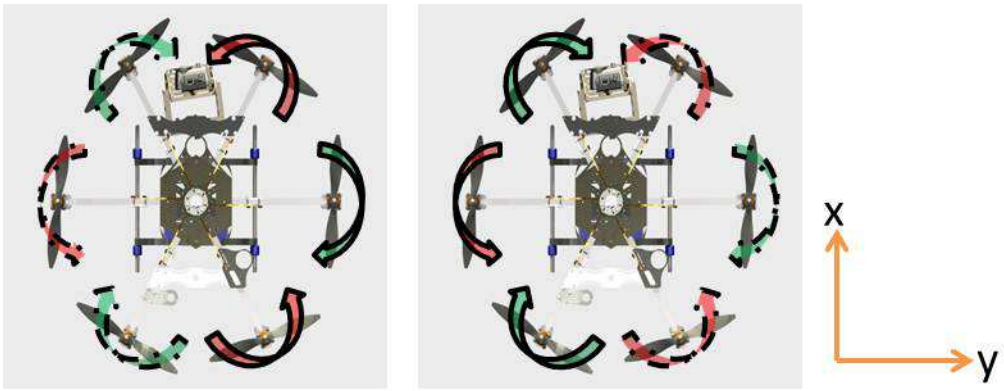


Figure 5.2: This figure details how changes in rotation about the x-axis (ROLL) are achieved. Figure LEFT results in -ϕ while Figure RIGHT results in ϕ.

Pitch Rotation: A pure pitch maneuver of the vehicle is achieved by increasing the angular speed of $[\Omega_3, \Omega_4]$ relative to the decrease in angular speed of $[\Omega_1, \Omega_6]$ causing a pitch of the vehicle causing the vehicle to translate in the +x-direction. If the angular speed of $[\Omega_3, \Omega_4]$ is decreased relative to the increase in angular speed of $[\Omega_1, \Omega_6]$, this will cause a pitch of the vehicle causing the vehicle to translate in the -x-direction. This is highlighted in Figure 5.3 below. An interesting fact about the HexaCopter platform is that in a pitch maneuver $[\Omega_2, \Omega_5]$ remain at the same angular speed as they have no ability to create a moment about the y-axis.

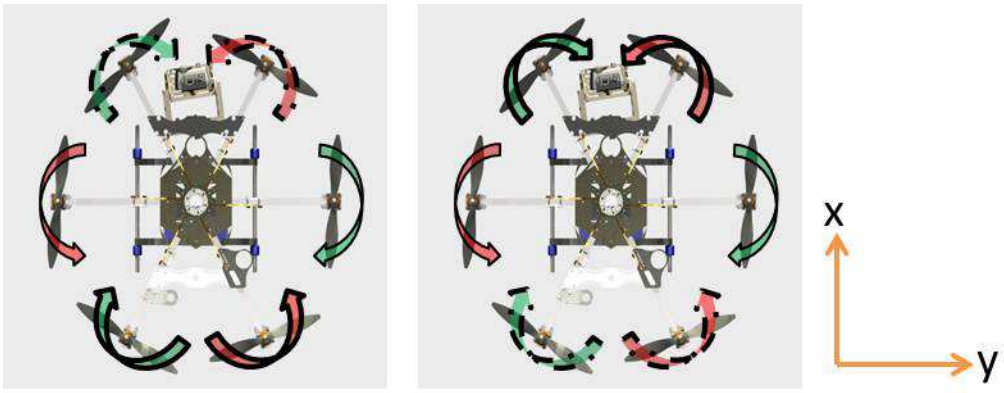


Figure 5.3: This figure details how changes in rotation about the y-axis (PITCH) are achieved. Figure LEFT results in $-\theta$ while Figure RIGHT results in θ.

Yaw Rotation: A pure yaw maneuver of the vehicle is achieved by increasing the angular speed of $[\Omega_1, \Omega_3, \Omega_5]$ relative to the decrease in angular speed of $[\Omega_2, \Omega_4, \Omega_6]$ causing a yaw of the vehicle clockwise direction. If the angular speed of $[\Omega_1, \Omega_3, \Omega_5]$ is decreased relative to the increase in angular speed of $[\Omega_2, \Omega_4, \Omega_6]$, this will cause a yaw of the vehicle in the counter-clockwise direction. The reason that the direction of rotation is opposite that of the direction of the higher angular speed propeller's is because the imposed higher aerodynamic drag on the higher angular velocity propellers. This is highlighted in Figure 5.4 below.

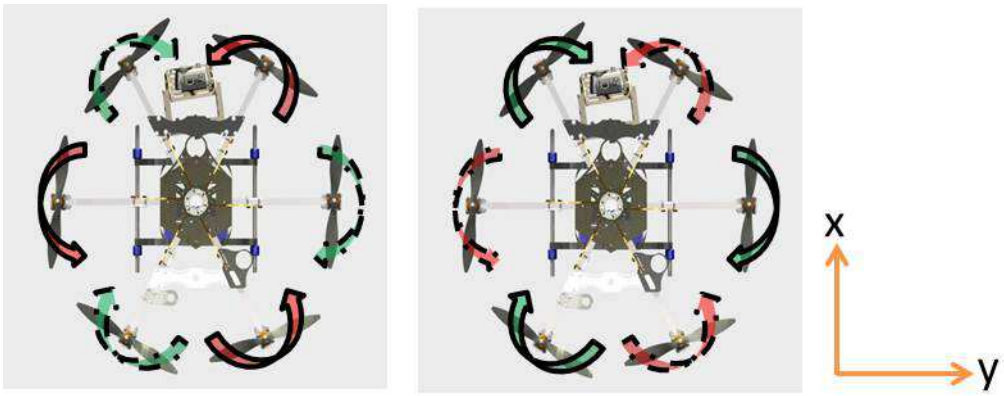


Figure 5.4: This figure details how changes in rotation about the z-axis (YAW) are achieved. Figure LEFT results in ψ while Figure RIGHT results in $-\psi$.

From here, it should be clear that a pure rotation about an axis is accomplished by causing an imbalance in moment about an axis by decreasing or increasing relative propellers angular

velocities. It was also seen from above that an increase or decrease in angular speed of all the rotors accomplishes changes in altitude or translational velocity. Expanding on these principles, it can be seen that a maneuver of combinations of roll, pitch, yaw, and thrust can be achieved by summing the desired changes in angular speed of the rotor during the maneuver for each individual rotor. Also, it should be clear that translational components of the vehicle are dependent on the angular result of the subsystem and the desired thrust component, however, there is no dependence of the rotational component on the translational component. This will be highlighted further and discussed in the control strategy section.

5.2 Equations of Motion

The first step into developing any form of control strategy is to analyze the system dynamics for the model in question. To do this, a mathematical derivation will be done to describe the dynamics of the system resulting in a nonlinear differential equation often referred to as the Equation of Motion (EOM). To develop the EOM of a system, typically the Lagrange [16, 20, 23, 24] or Newton-Euler Method [18, 20, 29, 30] can be employed. This work will focus on the Newton-Euler method of deriving the EOM by examining the sum of forces and moments on the body. First, we will highlight the assumptions that are made during this process.

Assumptions:

1. In order to employ the Newton-Euler method of modeling we must assume that the vehicle and all of its components are rigid.

2. Another rigidity constraint is that the propellers are assumed rigid. This allows the modeling process to ignore the effect referred to as blade-flapping. Blade-flapping occurs because of a difference in the advancing and retreating blade during horizontal translation. Blade flapping causes the rotor tip path plane to tilt effecting lift properties and requires extensive wind tunnel modeling and experimentation to validate [17]. The work concludes by suggesting that assuming rigid propellers is a valid for low velocity translational maneuvers.

3. Assuming that the vehicle is symmetrical allows us to reduce the cross products of inertia such as I_{xz}, I_{xy}, and I_{yz} to zero, reducing the inertia matrix to a diagonal

matrix where $\mathbf{I_n} = diag(I_{xx}, I_{yy}, I_{zz})$. Also, $\mathbf{I_n}$ is assumed constant in time. The values for the inertia matrix are generated based on the CAD model that had been developed for this work. The inertia matrix is defined as

$$\mathbf{I_n} = \begin{bmatrix} 0.0482 & 0 & 0 \\ 0 & 0.03431 & 0 \\ 0 & 0 & 0.07259 \end{bmatrix} Nms^2 \tag{5.1}$$

4. For this work, effects of ground effect have been ignored. Ground effect is the condition where there is a reduction of the induced airflow velocity and the reduction of rotor tip vortice loss. These two effects allow the rotorcraft to produce lifting forces at lower AOA and less power. However, this effect is only pronounced at altitudes through half of the rotor diameter, and therefore will be ignored for this work.

5. The center of gravity origin coinciding with the body fixed frame origin allows the EOM to reduce in complexity as there will be no net moment due to gravitational forces..

Next, we will examine the sum of the forces and moments acting on the system as set out by the Newton-Euler derivation and derive the EOM for the system.

5.2.1 Forces

From Newton's second law of motion for a mass system, we know that resultant of the external forces on any system of masses equals the total mass of the system times the acceleration of the center of mass $\sum F = m\vec{a}$. Therefore let us establish this equation in terms of the local-frame as shown in Equation 5.2.

$$\sum F = m\vec{a} \implies \sum F_n = m\underline{a}_n \tag{5.2}$$

The next step is to realize that we desire to derive the EOM relative to the body frame as this is the relative frame where measurements of the IMU are taken, and the commands of the control system are given. Therefore, we desire to relate $\underline{a}_n$ to $\underline{a}_b$ and $\sum F_n$ to $\sum F_b$.

$$\underline{a}_n = \dot{\underline{V}}_n \tag{5.3}$$

$$\dot{\underline{V}}_n = \dot{\widehat{\mathbf{R}^{\mathbf{n}}_{\mathbf{b}}\upsilon}} \tag{5.4}$$

$$\therefore \underline{a}_n = \dot{\widehat{\mathbf{R}^{\mathbf{n}}_{\mathbf{b}}\upsilon}} \Rightarrow \dot{\widehat{\mathbf{R}^{\mathbf{n}}_{\mathbf{b}}\upsilon}} = \dot{\mathbf{R}}^{\mathbf{n}}_{\mathbf{b}}\upsilon + \mathbf{R}^{\mathbf{n}}_{\mathbf{b}}\dot{\upsilon} \tag{5.5}$$

$$\therefore \underline{a}_n = \dot{\mathbf{R}}^{\mathbf{n}}_{\mathbf{b}}\upsilon + \mathbf{R}^{\mathbf{n}}_{\mathbf{b}}\dot{\upsilon} \tag{5.6}$$

$$F_n = \mathbf{R}^{\mathbf{n}}_{\mathbf{b}}F_b \tag{5.7}$$

Thus, substituting the result of Equations 5.6 and 5.7 into Equation 5.2 yields

$$\sum \mathbf{R}^{\mathbf{n}}_{\mathbf{b}}F_b = m\left(\dot{\mathbf{R}}^{\mathbf{n}}_{\mathbf{b}}\upsilon + \mathbf{R}^{\mathbf{n}}_{\mathbf{b}}\dot{\upsilon}\right) \tag{5.8}$$

The next step is to continue analyzing the right hand side of the Equation above. First, we can see that the time derivative of the rotation matrix yields

$$\frac{d}{dt}\mathbf{R}^{\mathbf{n}}_{\mathbf{b}} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \mathbf{R}^{\mathbf{n}}_{\mathbf{b}} \equiv \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \mathbf{R}^{\mathbf{n}}_{\mathbf{b}} = \mathbf{S}_{\mathbf{b}}\mathbf{R}^{\mathbf{n}}_{\mathbf{b}} \tag{5.9}$$

where $\mathbf{S}_{\mathbf{b}}$ is a skew-symmetric matrix defined as

$$\mathbf{S}_{\mathbf{b}} = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \tag{5.10}$$

Collecting like terms and dividing through yields Equations 5.11 and 5.12 respectively below

$$\sum \mathbf{R}^{\mathbf{n}}_{\mathbf{b}} \underline{F}_b = m\mathbf{R}^{\mathbf{n}}_{\mathbf{b}} \left(\mathbf{S_b} \upsilon + \dot{\upsilon}\right) \tag{5.11}$$

$$\sum \underline{F}_b = m \left(\mathbf{S_b} \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} \right) \tag{5.12}$$

Performing the matrix multiplication of the equation above yields

$$\sum \underline{F}_b = m \left(\begin{bmatrix} -vr + wq \\ ur - pw \\ -uq + vp \end{bmatrix} + \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} \right) \tag{5.13}$$

Equation 5.13 describes the forces causing the translational acceleration of the HexaCopter platform.

Note:

$$\underline{a}_n = \dot{\upsilon} + \omega^x \upsilon \tag{5.14}$$

where the linear acceleration vector in the body frame is $\dot{\upsilon} = \begin{bmatrix} \dot{u} & \dot{v} & \dot{w} \end{bmatrix}$, the linear velocity vector in the body frame is $\upsilon = \begin{bmatrix} u & v & w \end{bmatrix}$, the angular velocity vector is $\omega = \begin{bmatrix} p & q & r \end{bmatrix}$, and the x superscript denotes the cross-product. Using Equation 5.14 will yield the same results as those presented in Equation 5.13. Next, we must examine the forces on the vehicle to complete the left side of the equation. Figure 5.5 below highlights the primary forces on the vehicle such as the force due to gravity and the resulting force from the rotor thrust.

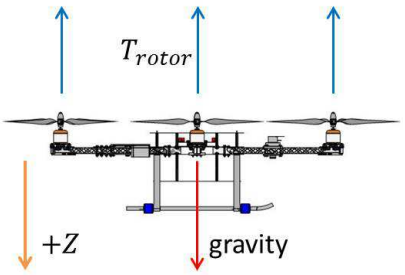


Figure 5.5: This figure highlights the forces assumed in the model derivation. Depicted is the directionality of the Z dimension along with the resulting forces from gravity and the propeller's thrust.

First, one can see by analyzing the forces due to gravity in local frame that

$$\underline{F}^{n}_{gravity} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{5.15}$$

where m is the total mass of the vehicle and g is the gravitational acceleration. We can then convert the gravitational force in the local frame to the body frame as shown in Equation 5.16 below.

$$\underline{F}^{b}_{gravity} = \mathbf{R}^{\mathbf{b}}_{\mathbf{n}}\underline{F}^{n}_{gravity} \;\therefore\; \underline{F}^{b}_{gravity} = \begin{bmatrix} -mg\sin(\theta) \\ mg\sin(\phi)\cos(\theta) \\ mg\cos(\phi)\cos(\theta) \end{bmatrix} \tag{5.16}$$

The last force shown in the Z-direction from Figure 5.5 above is the resultant force produced by the rotors. Since the rotors remain in the body frame, no conversion in needed, thus the force from the propellers is shown in Equation 5.17 below.

$$\underline{F}^{b}_{rotors} = -b\sum_{i=1}^{6}\Omega_i^2 \tag{5.17}$$

where $b\,[Ns^2]$ is the resulting thrust force as presented in the *Propeller Drag and Thrust Factors* section of this work. Authors of [20, 29] present many other forces in the x and y directions, however, many have suggested that these results are unimportant other than

furthering the completeness of the model because of their relatively small effect on the system. Also, our assumptions have allowed us to effectively say that the drag forces on the vehicle are negligible since the translational velocities will be low. Therefore, we can conclude that the result of Equation 5.13 with the substitution from the knowledge of Equations 5.16 and 5.17 yields

$$\underline{F}^b_{gravity} + \underline{F}^b_{rotors} = m\left(\begin{bmatrix} -vr + wq \\ ur - pw \\ -uq + vp \end{bmatrix} + \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix}\right) \tag{5.18}$$

Next, we are to examine the second half of establishing the EOM for the system by analyzing the general moment equations for rigid body motion known as Euler's equations.

5.2.2 Moments

Continuing to analyze motion for a mass system, we know that resulting moment about the mass center of the external forces on the body is equal to the time rate of change of the angular momentum of the body about the mass center $\sum M_n = \dot{H}_n = \mathbf{I_n}\dot{\omega}_n = \mathbf{I_n}\underline{\alpha}_n$. Therefore let us establish this equation in terms of the local-frame as shown below in Equation 5.19 below.

$$\sum M_n = \mathbf{I_n}\underline{\alpha}_n \tag{5.19}$$

The next step is to realize that we desire to derive the EOM relative to the body frame as this is the frame where measurements of the IMU are taken, and the commands of the control system are given. Therefore, we desire to relate $\underline{\alpha}_n$ to $\underline{\alpha}_b$ and $\sum M_n$ to $\sum M_b$.

$$\underline{\alpha}_n = \ddot{\Theta} \tag{5.20}$$

where $\Theta = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$ is defined as the generalized attitude presented as Euler angles. However, these definitions are still in the local frame, and we desire a relation in the body frame. Therefore, let us define a transformation matrix $T(\phi, \theta, \varphi)\, \epsilon \mathbb{R}^{3x3}$ that transforms ω to $\dot{\Theta}$ as defined below in Equation 5.21.

$$\dot{\Theta} = T(\phi, \theta, \varphi)\,\omega \;\therefore\; \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = T(\phi, \theta, \varphi) \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{5.21}$$

where $\dot{\Theta}$ is referred to as the angular velocity vector in the local frame. It can be noted that this definition matches the transformation presented in Equation 4.32 and reiterated in the Equation below for completeness.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & sin(\phi)tan(\theta) & cos(\phi)tan(\theta) \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & \frac{sin(\phi)}{cos(\theta)} & \frac{cos(\phi)}{cos(\theta)} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

However, we desire the second derivative of the generalized attitude vector in the local frame, therefore, taking the derivative of the equation above yields the desired results and is presented in the Equation below.

$$\ddot{\Theta} = T\dot{(\phi, \theta, \varphi)}\,\omega \tag{5.22}$$

Therefore, we can distribute the differential to each of the components on the right hand side of the equation as seen in Equation 5.23.

$$\therefore \underline{\alpha}_n = T\dot{(\phi, \theta, \varphi)}\,\omega \Rightarrow \underline{\alpha}_n = \dot{T}(\phi, \theta, \varphi)\,\omega + T(\phi, \theta, \varphi)\,\dot{\omega} \tag{5.23}$$

From here, the right hand side of Equation 5.19 has been transformed from the local frame to the body frame. Similarly, we transform the left side of Equation 5.19 into the body frame as shown in the Equation below. Substituting the result of Equations 5.23 and 5.24 into Equation 5.19 yields Equation 5.25 below.

$$\sum M_n = \sum T(\phi, \theta, \varphi)\, M_b \tag{5.24}$$

$$\sum T(\phi, \theta, \varphi)\, M_b = \mathbf{I_n}\left(\dot{T}(\phi, \theta, \varphi)\,\omega + T(\phi, \theta, \varphi)\,\dot{\omega}\right) \tag{5.25}$$

The next step is to continue analyzing the right hand side of the Equation above. First, we can see that the time derivative of the rotation matrix and collecting like terms will yield

$$\sum M_b = \mathbf{I_n}\dot{\omega} + \omega^x \mathbf{I_n}\omega \tag{5.26}$$

Substituting the appropriate terms into Equation 5.26 yields

$$\sum M_b = \mathbf{I_n} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix}^x \mathbf{I_n} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{5.27}$$

where $\mathbf{I_n}$ represents the inertia tensor of the vehicle defined from above. Substituting this knowledge in Equation 5.27 yields

$$\sum M_b = \begin{bmatrix} I_{xx}\dot{p} \\ I_{yy}\dot{q} \\ I_{zz}\dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix}^x \begin{bmatrix} I_{xx}p \\ I_{yy}q \\ I_{zz}r \end{bmatrix} \tag{5.28}$$

Performing the cross product multiplication from above yields

$$\sum M_b = \begin{bmatrix} I_{xx}\dot{p} \\ I_{yy}\dot{q} \\ I_{zz}\dot{r} \end{bmatrix} + \begin{bmatrix} (I_{zz} - I_{yy})\, qr \\ (I_{xx} - I_{zz})\, pr \\ (I_{yy} - I_{xx})\, qp \end{bmatrix} \tag{5.29}$$

Thus, Equation 5.29 completes the derivation of the right-hand side of the moment equation which demonstrates the torques on the HexaCopter platform causing the angular acceleration of the vehicle. As in a similar process to the force derivation equations from the prior section, we now must examine the left hand side of the equation from above. The two most prevalent torques acting on a multirotor platform are the torques induced by the resultant thrust vector for a pitch or roll maneuver and the gyroscopic moments as a result of the propellers rotation. Examining the moments caused by a roll and pitch maneuver yield Equations 5.30 and 5.31 below.

$$\Gamma_{roll} = b\left(\left(\Omega_4^2 + \Omega_6^2\right) l_x + l\Omega_5^2 - \left(\Omega_1^2 + \Omega_3^2\right) l_x - l\Omega_2^2\right) \tag{5.30}$$

$$\Gamma_{pitch} = b\left(\left(\Omega_3^2 + \Omega_4^2\right) l_y - \left(\Omega_1^2 + \Omega_6^2\right) l_y\right) \tag{5.31}$$

where b represents the equivalent thrust factor and l_y,l_x, and l is defined as

$$l_x = 13 \sin\left(30^o\right) * \frac{0.0254m}{1in} \approx 0.165m$$

$$l_y = 13 \cos\left(30^o\right) * \frac{0.0254m}{1in} \approx 0.286m$$

$$l = 13 * \frac{0.0254m}{1in} \approx 0.33m$$

In order for the aircraft to yaw, we rely on the inherit air resistance the propellers encounter when rotating. This results in an overall torque on the system which has an orientation opposite to the direction of increased rotation. Equation 5.32 below defines the resulting torque about the z-axis as a result of a yaw maneuver.

$$\Gamma_{rot-drag} = \left(d\left(\Omega_1^2 + \Omega_3^2 + \Omega_5^2 - \Omega_2^2 - \Omega_4^2 - \Omega_6^2\right)\right) \tag{5.32}$$

where d is the drag of the propeller. Another torque to examine is the inclusion of the torque caused by the rotors. Since each of the rotors are either spinning in a CW or CCW direction at varying speeds, this imbalance causes a gyroscopic effect on the vehicle. Also, when the vehicle is moving with angular velocities ω, the system experiences a torque about the CG of the HexaCopter. This equation accounts for the orientation of the torque as this is directly dependent on the direction of angular velocity and propeller direction. This concept is presented in Equation 5.33 below.

$$\Gamma_{prop-gyro} = \sum_{i=1}^{6} I_{rotor}\left(\omega^x \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T\right)\left(\Omega_i\right)\left(-1^{i+1}\right) \tag{5.33}$$

$$= \sum_{i=1}^{6} I_{rotor} \begin{bmatrix} -q \\ p \\ 0 \end{bmatrix} \left(\Omega_i\right)\left(-1^{i+1}\right)$$

where$\underline{\Omega}\epsilon\mathbb{R}^{6x1}$ is a vector representing the angular speed of the propellers defined as $\underline{\Omega} = \begin{bmatrix} \Omega_1 & \Omega_2 & \Omega_3 & \Omega_4 & \Omega_5 & \Omega_6 \end{bmatrix}^T$ and I_{rotor} is inertia due to the rotor assembly as defined later in this work.

Authors of [16, 17, 20] present other torques that have been included in their model such as

- Torque caused by the difference in air velocity over the propeller between the attacking and receding blade. This scenario is only valid when the helicopter exhibits a velocity in the x-y plane.
- Torque caused by the change in velocity of the rotors.
- Hub moment and inbalances as a result of translational flight.

These extra torques only increase the fidelity of the model, however, the authors of [16, 20] suggest that these effects are minimal in the flight regime considered for this vehicle, and also have minimal effect in the control performance between models that include these factors and models and those that do not. However, these recommendations were made with considerations of the flight regime considered, if high performance maneuvers are desired then the recommendation would be to include these factors as their effects will be increased beyond those results tested by the aforementioned authors.

Revisiting Equation 5.29 and substituting the knowledge of Equations 5.30,5.31,5.32, and 5.33 we can see that

$$\Gamma_{roll} + \Gamma_{pitch} + \Gamma_{rot-drag} + \Gamma_{prop-gyro} = \begin{bmatrix} I_{xx}\dot{p} \\ I_{yy}\dot{q} \\ I_{zz}\dot{r} \end{bmatrix} + \begin{bmatrix} (I_{zz} - I_{yy})\, qr \\ (I_{xx} - I_{zz})\, pr \\ (I_{yy} - I_{xx})\, qp \end{bmatrix} \tag{5.34}$$

Next we will examine the result of the governing equations of motion for this system.

5.2.3 Governing EOM Result

We can see that Equations 5.18 and 5.34 make up a nonlinear dynamic model of the Hexa-Copter. However, it is desired to put the EOM into its standard form while collecting like terms and simplifying the equations. First, let us redefine and collect the resulting propeller rotational speeds as a control vector$\underline{U}\epsilon\mathbb{R}^{4x1}$ defined in Equation 5.35 below.

$$\underline{U} = \begin{bmatrix} -b\sum_{i=1}^{6}\Omega_i^2 \\ b\left(\left(\Omega_4^2+\Omega_6^2\right)l_x + l\Omega_5^2 - \left(\Omega_1^2+\Omega_3^2\right)l_x - l\Omega_2^2\right) \\ b\left(\left(\Omega_3^2+\Omega_4^2\right)l_y - \left(\Omega_1^2+\Omega_6^2\right)l_y\right) \\ d\left(\Omega_1^2+\Omega_3^2+\Omega_5^2-\Omega_2^2-\Omega_4^2-\Omega_6^2\right) \end{bmatrix} = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \tag{5.35}$$

Substituting this knowledge into Equations 5.18 and 5.34 yields the two equations respectively below.

$$\begin{bmatrix} -mg\sin(\theta) \\ mg\sin(\phi)\cos(\theta) \\ mg\cos(\phi)\cos(\theta) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix} = m\left(\begin{bmatrix} -vr+wq \\ ur-pw \\ -uq+vp \end{bmatrix} + \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix}\right) \tag{5.36}$$

$$\begin{bmatrix} U_2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ U_3 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ U_4 \end{bmatrix} + I_{rotor}\begin{bmatrix} -q & q & -q & q & -q & q \\ p & -p & p & -p & p & -p \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\underline{\Omega} = \begin{bmatrix} I_{xx}\dot{p} \\ I_{yy}\dot{q} \\ I_{zz}\dot{r} \end{bmatrix} + \begin{bmatrix} (I_{zz}-I_{yy})qr \\ (I_{xx}-I_{zz})pr \\ (I_{yy}-I_{xx})qp \end{bmatrix} \tag{5.37}$$

Distributing terms, collecting vectors, and putting the equations into standard form yields the resulting nonlinear differential equation that governs the aircraft motion below in Equation 5.38. Constants that are used in this equation are discussed in a section to follow, and implementation of this mathematical equation is described and highlighted in its respective section to follow.

$$\begin{cases} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} -g\sin(\theta) \\ g\sin(\phi)\cos(\theta) \\ g\cos(\phi)\cos(\theta) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ U_1/m \end{bmatrix} + \begin{bmatrix} vr-wq \\ pw-ur \\ uq-vp \end{bmatrix} \\ \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \dfrac{(-I_{zz}+I_{yy})qr}{I_{xx}} \\ \dfrac{(-I_{xx}+I_{zz})pr}{I_{yy}} \\ \dfrac{(-I_{yy}+I_{xx})qp}{I_{zz}} \end{bmatrix} + \begin{bmatrix} U_2/I_{xx} \\ U_3/I_{yy} \\ U_4/I_{zz} \end{bmatrix} + I_{rotor}\left(\begin{bmatrix} -q & q & -q & q & -q & q \\ p & -p & p & -p & p & -p \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\underline{\Omega}\right) \odot \begin{bmatrix} I_{xx}^{-1} \\ I_{yy}^{-1} \\ I_{zz}^{-1} \end{bmatrix} \end{cases} \tag{5.38}$$

5.3 Measurement Model

The following measurement equations for the accelerometer, gyroscope, and magnetometer are similar from the Kalman filter section of the report. However, now we are trying to simulate the input the sensors would see so that this section of our model can output realistic measurements to that of the MEMS devices contained on-board the IMU. Equations 5.39-5.41 below simulate the measurements that the MEMS devices would be sending to the processor for the Kalman Filter to estimate the attitude of the aircraft.

$$\begin{bmatrix} A_{xsim} \\ A_{ysim} \\ A_{zsim} \end{bmatrix} = \begin{bmatrix} g\sin(\theta) \\ -g\sin(\phi)\cos(\theta) \\ -g\sin(\phi)\cos(\theta) \end{bmatrix} + \begin{bmatrix} -vr + wq \\ ur - pw \\ -uq + vp \end{bmatrix} + \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \underline{Accel}_{noise} Note: The right-hand side of this \tag{5.39}$$

$$\begin{bmatrix} p_{sim} \\ q_{sim} \\ r_{sim} \end{bmatrix} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \underline{Gyro}_{bias} + \underline{Gyro}_{noise} \tag{5.40}$$

$$\begin{bmatrix} Mag_{mx} \\ Mag_{my} \\ Mag_{mz} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ \sin(\phi)\sin(\theta) & \cos(\phi) & \cos(\theta)\sin(\phi) \\ \cos(\phi)\sin(\theta) & -\sin(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix} \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -\cos(i)\cos(d) \\ -\cos(i)\sin(d) \\ \sin(i) \end{bmatrix} \tag{5.41}$$

5.4 Motor Model

This section will cover the work done on modeling the propeller and motor assembly to determine factors such as the dependence of the PWM signal on thrust, RPM, and power capabilities of the motor.

Note: This section presents the results in RPM as this is a more intuitive unit for the propeller's angular speed. However, the implementation in the model contains conversions to handle the RPM incidences to conventional SI units of rad/sec to keep notation handling

consistent throughout the model.

5.4.1 Test Stand for Determining Motor Response Properties

A test stand was developed to determine the characteristic response for the motor and determine other properties during flight such as power, thrust capabilities, and to determine if there exists potential dead-zone and saturation regions of the motor. The test stand developed had consisted of several pieces of hardware and code to measure these characteristics and the graphic below in Figure 5.6 describes the test setup.

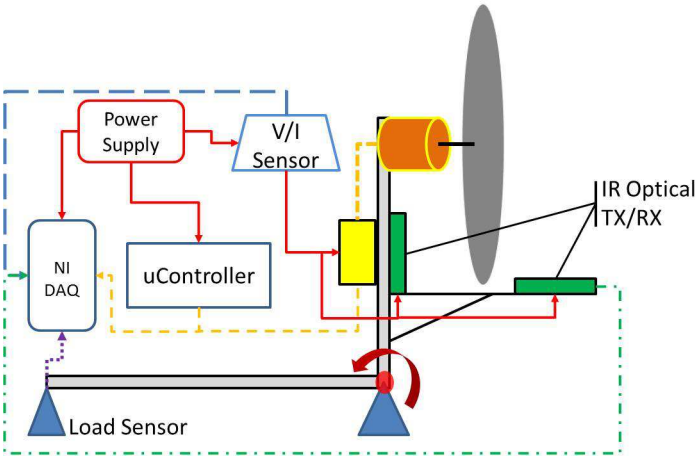


Figure 5.6: Example diagram detailing the test setup for determining the motor's thrust capability along with power consumption. Information such as current draw, supply voltage, triggering sequence, and commanded PWM signal were recorded through an NI DAQ.

The test stand was an automated process produced with both LabView and C++ code. The test procedure began by an operator selecting the start button on the LabView user interface displayed on a host computer. The microcontroller unit would then receive a start command and begin its process of stepping through incremental PWM values to the ESC. At each iteration of PWM signal, the microcontroller would send a trigger unit to the DAQ indicating for the unit to begin collecting data, and then issue the change in PWM command to the ESC controlling the motor. The NI DAQ unit would then collect 32k samples at a rate of 4kHz. Samples for collection included voltage, current, force, RPM, and the trigger line for later correspondence of data. The process was then repeated until the amperage of the ESC unit was reached, the motor wattage limits were reached, or the PWM values to

the ESC were within a dead-band or saturation region. The RPM was calculated based on the output of the IR optical unit developed for this work; schematic designs are shown in Appendix 1.

It was during this process that the initial proposed 12x4.7 APC propeller caused the motor to saturate its usable wattage rating before the end of the usable control signal band. At this point without replacing the motor the options were to 1)Saturate the control throws 2)Operate the vehicle on a lower battery voltage 3)Decrease the propeller size. The second solution is not a plausible one since as discussed before, some the electrical systems on-board the vehicle require these intended voltage levels. Also, we would like to achieve the maximum bandwidth of our control resolution on the PWM signal, therefore, it is generally not advisable to saturate your control throws to a very limited region of achievability. Thus, the solution was to lower the propeller size to a 11x4.7 APC propeller, this allowed the majority of the usable PWM signal to be achieved before saturating the motor wattage. Upon completion of this test, power consumptions based on this test validated the proposed flight durations from the prior sections and reaffirmed the possible lift capabilities of the vehicle. At this point, we desire to further characterize the angular velocity of the propeller to the thrust. A sample data output of the RPM sensor is shown below in Figure 5.7.

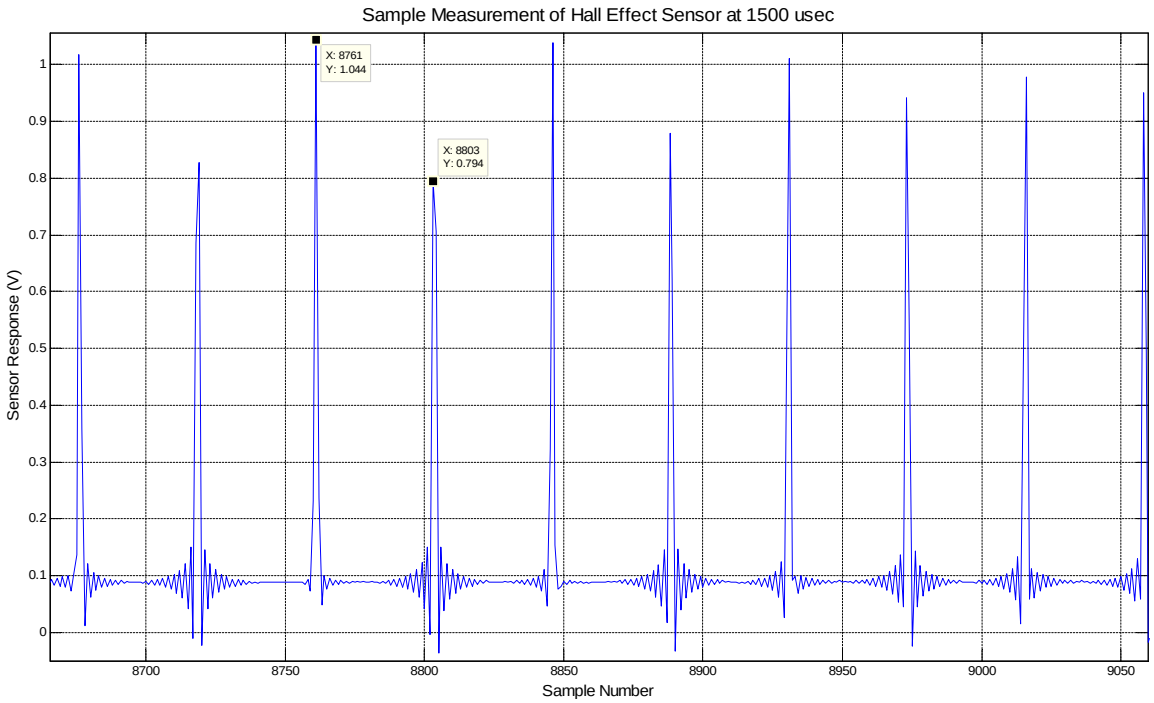


Figure 5.7: Example RPM sensor output as read by NIDAQ. The RPM of the propeller can be averaged over sequential measurements with knowledge of the f_s.

It can be seen that RPM can be calculated between pairs of sequential peaks by

$$RPM = \left(\frac{Number\ of\ samples\ between\ peaks}{Sampling\ rate} \right)^{-1} \left({}^{60s}/_{min} \right)$$

Each test at a new command signal resulted in 32K data points, therefore, we can calculate the propellers mean RPM and standard deviation over each data set collected at a commanded PWM signal. From here, we can develop a function describing the relationship between the commanded microsecond signal and RPM as shown in Equation 5.42 and highlighted below in Figure 5.8.

$$\mu Second_{PWM} = \left(RPM^2\right) 1.0034E-5 + (RPM)\, 0.0058 + 1132 \quad for\ 1400 \leq RPM \leq 7590 \tag{5.42}$$

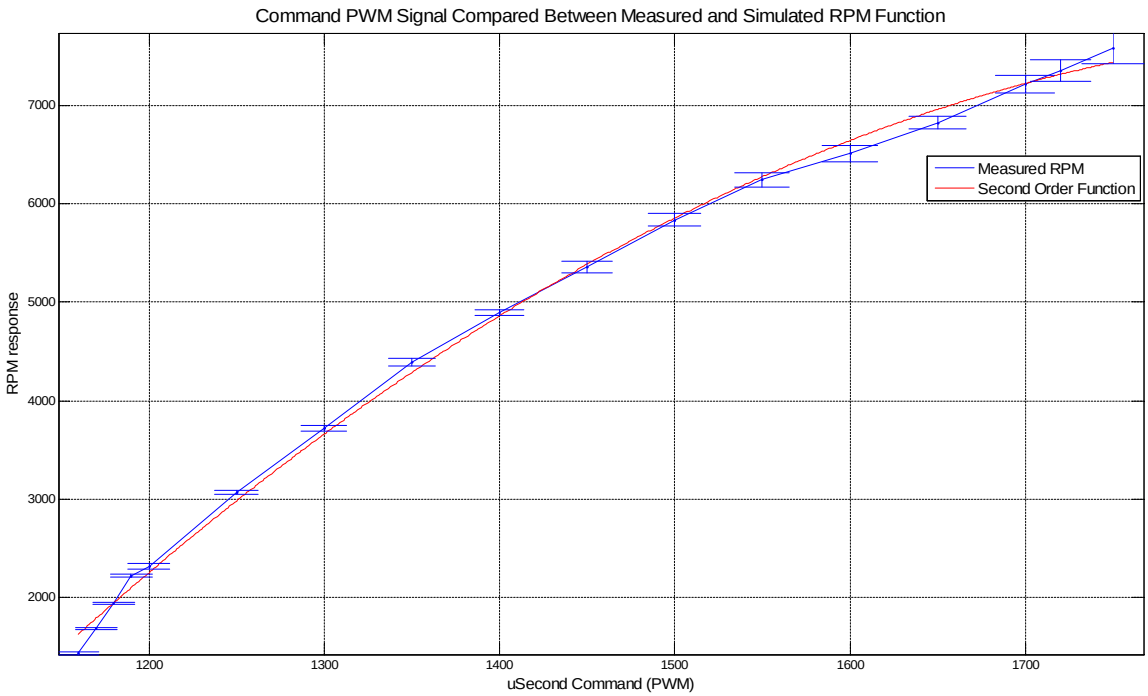


Figure 5.8: This figure shows the RPM response of the motor with coordinating PWM (microsecond) command from the flight computer.

Next, it was desired to analyze the resulting thrust from the rotor as a function of the RPM. It is described in a wide variety of literature that the resulting thrust from a rotor is typically a function of the angular speed of the propeller squared. It can seen that this trend holds true and can be seen from Figure 5.9 below.

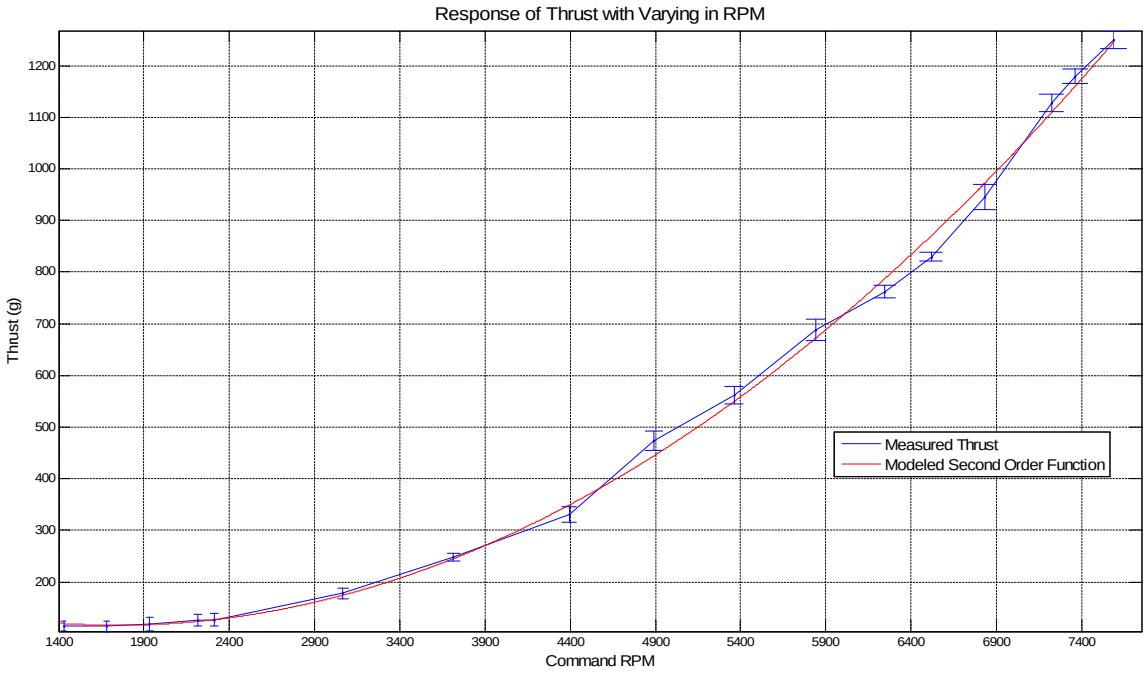


Figure 5.9: This graph shows the thrust response of the motors as a function of the commanded RPM speed.

To describe the second order response, a function was based on a second order polynomial fit of the data as shown in red in the figure from above and presented as Equation 5.43 below. It can also be seen that the standard deviation (represented as error bars) shown in the figure are small in magnitude. This is a result of a low-pass Butterworth filter applied to the load cell data. The original data contained high frequency noise as a result of the beam vibrating on the load cell and therefore caused large spikes of high frequency content of the data. Since this data compares well with other data information presented on this propeller, it is assumed that the second order transfer function presented will accurately describe the resulting thrust from a given rotor RPM.

$$Thrust(kg) \approx \frac{(RPM^2)\,3.304E-5-(RPM)\,0.115+215.51}{1000} \quad for\ 1400 \leq RPM \leq 7590 \tag{5.43}$$

From here, it was desired that we model the motor assembly further for implementation in the simulated environment. Many authors [16, 17, 20, 30] continue to model the DC brushless motor with a high order transfer function by characterizing such electrical properties of the

motor to include the impedance and inductance of the armature windings, friction of the motor, back EMF forces, and many other parameters depending on the desired fidelity of the motor model. However, for this model, it is desired that the we understand the relationship of the input to the motor and the response of the motor RPM value generating our thrust. The author of [48] had presented such detailed work in modeling the motor assembly as stated above, however, the author had concluded in his work with experimental comparisons that once the equations were simplified to a second order transfer function, that the modeled response function could be simplified further to a first order transfer function in response to a step input. Similar work was also presented in [20] . Thus we know the first order transfer function will take on the form of

$$MR(s) = \frac{K}{\tau s + 1}$$

Therefore, a test case was created to measure this data and the response was recorded and is shown in Figure 5.10 below. Parameters such as the constant gain term and the time constant (τ) were estimated through the final steady state response value and the rise time of the response respectively. This resulted in Equation 5.44 below and the accompanying Figure 5.10 highlighting the response of Equation 5.44 to the same step input of the test case.

$$MR_{C-Response}(s) = \frac{0.99}{0.12175s + 1} \tag{5.44}$$

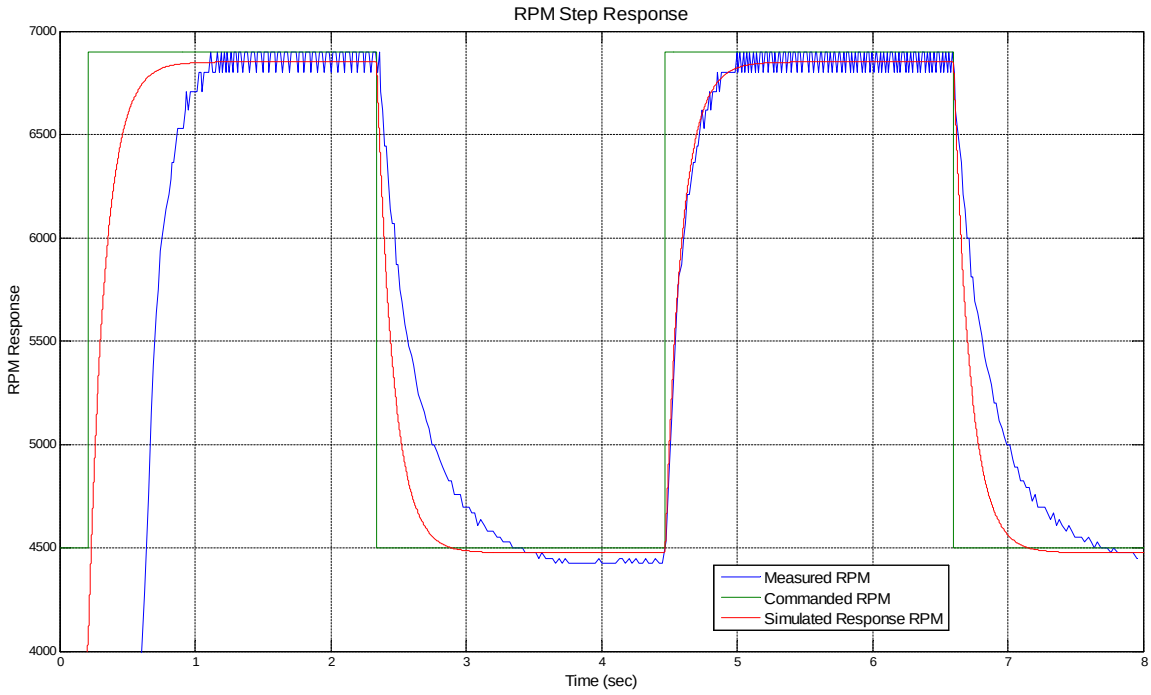


Figure 5.10: This figure highlights the RPM step response of the actual motor(blue) and simulated motor(red) in response to a step input(green). It is noted that the initial step response does not respond as well because of unmodeled nonlinearities in the motor when responding in start up. However, the region of performance from the aircraft is within the response region as shown in the second step response.

It can be seen from the figure above that the modeled initial step response as compared to the measured response does not coincide as well with each other as does the second step response. This is because the first step response was from the motor starting with zero initial speed. Therefore, the nonlinearities such as motor windup, EMF, armature inductance, etc play a large role in effecting the response of the motor. However, during flight, the aircraft should never have to see control throws with such a significant range as the assumption for flight is low velocities and hovering conditions. Examining the second step response yields a more realistic scenario to the proposed flight conditions and shows that the modeled response coincides well with the measured response.

5.4.2 Approximating Rotor/Motor Assembly Inertia

The rotor assembly causing inertia changes about the motor can be broken down into the following three pieces each with their own inertial properties: 1) propeller mount 2) propeller 3) stator. First, the propeller mount can be approximated to be a cylinder rotating about its axis such that the moment of inertia yields

$$I_{pmount} = .5mr^2 = .5 * 0.006kg * (0.0075m)^2 \approx 1.6875E-7 \left[kgm^2\right] \tag{5.45}$$

The propeller's moment of inertia can be approximated as a rotating rod such that

$$I_{propeller} = \frac{1}{12}ml^2 = \frac{1}{12}(0.014kg)\left[11in * \frac{1m}{39.37in}\right]^2 \approx 9.1075E-5 \left[kgm^2\right] \tag{5.46}$$

Lastly, the stators moment of inertia can be calculated to be the moment of inertia of a thin ringed cylinder

$$I_{stator} = mr^2 \approx 0.019kg * (0.014m)^2 \approx 3.7204E-6 \left[kgm^2\right] \tag{5.47}$$

Thus, summing Equations 5.45-5.47 yields an approximation of the rotor's moment of inertia as shown in the equation below.

$$I_{rotor} = I_{pmount} + I_{propeller} + I_{stator} \approx 9.4964E-5 \left[kgm^2\right] \tag{5.48}$$

5.4.3 Propeller Drag and Thrust Factors

First, we know from other works [29, 30] that the thrust factor ($b\,units\,of\,Ns^2$) can be determined through the conservation of momentum theory or blade element theory. Works of [29] had concluded that both techniques produced viably similar results that both be accurate for use in a simulation environment, therefore, this work had chosen to employ the conservation of momentum approach for its conciseness and ease. Thus, the thrust factor in a hovering scenario is the weight carried by an individual propeller divided by the speed of the propeller squared, as seen in Equation 5.49 below.

$$b_{TFactor} = \frac{(vehicle_{mass}/6)}{\Omega_{hover}^2} \approx \frac{(35.91N)}{6\left(587.47^{rad}/_{sec}\right)^2} \approx 1.73E-5\left[Ns^2\right] \tag{5.49}$$

Lastly, the drag factor is required to describe torque caused by the angular speed of the propeller producing a rotation about the z-axis of the vehicle. [30] details a derivation of the drag factor using blade element theory. The equation derived in his work is cited here for completeness. The drag factor is defined as

$$d_{DFactor} = N_B \rho_A c R_p^4 \left(\frac{C_D}{8} + a\lambda\left(\frac{\theta_{I_a}}{6} - \frac{\theta_{I_w}}{8} - \frac{\lambda}{4}\right)\right) \approx 3.47E-7\left[Nms^2\right] \tag{5.50}$$

where N_B is the number of blades per propeller, $\rho_A\left[^{kg}/_{m^3}\right]$ is the density of air, $c\,[m]$ is the average chord, $R_p$is the propeller radius, C_D is the drag coefficient, $a[rad^{-1}]$ is 2π, λ is the inflow ratio, $\theta_{I_a}[rad]$ is the zero angle of incidence, and lastly $\theta_{I_w}[rad]$ is the twist angle of incidence. Currently, UIUC [50] maintains a propeller database with much of this information available at varying operating RPMs. For this work, we will assume that these factors are constant and use the data associated with the flight RPM of the UAV. This concludes the various modeling parameters associated with the motor and propeller assembly.

5.5 Implementation of Dynamics in SIMULINK Model

Lastly, it is important to examine our available state vector for our model and how to incorporate the knowledge of our system from above. It is known that we can rewrite our system into state-space form as $\dot{\chi} = f(\chi, U)$ where U defines our inputs and $\chi_{model} \epsilon \mathbb{R}^{12x1}$. For this work we have chosen our state vector to be

$$\chi_{model} = \begin{bmatrix} \upsilon & \omega & \xi & P_n \end{bmatrix}^T = \begin{bmatrix} u & v & w & p & q & r & \phi & \theta & \psi & x & y & z \end{bmatrix}^T \tag{5.51}$$

with its derivative defined as

$$\dot{\chi}_{model} = \begin{bmatrix} \dot{\upsilon} & \dot{\omega} & \dot{\xi} & \dot{P}_n \end{bmatrix}^T = \begin{bmatrix} \dot{u} & \dot{v} & \dot{w} & \dot{p} & \dot{q} & \dot{r} & \dot{\phi} & \dot{\theta} & \dot{\psi} & \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T \tag{5.52}$$

and U defined from Equation 5.35 above. It is important to realize that many works deviate

from each other at this point in their definition of the state vector. Some make assumptions that the transformation matrix of Equation 4.13 is unity between the Euler rates and the body angular velocities if deviations from a hover are small [20, 29]. While other models [30] do not keep track of the vehicle position in simulation as they have decided their work will not consider an outer-loop positioning controller. Collecting the knowledge from above, we can rewrite our nonlinear function $f(\chi, U)$ as

$$
\begin{cases}
\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} -g\sin(\theta) \\ g\sin(\phi)\cos(\theta) \\ g\cos(\phi)\cos(\theta) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ U_1/m \end{bmatrix} + \begin{bmatrix} vr - wq \\ pw - ur \\ uq - vp \end{bmatrix} \\
\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \dfrac{(-I_{zz} + I_{yy})\, qr}{I_{xx}} \\ \dfrac{(-I_{xx} + I_{zz})\, pr}{I_{yy}} \\ \dfrac{(-I_{yy} + I_{xx})\, qp}{I_{zz}} \end{bmatrix} + \begin{bmatrix} U_2/I_{xx} \\ U_3/I_{yy} \\ U_4/I_{zz} \end{bmatrix} + I_{rotor} \left(\begin{bmatrix} -q & q & -q & q & -q & q \\ p & -p & p & -p & p & -p \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \underline{\Omega} \right) \odot \begin{bmatrix} I_{xx}^{-1} \\ I_{yy}^{-1} \\ I_{zz}^{-1} \end{bmatrix} \\
\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & sin(\phi)tan(\theta) & cos(\phi)tan(\theta) \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & \frac{sin(\phi)}{cos(\theta)} & \frac{cos(\phi)}{cos(\theta)} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \\
\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} c\theta c\psi & -c\phi s\psi + s\phi s\theta c\psi & s\phi s\psi + c\phi s\theta c\psi \\ c\theta s\psi & c\phi c\psi + s\phi s\theta s\psi & -s\phi c\psi + c\phi s\theta s\psi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}
\end{cases}
\tag{5.53}
$$

The next chapter discusses the incorporation and implementation of the EOMs from above and other dynamic limitations of the system into a full model in Matlab/SIMULINK. The full SIMULINK model is presented in Appendix C for reference. Some example subfunctions and and script files are also called out as examples from the model in Appendices C and D.

Chapter 6

Multirotor Nonlinear Control & Simulation

6.1 Lyapunov Theory

Often in control theory, and as followed in this work, the analysis implies that the equilibrium point is at the origin of $\mathbb{R}^n$; that is, $\overline{x} = 0$. However, we know that there is no loss in generality as the equilibrium point can be moved from the origin to any desired point simply through a change of variables [43]. For this work, stability will be assessed based on the application of Lyapunov's second method and Lasalle's invariance theorem, while the flight control system is derived based on conditions that are derived from Lyapunov function requirements. It should be noted that in this theory section, the variable x can be of scalar or vector size.

For Lyapunov theory, it is important to recognize the definitions of the implied definiteness of a proposed Lyapunov function. Therefore, we cite these definitions for completeness.

A scalar function, $V(x)$, is said to be respectively positive definite or positive semi-definite if $V(0) = 0$ and

$$V(x) \succ 0, \forall x \neq 0 \quad \& \quad V(x) \succeq 0, \forall x \neq 0$$

A scalar function, $V(x)$, is said to be respectively negative definite or negative semi definite if $V(0) = 0$ and

$$V(x) \prec 0,\ \forall x \neq 0 \quad \& \quad V(x) \preceq 0,\ \forall x \neq 0$$

A scalar function, $V(x)$, is said to be radially unbounded if

$$V(x) \to \infty \, as \, \|x\| \to \infty$$

First, we define Lyapunov's stability theorem as (Theorem 4.1 of [43]):

Let x=0 be an equilibrium point for $\dot{x} = f(x)$ and $D \subset \mathbb{R}^n$ *be a domain containing x=0. Let* $V : D \to R$ *be a continuously differentiable function such that*

$$V(0) = 0 \,\&\, V(x) > 0 \, in \, D - \{0\} \therefore \dot{V}(x) \leq 0 \, in \, D \tag{6.1}$$

then x=0 is stable. However, we can conclude asymptotic stability (AS) if

$$\dot{V}(x) < 0 \, in \, D - \{0\} \tag{6.2}$$

However, it should be noted that this theorem is only a sufficient condition and not a necessary one. This implies that if the proposed Lyapunov function, $V(x)$, is not able to satisfy these conditions, we cannot conclude instability of the equilibrium. All that can be said is that the Lyapunov candidate cannot assess the stability of the equilibrium, however, another proposed candidate may. A stronger condition of global asymptotic stability (GAS) can be concluded if

$$V(x) > 0 \,\forall x \neq 0 \,\&\, \dot{V}(x) < 0 \,\forall x \neq 0 \,\&\, V(x) \to \infty \, as \, \|x\| \to \infty \tag{6.3}$$

then the origin is globally asymptotically stable. Lasalle's theorem has several corollaries and lemmas that support the theorem that should be recognized first. First, if $V(x) > 0$ and $\dot{V}(x) \leq 0$ in some domain $D \subseteq \mathbb{R}^n$, then the set of $\Omega_c = \{x \in D \mid V(x) \leq c\}$ is a positively invariant set since V must increase in order for x to leave. A positively invariant set is accordingly defined as

$$x(0) \in M \Rightarrow x(t) \in M \forall t \geq 0.$$

Let $\Omega \subset D$ be a compact set that is positively invariant with respect to $\dot{x} = f(x)$. Let $V : D \to R$ be a continuously differentiable function such that $\dot{V}(x) \leq 0$ in Ω. Let E be the set of all points in Ω where $\dot{V}(x) = 0$. Let M be the largest invariant set in E. Then every solution starting in Ω approaches M as $t \to \infty$. The following control systems in this work are considered autonomous systems; that is $\dot{x} = f(x)$ has not explicit dependence on time such as $\dot{x} = f(x, t)$, therefore the theorems above apply and it is not necessary to introduce Barbalat's theory for non-autonomous systems. Further theory of nonlinear control along with detailed explanations and examples can be found in Khalil [43].

6.2 Backstepping Control Theory and Primer

Backstepping is a technique that has been used to control aircraft in the works of [20, 44, 45, 46, 47] because of its ability to handle the inherent cascaded structure of system that typically describes the aircraft dynamic motion. The general Backstepping approach is presented in [43] and is presented here for clarity of the modified control scheme presented in this work. First consider the system as described in Equation 6.4 below and detailed graphically in Figure 6.1.

$$\begin{aligned} \dot{\eta} &= f(\eta) + g(\eta)\xi \\ \dot{\xi} &= u \end{aligned} \tag{6.4}$$

where $\left[\eta^T, \xi\right]^T \epsilon R^{n+1}$ is the state and $u \epsilon R$ is the control input. The functions of $f : D \to R^n$ and $g : D \to R^n$ are smooth in a domain $D \subset R^n$ that contains $\eta = 0$ and $f(0) = 0$.

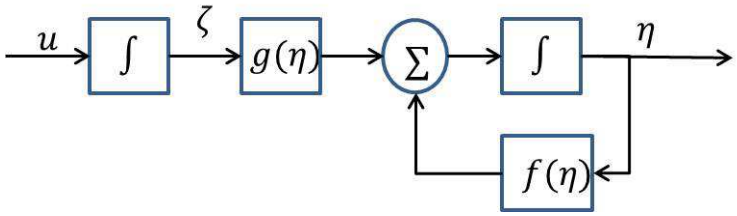


Figure 6.1: This figure details the cascaded connection between the input and the state of the presented nonlinear equation of 6.4.

As noted before, we desire to design a control input u that stabilizes the origin located

at $\eta = 0, \xi = 0$. Let us suppose that we can choose a smooth state feedback control law $\xi = \phi(\eta)$ where $\phi(0) = 0$ that asymptotically stabilizes Equation 6.4. Therefore we can define a candidate Lyapunov function $V(z)$ for stability that satisfies

$$\frac{\partial V}{\partial \eta}\left[f(\eta) + g(\eta)\phi(\eta)\right] \leq -W(\eta) \quad \forall \eta \epsilon D \tag{6.5}$$

We can therefore add and subtract the function $g(\eta)\phi(\eta)$ on the right hand side of Equation 6.4 to keep the definition of the cascaded system the same.

$$\begin{aligned} \dot{\eta} &= f(\eta) + g(\eta)\phi(\eta) + g(\eta)\xi - g(\eta)\phi(\eta) \\ \dot{\xi} &= u \end{aligned} \tag{6.6}$$

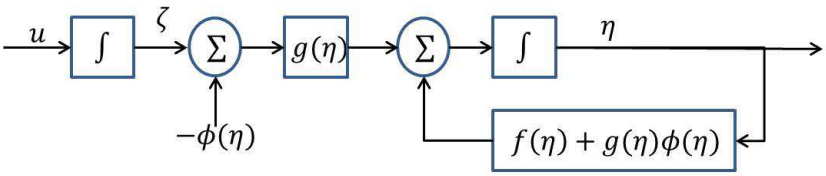


Figure 6.2: This figure highlights the introduction of the state feedback control law of ϕ.

Introducing a change of variables such as

$$z = \xi - \phi(\eta) \tag{6.7}$$

results in

$$\begin{aligned} \dot{\eta} &= f(\eta) + g(\eta)\phi(\eta) + g(\eta)z \\ \dot{z} &= \dot{\xi} - \dot{\phi}(\eta) = u - \dot{\phi}(\eta) \end{aligned} \tag{6.8}$$

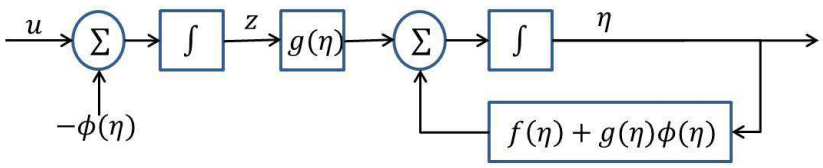


Figure 6.3: This figure highlights the change of variables sequence which results in the $\phi(\eta)$ backstepped through the integrator.

Since the functions f, g, ϕ are known, the derivative $\dot{\phi}$ can be computed as

$$\dot{\phi} = \frac{\partial \phi}{\partial \eta}\left[f(\eta) + g(\eta)\xi\right] \tag{6.9}$$

Letting $\nu = u - \dot{\phi}(\eta)$ reduces Equation 6.8 to Equation

$$\begin{aligned} \dot{\eta} &= f(\eta) + g(\eta)\phi(\eta) + g(\eta)z \\ \dot{z} &= \dot{\xi} - \dot{\phi}(\eta) = \nu \end{aligned} \tag{6.10}$$

Choosing a Lyapunov candidate function such as

$$\begin{aligned} V_c(\eta, \xi) &= V(\eta) + \tfrac{1}{2}z^2 \\ &= V(\eta) + \tfrac{1}{2}\left(\xi - \phi(\eta)\right)^2 \end{aligned} \tag{6.11}$$

$$\dot{V}_c = \frac{\partial V}{\partial \eta}\left[f(\eta) + g(\eta)\phi(\eta)\right] + \frac{\partial V}{\partial \eta}g(\eta)z + zv \leq -W(\eta) + \frac{\partial V}{\partial \eta}g(\eta)z + zv \tag{6.12}$$

Choosing ν as

$$\nu = -\left(\frac{\partial V}{\partial \eta}g(\eta) + kz\right), \; k > 0 \tag{6.13}$$

yields

$$\dot{V}_c \leq -W(\eta) - kz^2 \tag{6.14}$$

which shows that the origin is asymptotically stable, and if the assumptions hold globally

and the Lyapunov function is radially unbounded, the conclusion can be strengthened to globally asymptotically stable.

6.3 Application of Integral Backstepping Control for Attitude/Altitude Control

The next several sections will cover a sample case of deriving a single axis controller. The derivation between each axis controller design is fairly similar until the final steps where the dynamics of the axis are substituted into the equation and the desired controller is formed based on canceling undesired dynamics with desired dynamics. Each derivation and controller presentation concludes with its analytical and numerical Lyapunov analysis ensuring Lyapunov stability for the ideal system.

6.3.1 State analysis and setup

We have seen in the general case that Backstepping is a recursive based control technique that pertains to particularly structured nonlinear system. This cascaded structure allows the architect to develop a stabilizing controller for the inner most subsystem where the control input u is introduced and in theory derive stabilizing controllers for each cascaded level that the controller works through. Our system as defined in 5.38 will not necessarily meet the direct requirements as desired for this system, but rather a form of Block-Backstepping. However, there are measurements that appear in our control inputs that we do not have viable measurements for in our controller. For example, let us examine the pitch control of the vehicle by defining an error for pitch as defined in Equation 6.15.

$$e_\theta = \theta_d - \theta \tag{6.15}$$

It can be easily seen that we desire to reduce the error of the pitch angle between the current measured value and the desired pitch angle. Therefore, From Equation 6.15 it can be seen that to reduce the error, we desire that the slope of the error be represented by $\dot{e}_\theta = -ke_\theta$ where $k > 0$. Accordingly taking the derivative of Equation 6.15 yields

$$\frac{\partial e_\theta}{\partial t} = \dot{\theta}_d - \dot{\theta} \tag{6.16}$$

However, our assumption of a desired pitch angle means that we do not desire an angular velocity, therefore, $\dot{\theta}_d \to 0$. Substituting the knowledge of body angular velocity to Euler rates as defined in Equation 4.32 and adding and subtracting our desired derivative error to the right side of the equation forms our derivative error equation.

$$\frac{\partial e_\theta}{\partial t} = -\dot{\theta} \therefore \frac{\partial e_\theta}{\partial t} = -ke_\theta - \left(q + \frac{ke_\theta - r\sin(\phi)}{\cos(\phi)} \right) \cos(\phi) \tag{6.17}$$

From Equation 6.17 it can be seen that we have a choice between q and r as variables to reduce the equation to our desired form. Since we are considering a pure pitch maneuver, let us use q to gain our desired $\frac{\partial e_\theta}{\partial t}$ by defining q as

$$q = \frac{-ke_\theta + r\sin(\phi)}{\cos(\phi)} \tag{6.18}$$

From Equation 6.18 above, it can be seen that $\frac{\partial e_\theta}{\partial t}$ reduces to $-ke_\theta$. Continuing through the Backstepping procedure we can propose a Lyapunov candidate and respectively a corresponding Lyapunov derivative that ensures a stable system if q can be designed to be to as defined in Equation 6.18. Continuing the process would yield that q has its own dynamics which an error can be defined for and eventually the Backstepping technique will lead to the control input U_3 entering in the equation which can validate Lyapunov stability for the set of control laws. However, this entire derivation will require knowledge of the rotational velocity in the body frame, where the controller currently only has knowledge of the Euler angles. This procedure is referred to as block Backstepping and creates stable control laws but are often larger and complex implementations then desired on a real system. Many authors suggest that a small angle approximation can be made between the Euler rates and body angular velocities such that Equation 4.32 reduces to

$$\begin{bmatrix} \cos(\phi) \\ \sin(\phi) \\ \cos(\theta) \\ \tan(\theta) \end{bmatrix} \approx \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \therefore \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \approx [\mathbf{I_{3x3}}] \begin{bmatrix} p \\ q \\ r \end{bmatrix} \therefore \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} \approx [\mathbf{I_{3x3}}] \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \tag{6.19}$$

This approximation is valid for our work because the system will not be designed for large angular deviations beyond the hover condition and is only designed for low velocity flight profiles. Therefore, let our control state vector be defined as $\chi_{control} \, \epsilon \, \mathbb{R}^{6x1}$.

$$\underline{\chi}_{control} = \begin{bmatrix} \phi & \dot{\phi} & \theta & \dot{\theta} & \psi & \dot{\psi} \end{bmatrix} \tag{6.20}$$

$$\underline{\dot{\chi}}_{control} = \begin{bmatrix} \dot{\phi} & \ddot{\phi} & \dot{\theta} & \ddot{\theta} & \dot{\psi} & \ddot{\psi} \end{bmatrix} \tag{6.21}$$

where $x_1 = \phi$, $x_2 = \dot{x}_1 = \dot{\phi}$ and $x_3 = \theta$, $x_4 = \dot{x}_3 = \dot{\theta}$ and $x_5 = \psi$, $x_6 = \dot{x}_5 = \dot{\psi}$. Thus, using the knowledge from Equations 6.19-6.21 the angular accelerations of the EOM in Equation 5.38 can be rewritten as shown in Equation 6.22 below

$$\begin{cases} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{cases} = \begin{matrix} x_2 \\ \gamma_1 x_4 x_6 + \beta_1 \underline{\Omega} x_4 + {}^{U_2}/_{I_{xx}} \\ x_4 \\ \gamma_2 x_2 x_6 + \beta_2 \underline{\Omega} x_2 + {}^{U_3}/_{I_{yy}} \\ x_6 \\ \gamma_3 x_2 x_4 + {}^{U_4}/_{I_{zz}} \end{matrix} \tag{6.22}$$

where

$$\gamma = \begin{bmatrix} \dfrac{(-I_{zz} + I_{yy})}{I_{xx}} \\ \dfrac{(-I_{xx} + I_{zz})}{I_{yy}} \\ \dfrac{(-I_{yy} + I_{xx})}{I_{zz}} \end{bmatrix} \; \& \; \beta = I_{rotor} \left(\begin{bmatrix} -\dot{\theta} & \dot{\theta} & -\dot{\theta} & \dot{\theta} & -\dot{\theta} & \dot{\theta} \\ \dot{\phi} & -\dot{\phi} & \dot{\phi} & -\dot{\phi} & \dot{\phi} & -\dot{\phi} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \right) \odot \begin{bmatrix} I_{xx}^{-1} \\ I_{yy}^{-1} \\ I_{zz}^{-1} \end{bmatrix}$$

We can see that Equation 6.22 now is in the form we traditionally would desire for the implementation of a Backstepping controller. However, the works of [20, 44, 45, 46, 47] suggest that the Backstepping controller can be strengthened to improve its steady state error performance as seen from linear PID control through the additional of a integral term.

This control approach is referred to in literature as the Integral Backstepping control method. The incorporation of an additional integral error term can be done by augmenting the plant dynamic equation with an additional integral error dynamic term as seen below in Equation 6.23.

$$\begin{aligned} \dot{\varepsilon}_i &= c_1 \left(x_{i_d} - x_i \right) - \alpha_i \varepsilon_i \\ \dot{x}_i &= x_{i+1} \\ \dot{x}_{i+1} &= f\left(\underline{x}, \underline{\Omega} \right) + gu \end{aligned} \tag{6.23}$$

6.3.2 Attitude Control

Equation 6.23 highlights the general form that will be used in the Integral Backstepping control design for the roll, pitch, and yaw control systems. It should also be apparent that there will be two backstepping iterations before the control input U appears in the dynamics of the system. This section will derive the roll controller, however, it can be noticed that the pitch and yaw controllers are derived in the same way through the first several segments, and only deviates from each other in the last steps where the each axis dynamics are substituted. The results of the pitch and yaw controller derivation will be presented at the end of this section without further documentation in this work. To begin, consider the first subsystem of Equation 6.23.

$$\begin{aligned} \dot{\varepsilon}_1 &= c_1 \left(x_{1_d} - x_1 \right) - \alpha_1 \varepsilon_1 \\ \dot{x}_1 &= x_2 \end{aligned} \tag{6.24}$$

where $c_1 > 0$ and $\alpha_1 > 0$. The first equation from above defines our tracking error dynamics between the desired roll and the actual roll of the system. From here, consider x_2 to be the virtual control and define $e_1 = x_{1_d} - x_1$. Following the Backstepping procedure from above, let us propose a candidate Lyapunov function such that

$$V_1 = \frac{1}{2} e_1^2 + \frac{1}{2} \varepsilon_1^2 \tag{6.25}$$

$$\begin{aligned}
\dot{V}_1 &= e_1\dot{e}_1 + \epsilon_1\dot{\epsilon}_1 \\
&= e_1\left(\dot{x}_{1_d} - \dot{x}_1\right) + \epsilon_1\left[c_1\left(x_{1_d} - x_1\right) - \alpha_1\varepsilon_1\right] \\
&= e_1\left(\dot{x}_{1_d} - \dot{x}_1\right) + \epsilon_1\left(c_1e_1 - \alpha_1\varepsilon_1\right) \\
&= e_1\left(\dot{x}_{1_d} - \dot{x}_1 + c_1\epsilon_1\right) - \alpha_1\varepsilon_1^2 \\
&= e_1\left(\dot{x}_{1_d} - x_2 + c_1\epsilon_1\right) - \alpha_1\varepsilon_1^2
\end{aligned} \tag{6.26}$$

Since we consider x_2 our virtual control, we must design x_2 to ensure that our Lyapunov derivative from above is negative semi-definite. Therefore we can define our desired virtual control input as shown in Equation 6.27.

$$x_2 = \dot{x}_{1_d} + c_1\epsilon_1 + c_2e_1 \;\; \therefore \;\; \dot{V}_1 = -c_2e_1^2 - \alpha_1\varepsilon_1^2 \preceq 0 \tag{6.27}$$

for a positive definite constant $c_2 > 0$. However, it is important to realize that the above virtual control is the desired performance of our virtual control x_2. The problem at this stage is that the controller output U has not appeared yet in this virtual control, and this state inherently has its own dynamics that must be accounted for. Therefore, let us establish the error dynamics for state x_2 as shown below in Equation 6.28.

$$e_2 = x_{2_d} - x_2 \tag{6.28}$$

Using this definition we can rewrite $\dot{e}_1$ as

$$\begin{aligned}
\dot{e}_1 &= \dot{x}_{1_d} - \dot{x}_1 \;\&\; x_2 = x_{2_d} - e_2 \\
from\, definition\, x_2 &= \dot{x}_1 \therefore \dot{e}_1 = \dot{x}_{1_d} - x_{2_d} + e_2 \\
&= \dot{x}_{1_d} - \left(\dot{x}_{1_d} + c_1\epsilon_1 + c_2e_1\right) + e_2 \\
&= -c_1\epsilon_1 - c_2e_1 + e_2
\end{aligned} \tag{6.29}$$

Taking the derivative of Equation 6.28

$$\dot{e}_2 = \dot{x}_{2_d} - \dot{x}_2 \tag{6.30}$$

Substituting appropriate terms yields

$$\begin{aligned}\dot{e}_2 &= (\ddot{x}_{1_d} + c_1\dot{\epsilon}_1 + c_2\dot{e}_1) - \dot{x}_2 \\ &= (\ddot{x}_{1_d} + c_1\dot{\epsilon}_1 + c_2\left(-c_1\epsilon_1 - c_2e_1 + e_2\right)) - \dot{x}_2 \\ &= (\ddot{x}_{1_d} + c_1\dot{\epsilon}_1 - c_1\epsilon_1c_2 - e_1c_2^2 + e_2c_2) - \dot{x}_2\end{aligned} \tag{6.31}$$

We can see from the result of Equation 6.31 from above, $\dot{x}_2$ introduces our control input and the dynamics of the vehicle such that

$$\dot{e}_2 = (\ddot{x}_{1_d} + c_1\dot{\epsilon}_1 - c_1\epsilon_1c_2 - e_1c_2^2 + e_2c_2) - (\gamma_1x_4x_6 + \beta_1\underline{\Omega}x_4 + {}^{U_2}/_{I_{xx}}) \tag{6.32}$$

The question from here becomes, how can we effectively choose a function of the desired control input such that $\dot{e}_2$ creates the dynamics we desire? Let us reexamine the stability of another candidate Lyapunov function as proposed in Equation 6.33 that carries a similar structure of the original candidate Lyapunov function of 6.25.

$$V_2 = \frac{1}{2}e_1^2 + \frac{1}{2}\varepsilon_1^2 + \frac{1}{2}e_2^2 \tag{6.33}$$

Next, examine the derivative of the candidate Lyapunov function as

$$\begin{aligned}\dot{V}_2 &= e_1\dot{e}_1 + \epsilon_1\dot{\epsilon}_1 + e_2\dot{e}_2 \\ &= e_1\left(-c_1\epsilon_1 - c_2e_1 + e_2\right) + \epsilon_1\left(c_1e_1 - \alpha_1\varepsilon_1\right) + e_2\dot{e}_2 \\ &= -e_1c_1\epsilon_1 - c_2e_1^2 + e_1e_2 + c_1e_1\epsilon_1 - \alpha_1\varepsilon_1^2 + e_2\dot{e}_2 \\ &= -c_2e_1^2 + e_1e_2 + e_2\dot{e}_2 - \alpha_1\varepsilon_1^2\end{aligned} \tag{6.34}$$

Based on the desired definiteness of the Lyapunov function, we require that the derivative of Lyapunov function remain at a minimum negative semi-definite. Therefore, we desire to choose the dynamics of $\dot{e}_2$ to be such that

$$\dot{e}_2 = -e_1 - c_3e_2^2 \tag{6.35}$$

for some positive definite constant $c_3 > 0$. Yielding a negative semi-definite Lyapunov derivative as

$$\dot{V}_2 = -c_2e_1^2 - c_3e_2^2 - \alpha_1\varepsilon_1^2 \preceq 0 \tag{6.36}$$

Therefore, we know that there exists a choice of $\dot{e}_2$ that ensures the Lyapunov definiteness that we seek. Using the knowledge of the desired dynamics from Equation 6.35 and substituting into Equation 6.32 from above, the control input U_2 can be designed to ensure the requirements are met. U_2 is defined below in Equation 6.37.

$$\begin{aligned} U_2 &= I_{xx}\left(-\gamma_1 x_4 x_6 - \beta_1 \underline{\Omega} x_4 + \ddot{\phi}_d + e_1 + c_3 e_2 + c_1^2 e_1 - \alpha_1 c_1 \varepsilon_1 + c_2 e_2 - c_2 \varepsilon_1 c_1 - c_2^2 e_1\right) \\ &= I_{xx}\left(-\gamma_1 \dot{\theta}\dot{\psi} - \beta_1 \underline{\Omega} \dot{\theta} + \ddot{\phi}_{1_d} + e_1\left(c_1^2 - c_2^2 + 1\right) + e_2\left(c_2 + c_3\right) + \varepsilon_1\left(-\alpha_1 c_1 - c_2 \varepsilon_1\right)\right) \end{aligned} \tag{6.37}$$

Next, we must assess the stability of this control system using the general information on stability as seen at the beginning of this chapter. Therefore, using the Lyapunov functions as seen below

$$\begin{aligned} V_2 &= \tfrac{1}{2} e_1^2 + \tfrac{1}{2} \varepsilon_1^2 + \tfrac{1}{2} e_2^2 \\ \dot{V}_2 &= -c_2 e_1^2 - c_3 e_2^2 - \alpha_1 \varepsilon_1^2 \end{aligned}$$

it can be seen that $V_2\left(e_1, e_2, \varepsilon_1\right)$is a continuously differentiable function such that $V_2\left(e_1, e_2, \varepsilon_1\right) > 0 \; in \; D - \{0,0,0\}$ which implies that $\dot{V}_2\left(e_1, e_2, \varepsilon_1\right) \leq 0 \; \forall \; \left(e_1, e_2, \varepsilon_1\right)$ and can guarantee stability as seen from Equation 6.2. The stability can be assessed further since Equation 6.2 and Equation 6.3 hold true, we can conclude GAS of the origin. The corollary to this would be to apply Lasalle's invariance principle. From above we have met the conditions that V_2 is continuously differentiable and negative semi-definite in the compact set of Ω, which is defined as positively invariant. By letting E be the set of all points where $\dot{V}_2 = 0$, implies that $E = \{0,0,0\}$, we can define M to be the largest invariant set contained in E implying $M = \{0,0,0\}$, then every solution will converge to M as $t \to \infty$. Using the second corollary of Lasalle's theorem we can see that since the above conditions are met, and V_2 is radially unbounded then the origin is GAS. It should be noted that the feedback control law developed above and the ones to follow only make the stability guarantees under the knowledge of the known dynamics of the controller. Therefore, when saturation, noise, and additional transfer functions are added to the simulated model to match the system, the controller has no knowledge of these dynamics, and thus the guarantees are no longer valid. However, this work will continue to test the developed control system here with these unknown parameters and determine if the controller is still effective.

The next section will further cover the implementation of the the above control inputs with and without disturbances in the system. However, it is important to note that before additional components are added to the system, we need to verify that the above control inputs can guarantee Lyapunov stability. Therefore, a simple test case is run to stabilize the system from an IC of roll to the origin to evaluate Lyapunov stability and that the requirements of above are met. The Lyapunov and Lyapunov derivative are computed and compared as shown in Figure 6.4 below.

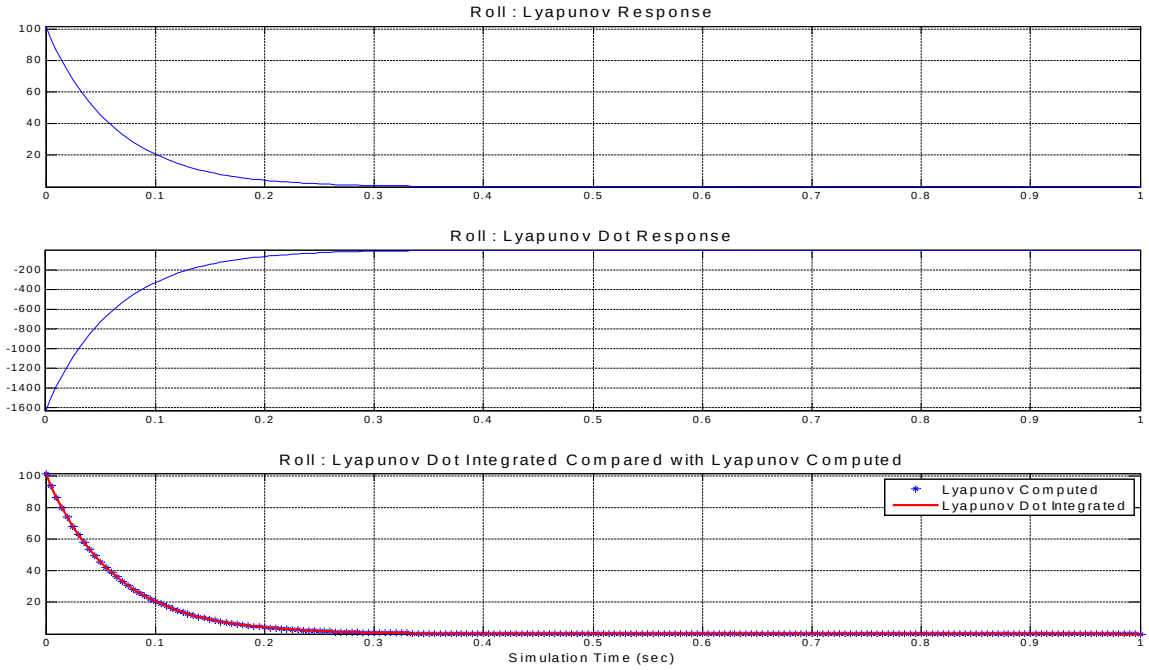


Figure 6.4: This figure shows that the Lyapunov, Lyapunov derivative, and the Lyapunov derivative integrated for the roll axis are correct in both rate and definiteness, and that the choices of control law U_2 guarantees Lyapunov stability for the ideal system.

It can be seen here that the Lyapunov function for roll axis yields a positive definite decreasing function and that the integrated Lyapunov derivative after integrated aligns with the original Lyapunov function; verifying that the computation of the derivative values for the errors and the integral state were computed correctly and perform as desired and designed. This supports the conclusions from above that the controller as presented above will ensure GAS.

Following a similar procedure for deriving the roll controller above (and substituting the corresponding dynamics), a controller can be designed that will guarantee the Lyapunov

requirements and support GAS of the ideal system in both the pitch and yaw rotations. This derivation yields the resulting control inputs as shown in Equation 6.38 below, while Figures 6.5 and 6.6 verify Lyapunov stability as it did in the roll case from above.

$$\begin{bmatrix} U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} I_{yy}\left(-\gamma_2\dot{\phi}\dot{\psi} - \beta_1\underline{\Omega}\dot{\phi} + \ddot{\theta}_d + e_3\left(c_4^2 - c_5^2 + 1\right) + e_4\left(c_5 + c_6\right) + \varepsilon_2\left(-c_4\alpha_2 - c_5c_4\right)\right) \\ I_{zz}\left(-\gamma_3\dot{\phi}\dot{\theta} + \ddot{\psi}_d + e_5\left(c_7^2 - c_8^2 + 1\right) + e_6\left(c_8 + c_9\right) + \varepsilon_3\left(-c_7\alpha_3 - c_8c_7\right)\right) \end{bmatrix} \tag{6.38}$$

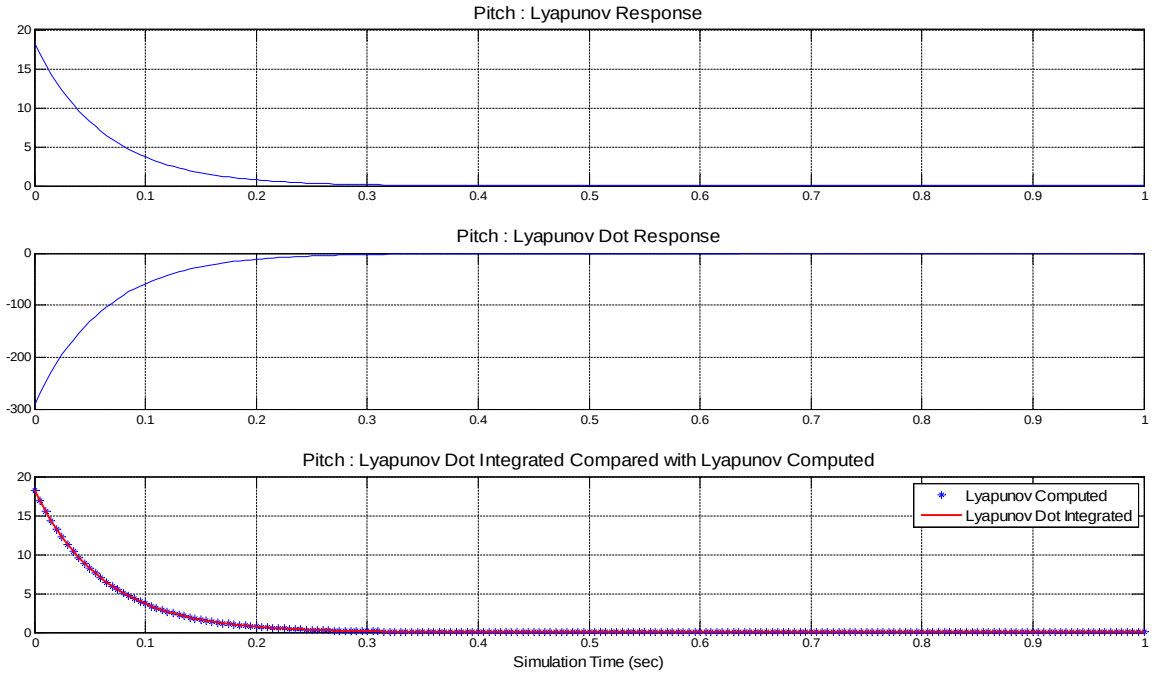


Figure 6.5: This figure shows that the Lyapunov, Lyapunov derivative, and the Lyapunov derivative integrated for the pitch axis are correct in both rate and definiteness, and that the choices of control law U_3 guarantees Lyapunov stability for the ideal system.

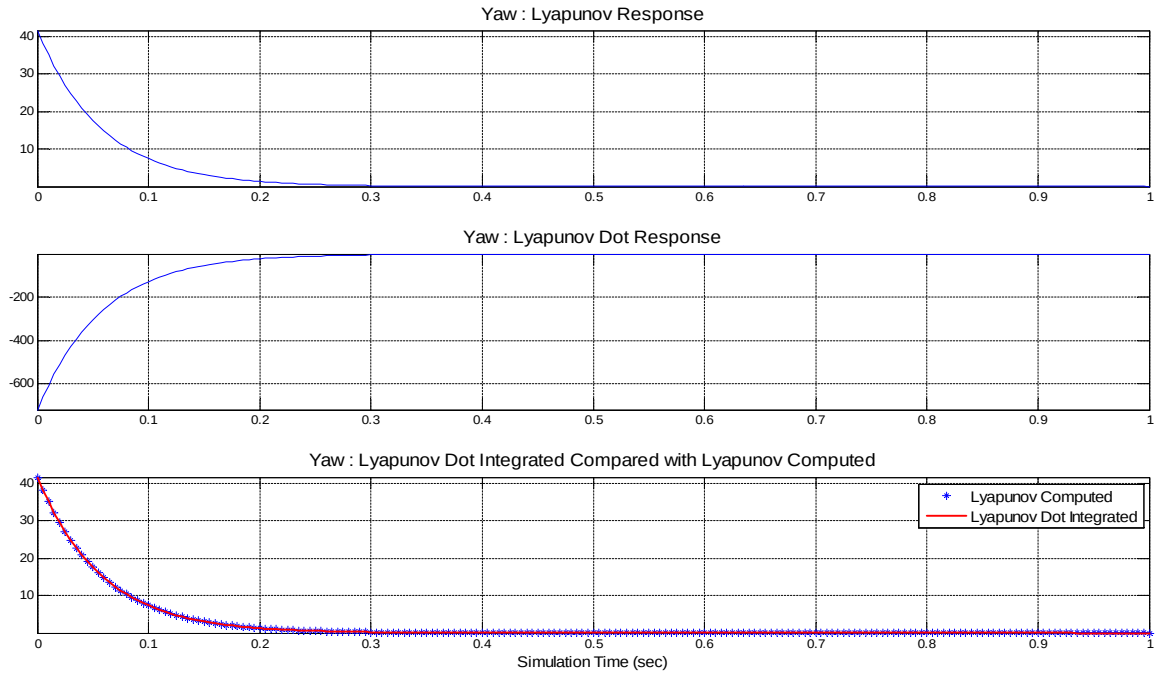


Figure 6.6: This figure shows that the Lyapunov, Lyapunov derivative, and the Lyapunov derivative integrated for the yaw axis are correct in both rate and definiteness, and that the choice of control law U_4 guarantees Lyapunov stability for the ideal system.

6.3.3 Altitude Control

It can be seen from the prior sections that U_2, U_3, and U_4 deal with the rotational subsystem of the vehicle. Now, we consider the translational subsystem of the vehicle. Typically, the translational subsystem would contain an outer-loop control system to govern X,Y, and Z translations. However, this work will only be considering the Z translational system as that is to the extent this work will analyze. Examining X and Y translational control systems opens an entire new array of available sensing components and state estimation options. As stated at the beginning of this work, the outer-loop control system depends on a stable and reliable inner-loop control system, and thus the focus of this work. The application of the derivation of Equations 6.24-6.32 applies for altitude control. The deviation from the derivation occurs when substituting in the correct dynamics for $\ddot{z}$. This substitution can be seen in Equation 6.39 below.

$$\dot{e}_{2a} = (\ddot{z}_d + c_{1a}\dot{\epsilon}_{1a} - c_{1a}\epsilon_{1a}c_{2a} - e_{1a}c_{2a}^2 + e_{2a}c_{2a}) - \left(g - \frac{\cos(\phi)\cos(\theta)}{m}U_1\right) \tag{6.39}$$

Based on the desired stability of the Lyapunov function, we require that the derivative of Lyapunov function remain at a minimum negative semi-definite. Therefore, we desire to choose the dynamics of $\dot{e}_2$ to be such that

$$\dot{e}_{2a} = -e_{1a} - c_{3a}e_{2a}^2 \tag{6.40}$$

for some positive definite constant $c_{3a} > 0$. Yielding a negative semi-definite Lyapunov derivative as

$$\dot{V}_{2a} = -c_{2a}e_{1a}^2 - c_{3a}e_{2a}^2 - \alpha_{1a}\varepsilon_{1a}^2 \preceq 0 \tag{6.41}$$

Therefore, we know that there exists a choice of $\dot{e}_{2a}$ that ensures Lyapunov stability. Using the knowledge of the desired dynamics from Equation 6.40 and substituting into Equation 6.32 from above, the control input U_1 can be designed to ensure the requirements are met. U_1 is defined below in Equation 6.42.

$$\begin{aligned} U_1 &= \tfrac{m}{\cos(\phi)\cos(\theta)}\left(g - \ddot{z}_d - c_{1a}^2 e_{1a} - +c_{1a}\alpha_{1a}\varepsilon_{1a} - c_{1a}\epsilon_{1a}c_{2a} - e_{1a}c_{2a}^2 - e_{2a}c_{2a}\right) \\ &= \tfrac{m}{\cos(\phi)\cos(\theta)}\left(g - \ddot{z}_d + e_{1a}\left(-c_{1a} + c_{2a} - 1\right) + e_{2a}\left(-c_{2a} - c_{3a}\right) + \varepsilon_{1a}\left(c_{1a}\alpha_{1a} + c_{1a}c_{2a}\right)\right) \end{aligned} \tag{6.42}$$

Again, we verify that this controller guarantees Lyapunov stability by simulating the system with some IC and evaluating that the system trajectory tends towards the origin while evaluating the Lyapunov function. It can be seen from Figure 6.7 below that the Lyapunov function for the altitude controller yields a positive definite decreasing function and that the integrated Lyapunov derivative after integrated aligns with the original Lyapunov function; verifying that the computation of the derivative values for the errors and the integral state were computed correctly and perform as desired and designed concluding GAS.

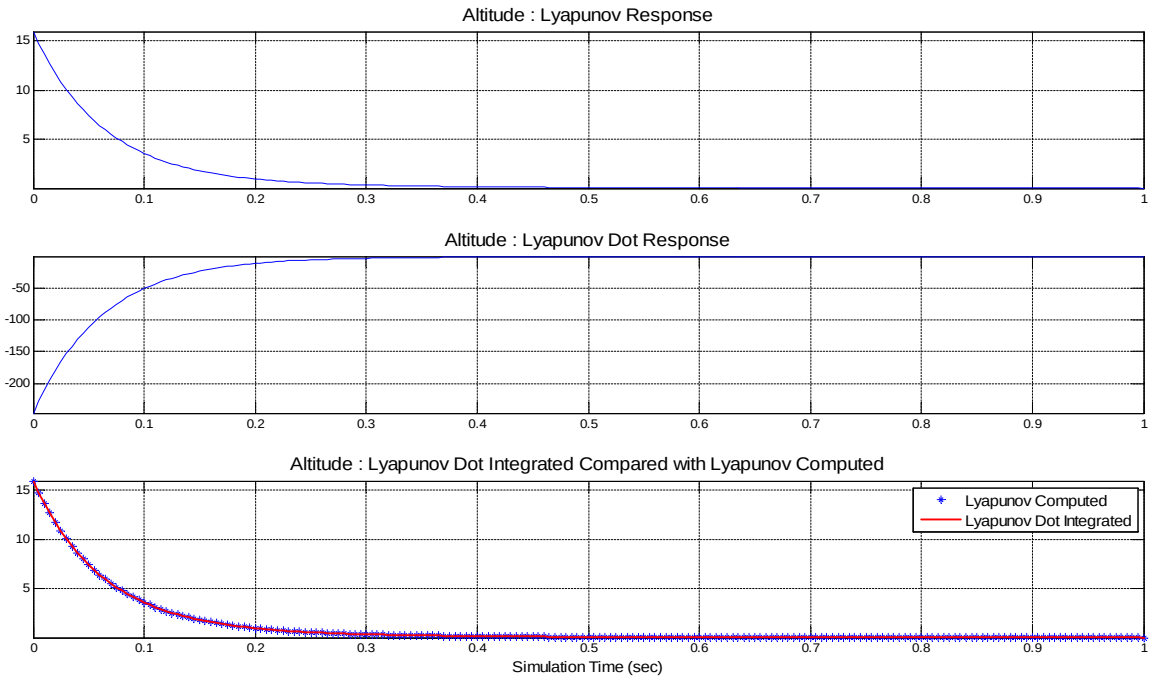


Figure 6.7: This figure shows that the Lyapunov, Lyapunov derivative, and the Lyapunov derivative integrated for the yaw axis are correct in both rate and definiteness, and that the choice of control law U_1 guarantees Lyapunov stability for the ideal system.

6.4 Simulation Realization: From Ideal to Realistic

To fully test the model and validate correct operation, the model had begun development under the ideal case as shown in where each control system was modeled and the control development simulated independent of each other axis. For example, the altitude controller had only considered the dynamics in the Z-direction and had not included the dynamics in the remaining directions, resulting forces in the other directions, and the control had no knowledge of the attitudes of the vehicle. Once each of the axis was modeled and validated in simulation, a full fidelity model was created incorporating all independent control systems together, all of the dynamics together, and lastly had incorporated the different components that increased the realism of the model as discussed from above. The steps below describe in a bit further detail the breakdown of how the system model was modified with additional components at each stage to become the final model used in this work.

1. Under the ideal case, components such as noise, delays, external forces, and motor

response were neglected and it was assumed that the controller had perfect knowledge of the state of the aircraft. Therefore, the IMU component and filtering components were also neglected. Also, the the controller output was a directly mapped force onto the system dynamics in which the response was captured.

2. The next component of building the model was to incorporate the motor mapping, motor dynamics, and motor RPM to thrust conversions as determined in the prior sections. This was the largest single influential component on response of the system.

3. After the motor model was incorporated, it was desired that the system controller update and collect measurements at the sampling frequencies/update frequencies of the system. This was done in SIMULINK through rate-transition blocks and buffer holds throughout the model.

4. To add additional realism to the model, Gaussian distributed white noise was added to components that the sensors would be reading, and accordingly a LPF was implemented to aid in removing the high frequency components of the system in hopes of leaving only the required bandwidth to capture the vehicles dynamics correctly and avoid aliasing.

5. Sampling from sensors in any environment implies a discretized system with bounds on the capabilities of the measurement signal. For many systems, this corresponds to resolution and bandwidth properties that add to the difficulty in achieving accurate and desirable measurements. For this work, this limitation was primarily applied to the limitations to the sonar sensor used in altitude. The reason limitations were not simulated on the MEMS components of the IMU was because at the disturbances and uses for this system, the sensors would not be able to saturate and drastically effect the model.

6. Lastly, programmable external disturbances were added to the model to simulate realistic environmental use. This was the final testing done in the simulation environment to ensure that the controller would be able to recover the system and achieve steady state commanded signals from the operator.

Each of the six steps above were incrementally added to the model as gains were redesigned at each stage to achieve the desired performance of the architect. The next several subsections describe this integration further and the practical issues of these disturbances from the ideal

system in realizing a UAV. The SIMULINK model created for this work and sample Matlab functions can be found in Appendices B and C respectively.

6.4.1 Ideal Response

The ideal response model is self-explanatory. This model had taken the desired force from the controller output and fed this into the derived system dynamics for each independent axis and controller pairing. Performance standards such as rise time, settling time, and overshoot were established based on designed control systems successfully realized from other authors works. Table 6.1 below details the gains used in the simulation for each axis controller, while performance metrics were recorded against the desired performance of the system. The value to the right of the slash in the tables below denote the minimum desired specification that the controller was tuned to. Figure 6.8 below gives an example of the ideal performance in the altitude control scenario.

	α	c_1	c_2	c_3	Rise Time	Settling Time (5%)	Overshoot (%)
Roll	14.75	10.6	4.38	4.475	$0.1\text{s}/\leq 0.125s$	$0.247\text{s}/\leq 0.3s$	8.72%
Pitch	13.25	9.5	3.375	3.475	$0.113\text{s}/\leq 0.125s$	$0.28\text{s}/\leq 0.3s$	8.25%
Yaw	9.75	9.5	4.6	4.65	$0.12\text{s}/\leq 0.5s$	$0.414\text{s}/\leq 1.0s$	7.13%
Altitude	6.75	10.5	4.75	8	$0.34\text{s}/\leq 0.5s$	$0.95\text{s}/\leq 1.0s$	1.48%

Table 6.1: Gain and performance metrics recorded of the ideal scenario for control.

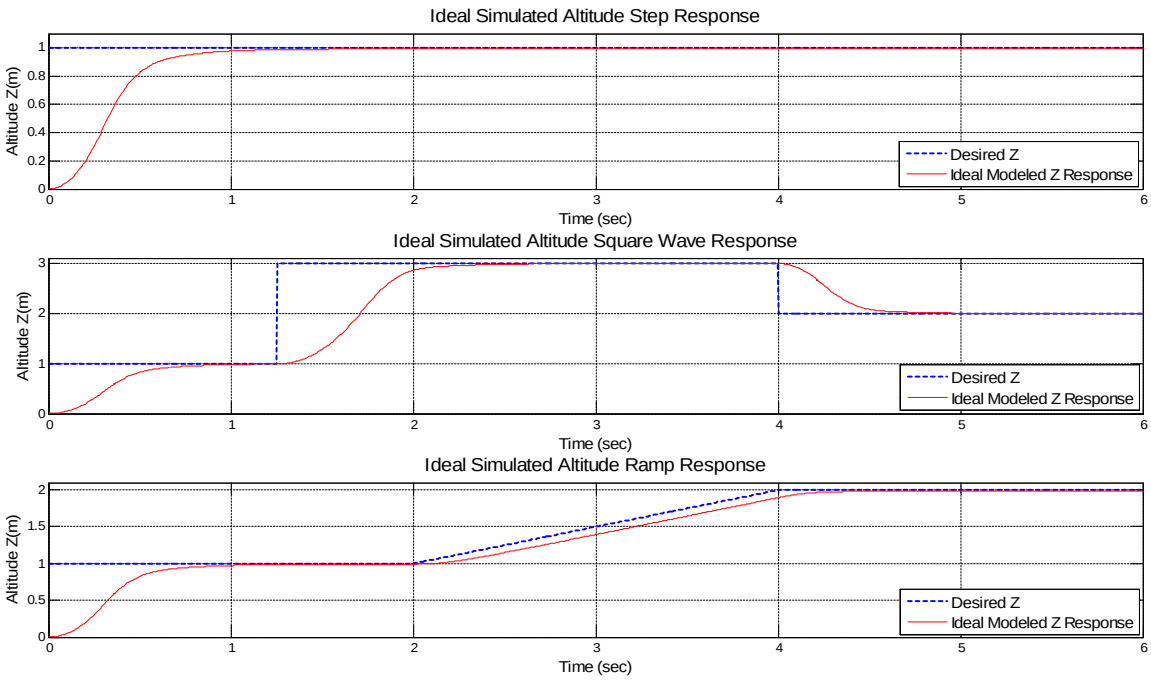


Figure 6.8: This figure details the ideal case response of the HexaCopter platform for various desired inputs such as the step response, square wave response, and ramp response. It can be seen that the system achieves the desired steady state value and tracks the desired performance.

6.4.2 Incorporating the Motor Dynamics

The largest sole influential factor on performance from the ideal model will be the response of the motor dynamics. This presents the largest lag and variable component in response from the controller desired force output to the actual force output of the system. The portion of the SIMULINK model that incorporates the motor response from Section 5.4 above is highlighted below in Figure 6.9. The next several sub-sections cover the purpose of each blockset along with practical implementation issues that had to be recognized in the integration and realization of the actual system.

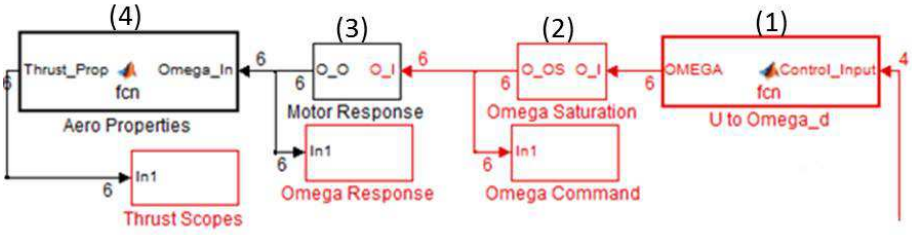


Figure 6.9: This image incorporates how the motor dynamics and response are captured in the model developed for this work. These functions mathematically map the desired control input into the PWM signal for the motor controllers that generate a thrust and produce desired rotations or translations of the system.

6.4.2.1 Converting the control to desired omega

First, the controller has generated a desired control output, $U \epsilon \mathbb{R}^{4x1}$, in which this is the desired force/torque on each respective axis of the system. From here, we must convert this desired force to a PWM signal that the motor controllers can understand. First, we must determine the desired rotational velocity of the propellers through the implementation of Equation 5.35 which is revisited below and implemented in Block(1) of Figure 6.9 above.

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} -b & -b & -b & -b & -b & -b \\ -l_x b & -lb & -l_x b & l_x b & lb & l_x b \\ -l_y b & 0 & l_y b & l_y b & 0 & -l_y b \\ d & -d & d & -d & d & -d \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \\ \Omega_5^2 \\ \Omega_6^2 \end{bmatrix} = bd_{transform} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \\ \Omega_5^2 \\ \Omega_6^2 \end{bmatrix}$$

To determine Ω_i^2 ,where $i = 1:6$, we need to multiply the left of each side by the inverse of $bd_{transform}$. However, typically linear algebra defines a matrix that is invertible if the matrix is of size nxn (square) and when multiplied by the inverse results in the identity matrix. A square matrix which has an inverse is then invertible or non-singular. However, the matrix $bd_{transform} \epsilon \mathbb{R}^{4x6}$, thus this matrix is not square. However, we can apply the Moore-Penrose Inverse. The Moore-Penrose Inverse (pseudo-inverse) is a result of the singular value decomposition thereon stating that if we consider a matrix $\varpi \epsilon \mathbb{R}^{nxm}$ with $rank\,(\varpi) = r$ we

can rewrite ϖ as $\varpi = u\Sigma v^T$; where

$$u\epsilon\mathbb{R}^{nx1} \therefore u^T u = I$$
$$v\epsilon\mathbb{R}^{mx1} \therefore v^T v = I$$
$$\Sigma = diag\left(\sigma_i\right) \; with \; \sigma_1 \geq \sigma_2 \geq ...\sigma_2 > 0 \; and \; \sigma_i \; are \; the \; singular \; values \; of \; \varpi$$

Therefore, we can denote the pseudo-inverse by $\varpi^\dagger = v\Sigma^{-1}u^T\epsilon\mathbb{R}^{mxn}$ with two situations to consider. If $rank\left(\varpi\right) = n$, then we have a full row rank matrix ϖ where the pseudo-inverse is defined by Equation 6.43 below which is often referred to as a right inverse of ϖ.

$$\varpi^\dagger = \varpi^T\left(\varpi\varpi^T\right)^{-1} \tag{6.43}$$

If $rank\left(\varpi\right) = m$, then we have a full column rank matrix ϖ where the pseudo-inverse is defined by Equation 6.44 below which is often referred to as a left inverse of ϖ.

$$\varpi^\dagger = \left(\varpi^T\varpi\right)^{-1}\varpi^T \tag{6.44}$$

For the problem of the inverse of the mapping matrix for the HexaCopter, the $rank\left(bd_{transform}\right) = 4$, therefore, we have a full row rank matrix which implies we can generate a right inverse of $bd_{transform}$ to solve for each motor's desired squared rotational velocity and thus the desired rotational velocity after applying a square root.

6.4.2.2 Saturation limits and discretized signals

There are three areas of saturation that need to be addressed in the model and dealt with in the system implementation. All of the saturation's occur during the generation of the control vector U and calculating the correct propeller speed through the square-root of a pseudo-inverse. The first saturation point occurs at the control generation of the altitude control input of U_1. The control input generated here relates a desired force from the system in each axis. Therefore, U_1 can never be less than zero. This is because the motors cannot generate a force in the direction of gravity, but rather can merely at the lowest point generate zero force by not spinning. Therefore, we require that $U_1 \geq 0$.

The next saturation point is a result from Equation 5.35 where the control input is defined as

a function of the propeller speed squared. Therefore, to determine the desired PWM signal, and accordingly the desired propeller RPM, we must take the square root of Ω^2. Before applying a square root, we must ensure that $\Omega^2 \geq 0$ to ensure the result of the square root is not imaginary.

The last saturation point occurs because the motor has its own inherent saturation limits on the rotational velocity of the propeller as shown from the tests in the prior section. From Figure 5.9, it can be seen that the motor has practical limitations of $0 \leq RPM \leq 7800$. However, examining Figure 5.9, it can be seen that below approximately 3000 RPM, the motor produces a relatively constant thrust value (of which is too little to cause any lift of the vehicle). Thus, if we allow the control effort to achieve such low RPMs, the modeled response of the vehicle will not change as desired and we could possibly encounter the drastic windup nonlinearities as seen in Figure 5.10 and impact the performance of the control system. Therefore, we will impose a further saturation limit on the RPM of the rotor as $2800 \leq RPM \leq 7800$.

Lastly, we require that the RPM signal be mapped into a PWM signal for the motor controller as mapped from Equation 5.42 above. The RPM signal may be mapped into a decimated microsecond signal. Here, we could truncate the result to truncate digits beyond the decimal point, round the result, or force a ceiling or floor result of the signal. For this work, it was decided to ceiling the result since there will be an inherent response time from the transfer function of the motor which will delay the motor response anyway. The above saturation and discretization check of the signals occur in Blocks (1 & 2) from Figure 6.9 above.

6.4.2.3 Motor response transfer function and generated thrust

At this point, we have a desired rotational speed for our rotors from Block(2). Next, the motor responds accordingly to the continuous time-domain transfer function as seen from Equation 5.44. However, the actual and simulated systems are discretized in time, which means we must convert the s-domain transfer function into the z-domain. With the aid of Matlab, the discretized motor response function is as shown in Equation 6.45 below.

$$MR_{C-Response}(s) = \frac{0.99}{0.12175s + 1} \overset{zoh}{\Rightarrow} MR_{D-Response}(z) = \frac{0.03983}{z - 0.9598} \tag{6.45}$$

This RPM is then used to calculate the thrust produced by each individual rotor (Block (4))

from Equation 5.43, which then is used in determining the dynamic response of the system through the EOM derived in the prior sections.

6.4.2.4 Performance with motor response incorporated

At this stage performance was evaluated with the motor model incorporated. Logically, performance suffered because of the response of the motor. The majority of this performance degradation from the original set of gains was a result of the integral state of the controller. As the motor lagged in response to the commanded signal, the integral state would wind-up quickly as the error state did not reduce in value. To compensate this the gains were adjusted again until the performance of the system was at an acceptable level. Table 6.2 below details the new system's gains and performance metrics. Figure 6.10 below the table presents the altitude performance as a result of the motor model incorporation. It can be seen that the rise and settling times had suffered as did the overshoot of the state. Figures for the attitude response with the motor model incorporated are found in Appendix F.

	α	c_1	c_2	c_3	Rise Time	Settling Time (5%)	Overshoot (%)
Roll	7.9	4.25	3.5	22.25	$0.214\text{s}/\leq 0.125s$	$0.25\text{s}/\leq 0.3s$	$1.5\%/\leq 10\%$
Pitch	10.25	3.75	2.75	28.5	$0.219\text{s}/\leq 0.125s$	$0.248\text{s}/\leq 0.3s$	$<1\%/\leq 10\%$
Yaw	9.75	9.5	4.6	4.65	$0.17\text{s}/\leq 0.5s$	$0.414\text{s}/\leq 1.0s$	$7.14\%/\leq 10\%$
Altitude	7.5	12.25	4.25	25	$0.28\text{s}/\leq 0.5s$	$0.95\text{s}/\leq 1.0s$	$17\%/\leq 20\%$

Table 6.2: Gain and performance metrics recorded of motor model incorporation simulation.

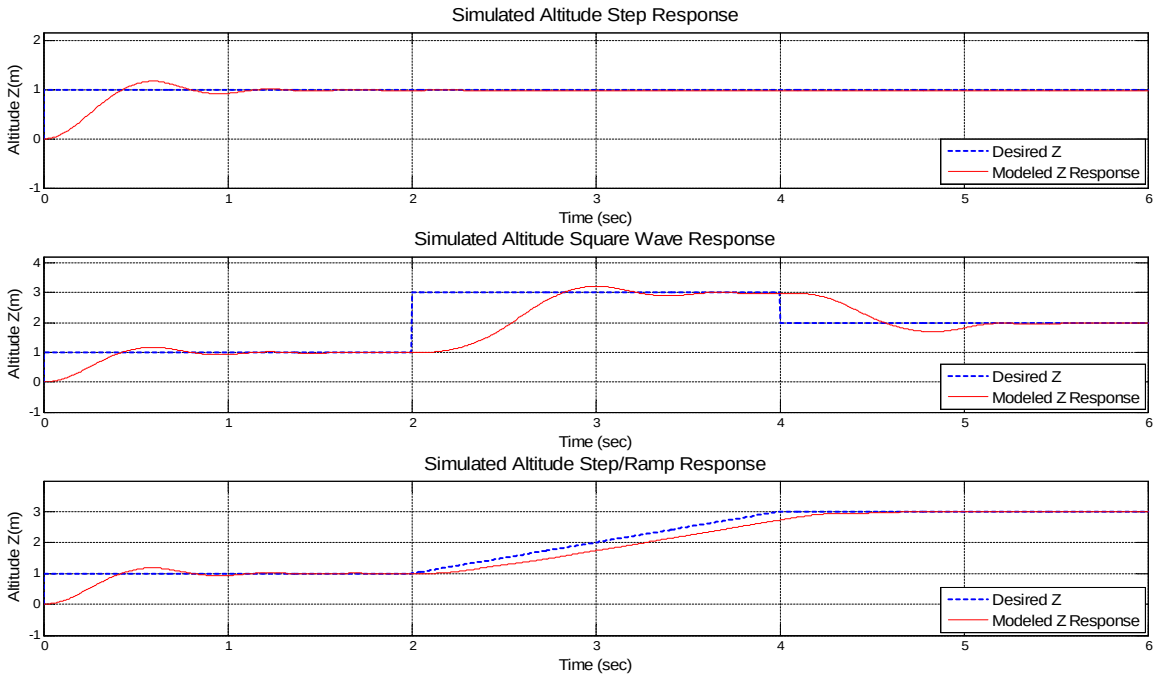


Figure 6.10: This figure demonstrates the altitude controller performance after the motor response and been incorporated into the full system model. It can be seen that performance had suffered, but tracks the typical case of step/ramp input and achieves zero steady state error over-time.

6.4.3 Sampling Time and Update Rates

The difficulty in implementing a system with many individualized components is that each individualized component has their own update frequency based on computer clock speed and the amount of clock cycles each program must execute to produce the desired result. For example, the ultrasonic sensor and GPS unit only receive updates at 20 and 4 Hz respectively. Therefore, altitude control cannot update itself any faster than 20Hz, while position updates would only be achieved at a maximum of 4 Hz (if GPS was used as the sole source of position update). Also, the IMU is clocked at running at approximately 70Hz, however, the governing computer can only perform updates at approximately 65Hz, therefore, the attitude update loop can only occur at a maximum of 65Hz. To accomplish this varying update rates in our model we can implement SIMULINK's rate transition blockset. The rate transition blockset allows for data sets to operate and interact at different rates such as the vehicle's simulation dynamics (overall model simulation step size of 0.005 seconds), altitude control update rates,

and attitude control update rates. This is accomplished by copying the input to the output buffer of these iterations repeatedly until the next timing update in which it updates the output buffer of the above functions. The model code has been color coordinated based on update rate and sampling time as seen in Appendix B for further operator understanding.

6.4.4 Additive Noise Components and Low Pass Filters

Additional noise in the feedback and IMU sensing were considered important as this is realistic to the measurement devices of the IMU. We are given typical noise performance levels of each component aboard the IMU; therefore we can add this as a Gaussian distributed white noise to the state of the measurement. We will also add additional noise components to the output vector of the state from the IMU based on vibrations seen from testing on a stationary IMU with/without the motors running. To better correct these noise levels we know that the sampling frequency of the system occurs at 65Hz, thus we need to design an anti-aliasing filter with a maximum roll-off frequency of approximately 32Hz. However, for this work since we are assuming small and smooth maneuvers, the dynamics of the vehicle will not occur at this rate. Therefore, after some experimentation it was determined that a roll-off frequency of 12Hz would yield performance results that were desired by the developer and a low-pass 4th order Butterworth filter was implemented on raw sensor data collected by the IMU. Figure 6.4.4 below highlights the performance of this filter in the frequency domain and Appendix E shows a sample set of how to discretely implement this filter. Further details of this are beyond the scope of this work, but should be noted that anti-alias filtering is highly important in the performance of discretely measuring the signals from the MEMS devices.

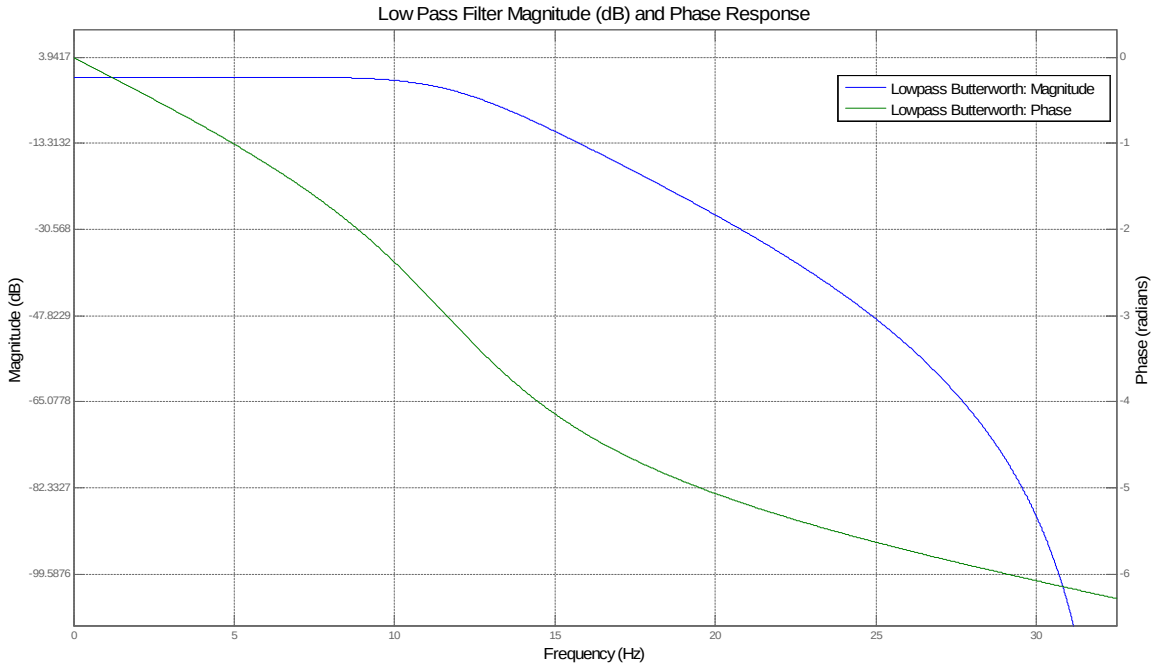


Figure 6.11: This figure shows the magnitude and phase plot of a 4th order Butterworth low-pass filter implemented to filter the measurement signals from unwanted noise and to prevent aliasing.

6.4.5 Altitude Limitations that Hinder Take-Off Procedures

Another restriction that needs to be applied to the model and realized system is the limitations and problems that occur because of the use of a sonar sensor for low-level altitude control. First, the readings from the sonar sensor have a resolution of 1 cm. Therefore, the model must take this into account by rounding its determined altitude in an incremental value. Secondly, there is a limitation as to the range of operation a sonar can successfully be implemented and reliable at. This implies yet another saturation point in the system such that the following must be true: $20cm < z_{read} < 765cm$. However, the low end restriction of the sonar sensor applies yet another practical implementation issue that needs to be considered. If we are to assume that the UAV had not been placed for takeoff on a level surface (which is going to be almost always true) we do not want the controller gains to be winding up the error states when the motors aren't running. Accordingly, we do not want the error states to wind up even if the rotors are running but we are on the ground. Therefore, we propose that the control system not be active until the altitude control has verified that its

altitude is above some set-point value. This implies that the take-off and landing scenarios can be handled in either of two manners. First, the vehicle could be taken-off and landed under manual control. This procedure is in place with many larger aircraft systems for safety of both operators and aircraft, however, this places requirements of the operator to be an experienced pilot. Another option would be to at takeoff, the altitude controller would have to quickly ramp the motor RPMs to near saturation to quickly put altitude between the vehicle and the ground. Upon landing the vehicle, the altitude controller would continually decrease the RPMs of the rotors and that desired RPM value would act as a ceiling on the commanded signal. Therefore, this would guarantee that corrections not increase altitude of the vehicle. A combination of both options had been implemented in this work and testing continues to determine the best procedure. The whole purpose in carefully considering operations near the ground is to prevent unnecessary rollovers and vehicle damage. Also, it should be noted that the lower saturation point of the altitude controller hinders the performance of the altitude controller as the error has no knowledge that it has no velocity or needs to change RPM values until the commanded altitude goes beyond this point and then has to catch up to the desired value.

6.4.6 Additional Force/Torque Components

Additional force components are an important way in analyzing the controller's ability to handle and reject disturbances. To simulate external disturbances, a force/torque can be added in the X,Y,Z component and then broken down into the body forces/torques through the transformation presented in Equation 4.11 above. For this process we are assuming that the external disturbances can be broken down discretely into individual components in their respective directions. However, it should be noted that the additional force/torque component here is a force in Newtons, not a wind velocity etc. The reason for this is that the drag profile of the aircraft has not been estimated. Forces/torques will be simulated in steady and unsteady conditions while evaluating performance responses of the vehicle till the vehicle either achieves steady-state or fails. The force/torque component has been added into the model as shown in Figure 6.12 to the left.

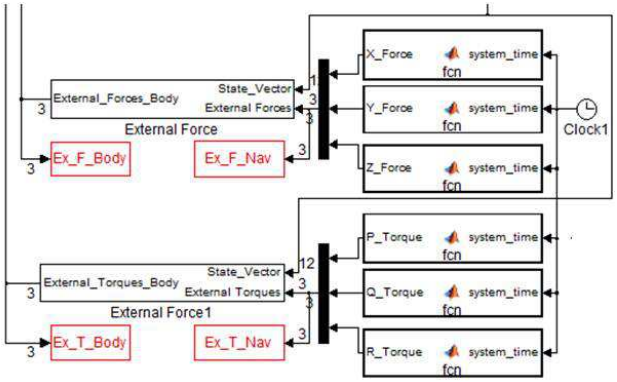


Figure 6.12: This figure shows the external force/torque blockset implemented in SIMULINK. The blocks on the left converted the force/torque from the local to body frame while the blocksets on the right are the desired force/torques generated on a time-set basis in the system.
13.25

6.4.7 Desired Signal Generation

It is important to incorporate a method by which the model's flight dynamics can be quantitatively compared to the realized multirotor dynamics. Also, we desire that there exists an input into the system where the desired established flight attitudes and altitudes can be entered or manipulated. Therefore, four different signal generation blocks were added (3 for attitude, 1 for altitude) to command the multirotor to perform the desired flight commands. These signal generation blocks could create a variety of flight maneuvers that were realistic to a desired flight envelope for the vehicle. Ramps, square waves, sinusoidal waves, steps, and various other input responses were tabulated and evaluated for their performance metrics. The purpose of this input block was to later once the vehicle was realized in flight, to take the issued flight commands to the realized vehicle and issue the said flight commands to the modeled vehicle. At this point, attitude performance could be compared between the actual system and modeled system to evaluate the accuracy of the modeled system dynamics in MATLAB. The incorporation of the desired signal generation block also required a rate transition block to compensate for the varying sample and execution times of the system. Figure 6.13 highlights this blockset from the SIMULINK model.

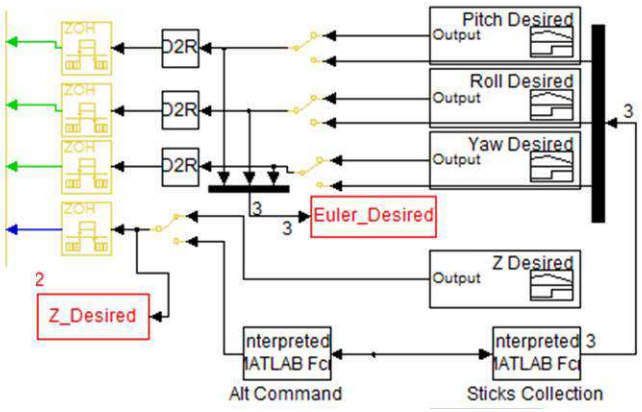


Figure 6.13: This figure highlights the desired input block generation in the program. The ZOH blocks account for the varying rate transitions in the model, while the switch blocks allow for easy switching between a generated signal in the model to commands from flight testing for model validation performance.

6.4.8 Realistic response to various maneuvers and ICs

In order to fully test the controllers abilities to handle a variety of initial conditions, remain stable, and achieve its desired steady state value, simulations were run under a variety of initial conditions as the author saw that the UAV may see during operation. The gains were progressively tuned through multiple iterations of case scenario. Once the performance of the vehicle was satisfactory, random forces and moments representing disturbances were added to the model to evaluate controller performance. The gains that were found to be ideal to the author's performance standards are shown below in Table 6.3. Also, below in Figure 6.14 highlights the performance of the pitch axis flight controller in response to a desired angular trajectory.

	α	c_1	c_2	c_3
Roll	7.85	3.25	2.38	3.65
Pitch	9.25	4.13	1.90	3.40
Yaw	6.75	5.50	2.75	9.25
Altitude	7.00	1.75	2.30	8.00

Table 6.3: Gain and performance metrics recorded of realistic simulation.

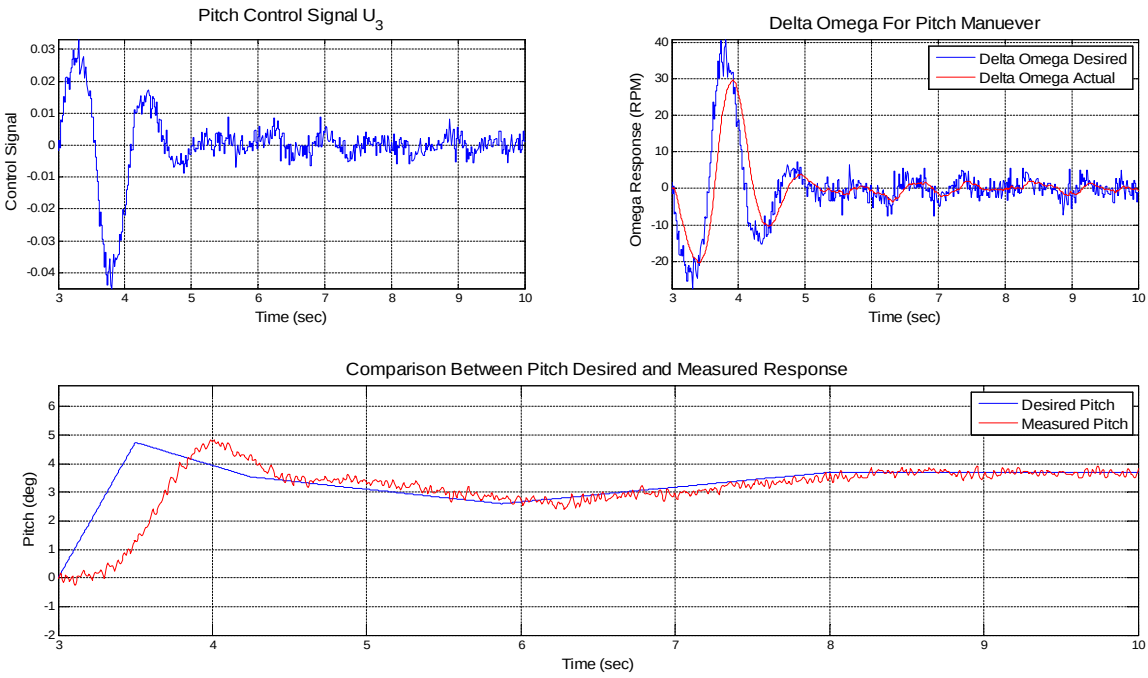


Figure 6.14: Simulated performance of pitch control on the fully realized model in SIMULINK. The upper left figure details the control signal generated for U_3. The upper right figure shows the difference between the desired and actual velocities of Ω_1 and Ω_3. Lastly, the figure at the bottom details the performance of the pitch controller to a generated step and varying response desired trajectory.

Once the fully realized simulation model was completed to a satisfactory performance, it was desired to evaluate that during the entire foreseeable flight envelope of the UAV, the controller would attempt to achieve its desired value and not become unstable. Therefore, a Matlab script was executed to test over 3,000 case scenarios implementing a new random set of initial conditions at the beginning of each execution. The initial conditions that were possible within the simulation are highlighted below along with the desired final condition.

$$\begin{aligned}
z_{alt-final} = 1m \quad &\& \quad 0m \leq z_{IC} \leq 2m \\
\phi_{final} = 1^o \quad &\& \quad -15^o \leq \phi_{IC} \leq 15^o \\
\theta_{final} = 1^o \quad &\& \quad -15^o \leq \theta_{IC} \leq 15^o \\
\psi_{final} = 1^o \quad &\& \quad -45^o \leq \psi_{IC} \leq 45^o \\
p_{final} = 0^{deg/sec} \quad &\& \quad -5^{deg/sec} \leq p_{IC} \leq 5^{deg/sec} \\
q_{final} = 0^{deg/sec} \quad &\& \quad -5^{deg/sec} \leq q_{IC} \leq 5^{deg/sec} \\
r_{final} = 0^{deg/sec} \quad &\& \quad -5^{deg/sec} \leq r_{IC} \leq 5^{deg/sec}
\end{aligned}$$

The program then evaluated each response at the end of the simulation to ensure that steady state response was achieved and logged overshoot, rise time, and settling time. It was found that under the potential initial conditions from above, there were some cases that had presented the controller some difficulty in arriving to a steady state condition in a short amount of time (<3sec). However, after further investigation, the conditions that caused the longer settling times were often very difficult conditions that most likely would not be seen in the normal flight envelope this system was designed for. In conclusion, the results of this case study had shown that not one test case produced a result that had not achieved the desired steady state performance condition. This however, does not mean that there does not exist a condition that may cause the controller to become unstable, but this situation is considered unlikely given the envelope considered. To visualize the performance of the system a viewer program was created to watch varying simulation scenarios. Figure 6.15 shows a sample program that was developed in this work so that an operator may visualize simulated attitudes/altitude of the UAV system to better understand how the system will perform in flight.

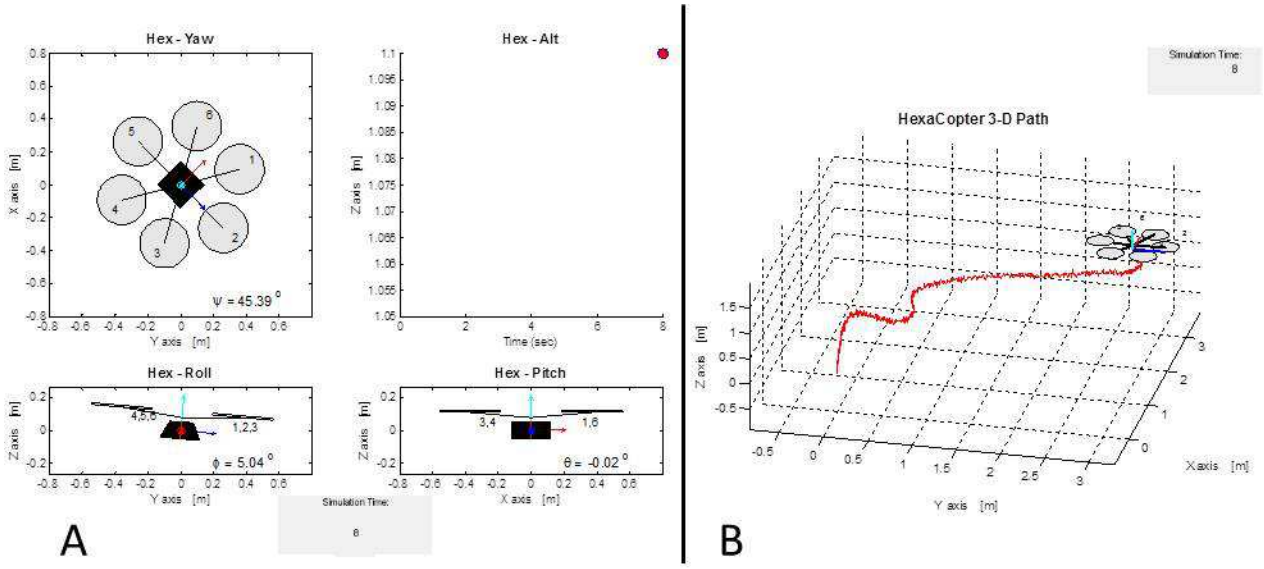


Figure 6.15: This figure demonstrates the HexaCopter visualization tools developed to observe simulated attitudes and altitude of the UAV. Figure (A) helps in visualizing flight attitude/altitude while Figure (B) helps in visualizing the potential 3D flight path the vehicle will take during operating.

Chapter 7

Implementation Results

This last chapter discusses the HIL simulation done to ensure correct programming of the flight computer, and presents a sample set of results highlighting a logged set of flight data and compares the performance to that of the simulated system.

7.1 Hardware-in-the-Loop Simulations

HIL simulations had taken place once the controller had achieved performance results in the SIMULINK simulation that were desirable of the author. HIL simulations are a vital part of ensuring successful operations and integrations of the software system. First, the controller code was ported to the flight computer and then tested against data obtained from the model. Therefore, the SIMULINK model determined parameters such as the dynamics and the response of the system and allowed the flight computer to execute the controller code so that an operator could validate that the controller aboard the computer was performing properly. Testing then continued to include other peripherals and features of the vehicle to evaluate successful implementation of software and communication protocol between the vehicle and the UGCS. The HIL simulation is highlighted graphically below in Figure 7.1.

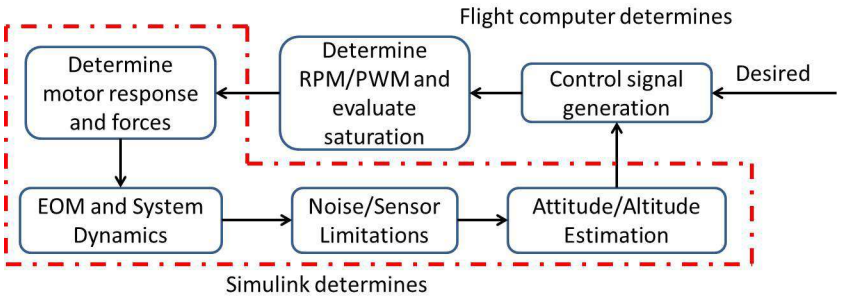


Figure 7.1: This figure shows how HIL testing had taken place to validate the controller code aboard the flight computer.

7.2 Flight Testing & Model Validation through Experimental Data

Flight testing had taken place throughout the time-span of a year continually developing and modifying code to increase performance and optimizing data throughput to the operator. Testing had consisted of both command operations from the pilots remote, commands received from the UGCS, and commands generated internally based on a script execution file telling the vehicle the flight maneuvers to perform. Modifications were made in the design of the vehicle as failures were seen in some mechanical components and electrical components during flight operations. Figure 7.2 below shows a sample of the flight test setup taking place at Kentland Farms on the left. The figure on the right is a close-up of V1.5 of the HexaCopter in flight. The current model of the HexaCopter platform is denoted as 2.1.

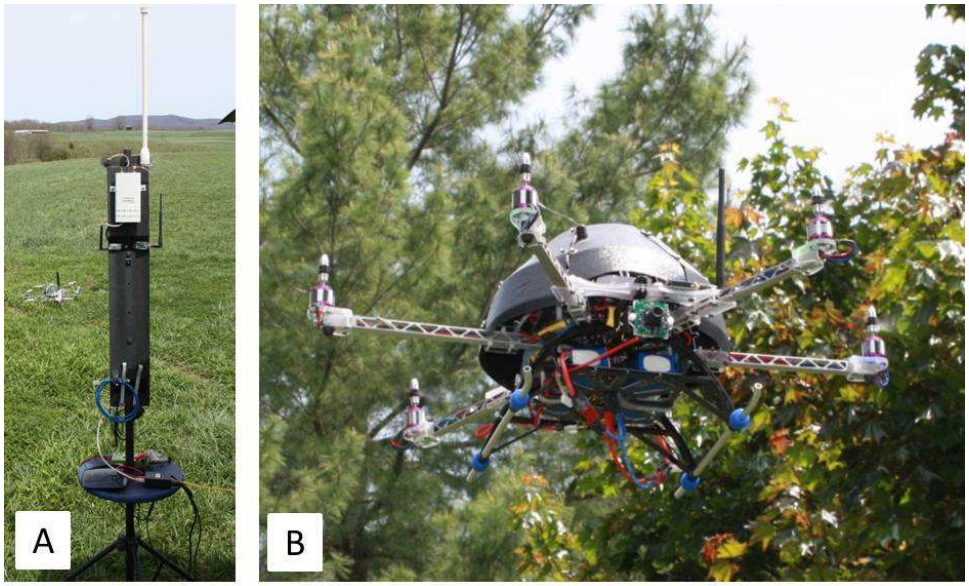


Figure 7.2: Figure (A) on the left shows the radio antenna setup used for flight operations along with the HexaCopter platform in the background. Figure (B) on the right is a flight photo of the original HexaCopter platform.

Testing had taken place in a variety of environmental conditions with varying levels of external disturbances to evaluate the effectiveness of the flight control system. The last important task of this work was to compare actual flight data to that of the simulated response data. For this test, a set of flight commands was issued to the vehicle and logged along with the coordinating response of the vehicle. The same issued flight commands were then implemented into the simulation as the desired set of parameters. The response of the simulated and actual systems in the roll and pitch axis' can be seen below in Figures 7.3 and 7.4. These figures highlight that the simulated response of the system compares well to that of the actual system in both magnitude and phase. However, the flaw in the simulated response is the apparent lag of the response compared to that of the actual system. This could be a result from an overestimation of the moments of inertia, underestimation of the thrust factor of the propellers, or unmodeled dynamics in the platform. Overall, this flight data supports that the SIMULINK model developed in the work is a viable application for testing and evaluating flight control strategies developed for this HexaCopter platform.

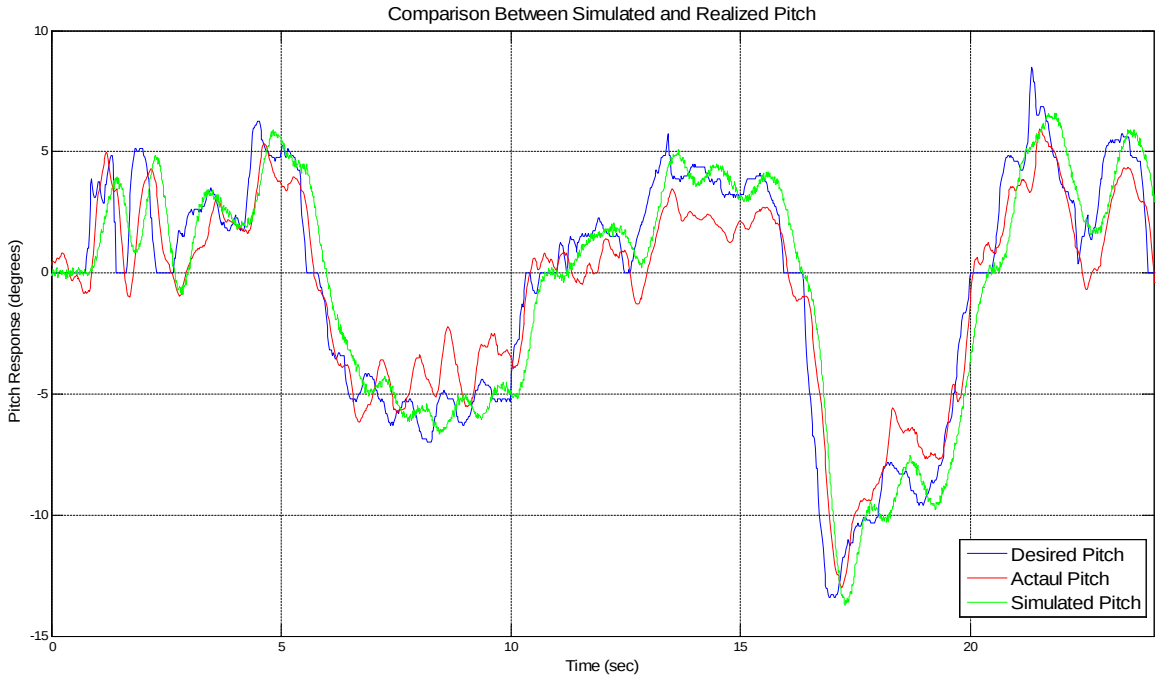


Figure 7.3: This image shows a sub-set of the pitch flight data as compared with the simulated response model data to the same input. It can be seen that the simulated pitch response lags the actual pitch response of the system, however, general trends and magnitudes are correct when comparing the datasets.

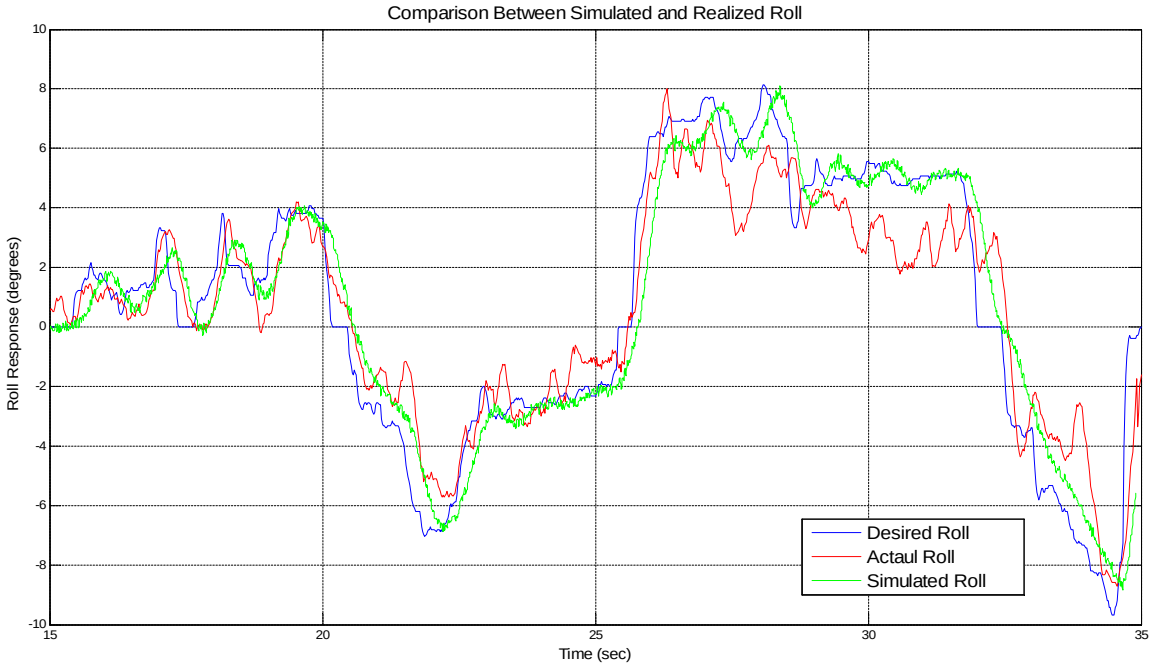


Figure 7.4: This image shows a sub-set of the roll flight data as compared with the simulated response model data to the same input. It can be seen that the simulated roll response lags the actual roll response of the system, however, general trends and magnitudes are correct when comparing the datasets.

Chapter 8

Conclusions and Recommendations

This thesis had presented the research and work done to develop a fully deployable multirotor UAV system for use in further testing and evaluating unmanned systems technologies. A summary of the work conducted towards this research has been collected here along with a summary of some of the major suggestions that were made throughout the work to improve the overall system and to add work in the development of the platform. While each chapter of the thesis addresses a distinct area of work, all contribute to the overall design and successful implementation of multirotor UAV system with a integral backstepping controller for attitude/altitude stabilization.

It is believed that by the end of this work, it has been shown that this work has meet the desired goals as initially set forth in this work. The goals of this work were derived on the recommendations from other works, and the overall generic goals of this work were to :

- Design and develop a UAV system. This UAV system will have to be able to handle not only its own weight for flight, but also be able to incoorporate necessary additional payloads for practical uses and further flight research. These payloads may include a LIDAR system, scanning IR system, cameras, additional flight computers, and other typical flight research payload platforms. Also, this UAV system will require a reasonable amount of flight endurance so that using this vehicle as a test platform is a viable option.
- Derive/Test/Implement a reliable way in estimating the vehicle state. It can be seen from the literature review that many works have computational complications as a

result of trying to estimate far greater states than potentially what is needed for a minimalistic control scheme and inner-loop control.

- Design/Test/Implement necessary flight electronics to reduce weight and increase efficiencies.
- Implement an efficient and robust communication protocol for data transfer between the operator on the ground and the air vehicle. This will include an effective way to present information to the operator so that he/she may understand vehicle status and vehicle information.
- Develop a reliable and accurate mathematical model environment in which to test derived control strategies aboard the vehicle.
- Implemenet/Test flight control strategies in an untethered environmental setting and compare realized system perfomance to that of the mathematical model performance.

8.1 Summary of Work

First, a set of design specifications and desirable characteristics of a flight vehicle were tabulated and assigned metric values as the architect saw in the importance of the design and the foreseeable uses of this UAV system. Once a type of UAV was determined, a motor/ESC combination was selected based on desirable payload capacities and potential estimated flight weight of the vehicle. As the HexaCopter platform became further matured in the design process through a CAD model, analysis was done on areas where many other works sugguested to be problem inducing areas affecting flight performance. First, the supporting arms to the motor mount assemblies were analyzed through FEA to determine the amount of weight and material removal possible while maintaining strength and rigidity. Motor mounts were designed to isolate motor induced vibrations from the system and the IMU. Next, electrical systems were designed, developed, and tested to ensure reliability and distribution to the systems aboard the UAV. Lastly, a communications architecture was implemented in a network based system to facilitate data transfer between the air vehicle and a ground station developed for this work.

In establishing a coordinate frame system for the UAV, relations about the vehicle's attitude were developed and transformations were proposed between the body and local coordinate

frame systems. Next, sensor modeling was analyzed based on presented works in literature, and an EKF with low-pass butterworth filter was implemented aboard an IMU for sensing and estimating the attitude of the HexaCopter. Modeling and flight control were the next developments of this work. Using the coordinate frames established from above, EOMs were derived using the Newton-Euler formulations for force and moment calculations of the system. Parameters that were negligible as sugguested by other literature works were ignored in this derivation. Unknown parameters of the system were collected either through experimentation or analysis aid from the CAD model.

Next, a nonlinear integral backstepping control system was derived to gaurantee Lyapunov stability of the ideal attitude/altitude systems. A dynamic model in SIMULINK was created to facilitate the testing and validation of the control system for the HexaCopter platform. The simulation grew as the model of ideal system had iteratively added components realistic to that of the realized system. These components had included factors such as saturation and discretization points, motor response, sensor limitations, sampling and update rates, corrupted sensor measurements from white noise, and lastly external disturbances on the system. At each iteration the flight control system and filtering schemes were tuned to ensure correct and desirable performance metrics, while thousands of test cases and HIL simulations were executed to ensure correct perfromance of the system.

Lastly, the flight control system was realized and tested in varying flight scenarios and demonstrated. An operator was able to view streaming data and operate the vehicle from the developed UGCS terminal and interact with the provided payloads for improved situational awareness. Flight test cases were performed in order to validate the accuracy and reliability of the model. Therefore, input and output data was recorded from the realized UAV system during flight. Then, the inputs to the flight were used as inputs into the SIMULINK modeled system and the output of the model was recorded. The two outputs were then compared for accuracy and trends to ensure that the Simulink model could be used further as a accurate model for the HexaCopter platform. It was concluded that the SIMULINK model is accurate and a useful tool in further developing control systems or modeling performance of the HexaCopter platform.

8.2 Suggestions for Future Work

Efforts were made throughout the duration of this work to effectively minimize the weight of the HexaCopter platform by removing material where possible, using composite materials and additive manufacturing, and developing custom circuit components to reduce the overall weight and size of PCBs. However, future work could include further analysis into material reduction through the entirity of the vehicle. Also, further CAD work into the design of the platform should be done to reduce the amount of required hardware components (nuts, bolts, washers, etc.) to assemble the vehicle. Plastics were used where possible for hardware components where the author thought loads were minimal, however, analysis on these connection points should be done to determine the proper hardware for the loading conditions.

Another proponent in the larger weight of the UAV is a result from the numerous connectors and wiring aboard the vehicle. It is difficult within a system that contains numerous required subsystems to avoid wires and connectors running throughout the vehicle. However, analysis should be conducted to determine the correct gauge wiring for both signal and power lines aboard the UAV. Also, redesigning and consolidating circuits would reduce the number of connectors contained within the system. Connections that are permanent could be replaced with direct solder joints to the PCB rather than an intermeddiate connecting component. An example of this would be the power distribution hub between the batteries and the remaining components of the system. By ridding these connectors from the system, the vehicle would be approximately 0.145 pounds lighter. Although this is only a single example, if the amount of connectors could be reduced further the weight savings could amount into longer flight durations as seen from Figure 3.18.

With regard to the state estimation of a UAV system, an EKF has been tested for the implementation of this work. It has been shown to be an invaluable component for the control system to achieve the results that it has been able to. However, Kalman filters in general are very processor taxing when it comes to implementation. At the conclusion of this report, the author has been testing the possibilites of moving the control and attitude estimation code onto a real-time FPGA processor from National Instruments. There are numerous advantages to this change that result in hardware reduction and speed possibilites. First, this would remove the processing component from the IMU. Removing the processing component from the IMU and allowing the sbRIO to read the signals from the MEMS

components directly would allow for increased update rates of the vehicles attitude and state for the flight controller. Logically, if the control can move faster as well, the controller would be able to update the ESCs faster and accordingly responsd to outside disturbances at a higher rate while achieving steady-state performance faster. Also, if there were still available processing room aboard the FPGA, consolidation could be made from PPM encoder recieving the RC signal and the Pololu unit controlling the ESCs. This consolidation would reduce the complexity of the code structure on the vehicle, while reducing the amount of wiring and hardware components that are adding weight to the vehicle. Also, by reducing the amount of processing components and hardware components in turn decreases the amount of potential points of failure throughout the system. Work is currently under-way to determine if this is a vaible solution for a flight control and processing computer.

As stated above, changing the flight control processing computer and the microprocessor contained aboard the IMU would potentially increase updates to the ESCs and result in achieving higher performance results of the system. However, the limitation in this is the ability for the ESCs to handle the higher update rates and the effectiveness of the rates have on updating the inertia as seen in the rotor assembly. Having higher update rates of the thrusters would not hinder the performance, however, a study into the effectiveness in acquiring newer ESCs versus the response updates potentially gained would need to be completed.

Much of the limitations in the capabilities of the project had come in the form of software fixes and patches that were produced to fix problems in the software architecture, however, the effects of execution time was not evaluated. Therefore, some of the code may have not been produced to run in the most efficient manner possible. An example of this is the calculation of the psuedo-inverse; there are many multiplication operations present which take many clock cycles to compute. Thus, to continue this project it is recommended that the software on both the ground control station and air vehicle be reviewed and trimmed.

Next, it is recommended that potential avenues of other nonlinear control systems be explored. With the ability for added computational power, stronger non-linear control systems could be implemented in order to gain more robustness and gaurentees from the control system. The control presented in this work, as stated before, only gaurantess Lyapunov stability for the ideal case where the controller has full knowledge of the dynamics of the system. However, it should be trivially seen that the controller presented does not have full knowledge of the motor response, system saturation points, and other external disturbances

on the system. Also, it should be noted that the flight envelope of this vehicle is only a small subset of the possibilites of the operating range that this vehicle could achieve. Therefore, it would be recommended that other non-linear control schemes be investigated for their ability to ensure stronger gaurentees on the system. For example, Model Reference Adaptive Control (MRAC) and other similar variants should be explored as this class of controllers can adapt to various unmodeled and uncertain dynamics from the system. However, if the uncertainties can often be assumed bounded, often further stabilities can be gauranteed, and robustness properties can be expanded for the system.

Lastly, it is highly recommended that a control scheme be further investigated for outer-loop trajectory control. Even though currently flight operations are stable enough for manual and assisted/stabilized flights, there is no outer loop to keep the system in a single desired position. Therefore, the implementation of this vehicle in its current state isn't as useful to the lab as a research platform for this reason. It is recommended that the areas of vision coupled with a GPS be used for localization and outer-loop control. Currently, an outer-loop control system could be developed to interact with the implemented inner-loop flight control by developing a desired set of roll, pitch, and yaw commands based on its desired positioning and orientation of the aircraft. However, the computer in its current state may not be able to handle the overhead of an additional control scheme if it were executed at any substantial update rate. Therefore, it follows that from the recommendation above that the control and processor code be overhauled for efficiency.

By implementing some of the sugguestions from above, the realized UAV system could achieve more robustness, stability, and become a viable research platform for use in future developments at the Unmanned Systems Lab. In conclusion, the research and developments created in this work lay out the ground-work to achieve the desired flight system from UAV needed in this laboartory setting.

Bibliography

[1] Ehrhard, Thomas P. “Air Force UAVs: The Secret History.” Mitchell Institute, July 2010. Web.

[2] Castillo, Pedro, R. Lozano, and Alejandro E. Dzul. “Sensors, Modems and Microcontrollers for UAVs.” Modeling and Control of Mini-flying Machines. London: Springer, 2005. 185-230. Print.

[3] Titterton, D. H., Weston, J. L., Strapdown Inertial Navigation Technology, American Institute of Aeronautics and Astronautics & Institution of Electrical Engineers, 2004. Print.

[4] Rogers, M.R., Applied Mathematics in Integrated Navigation Systems, AIAA, Inc., 2000. Print.

[5] Kayasal, Ugur. "Modeling and Simulation of a Navigation System with an IMU and a Magnetometer." Thesis. Middle East Technical University, 2007. Print.

[6] Flenniken, Warren S., IV. “Modeling IMUs and Analyzing the Effect of Their Errors in Navigation Applications.” Thesis. Auburn Universtiy, 2005. Print.

[7] Walchko, Kevin J., and Paul A.C. Mason. "Inertial Navigation." Recent Advances in Robotics. Florida. 2002. Lecture.

[8] Kumar, Vikas N. "Integration of Inertial Navigation System and Global Positioning System Using Kalman Filtering." Thesis. Indian Institute of Technology, 2004. Print.

[9] Marmion, Mathieu. "Airborne Attitude Estimation Using a Kalman Filter." Thesis. Norwegian University of Science and Technology, 2006. Print.

[10] Fux, Samuel. "Development of a Planar Low Cost Inertial Measurement Unit for UAVs and MAVs." Thesis. Swiss Federal Institute of Technology, 2008. Print.

[11] Ronnback, Sven. "Development of a INS/GPS Navigation Loop." Thesis. Lulea University of Technology, 2000. Print.

[12] King, Peter H. "A Low Cost Localization Solution Using a Kalman Filter for Data Fusion." Thesis. Virginia Polytechnic Institute and State University, 2008. Print.

[13] Nonami, Kenzo. "Design and Implementation of Low-Cost Attitude Quaternion Sensor." Autonomous Flying Robots: Unmanned Aerial Vehicles and Micro Aerial Vehicles. Tokyo [etc.: Springer, 2010. 251-66. Print.

[14] Simon, Dan. Optimal State Estimation: Kalman, H [infinity] and Nonlinear Approaches. Hoboken, NJ: Wiley-Interscience, 2006. Print.

[15] Nonami, Kenzo. "Guidance and Navigation for Small Aerial Robots." Autonomous Flying Robots: Unmanned Aerial Vehicles and Micro Aerial Vehicles. Tokyo [etc.: Springer, 2010. 219-50. Print.

[16] Martinez, Vicente. “Modeling of the Flight Dynamics of a Quadrotor Helicopter.” Thesis. Cranfield University, 2007. Print.

[17] Wierema, Menno. “Design, implementation and flight test of indoor navigation and control system for a quadrotor UAV.” Thesis. Delft University of Technology, 2008. Print.

[18] Stepanik, Michael J. "A Quadrotor Sensor Platform." Diss. Russ College of Engineering and Technology of Ohio, 2008. Print.

[19] Wu, Yiting. "Development and Implementation of a Control System for a Quadrotor UAV." Thesis. University of Applied Science Ravensburg-Weingarten of Germany, 2009. Print.

[20] Bouabdallah, Samir. "Design and Control of Quadrotors with Application to Autonomous Flying." Thesis. Swiss Federal Institute of Technology, 2007. Print.

[21] Tayebi, A. and McGilvray, S. (2006). Attitude stabilization of a vtol quadrotor aircraft. IEEE Transactions on Control Systems Technology, 14(3):562–571.

[22] Kemp, C. (2006). Visual Control of a Miniature Quad-Rotor Helicopter. PhD thesis, University of Cambridge.

[23] Hoffmann, G. M., Huang, H., Waslander, S. L., and Tomlin, C. J. (2007). Quadrotor helicopter flight dynamics and control theory and experiment.

[24] Guenard, N., Hamel, T., and Moreau, V. (2005). Dynamic modeling and intuitive control strategy for an X4-flyer. In ICCA 05 Budapest.

[25] Madani, T. and Benallegue, A. (2006b). Control of a quadrotor mini-helicopter via full state backstepping technique. In Proceedings of the 45th IEEE Confer- ence on Decision & Control, San Diego, CA, USA.

[26] Altug, E. (2003). Vision based control of unmanned aerial vehicles with applications to an autonomous four rotor helicopter, quadrotor. PhD thesis, GRASP Lab, University of Pennsylvania.

[27] Xu, R. and Ozguner, U. (2006). Sliding mode control of a quadrotor helicopter. In Proceedings of the 45th IEEE Conference on Decision & Control, San Diego, CA, USA.

[28] Dunfied, J., Tarbouchi, M., and Labonte, G. (2004). Neural network based control of a four rotor helicopter. In IEEE International Conference on Industrial Technology (ICIT).

[29] Bucholz, Thorhallur T., and Dagur Gretarsson. "Construction of a Four Rotor Helicopter System." Thesis. Technical University of Denmark, 2009. Print.

[30] Bresciani, Tommaso. "Modelling, Identification and Control of a Quadrotor Helicopter." Thesis. Lund University of Sweden, 2008. Print.

[31] "Find the Magnetic Declination at Your Location." Magnetic Declination. Web. <http://magnetic-declination.com/>.

[32] Caruso, Michael J. Applications of Magnetic Sensors for Low Cost Compass Systems. Tech. Honeywell, SSEC. Web. <http://www.honeywell.com/sites/servlet/com.merx.npoint.servlets.DocumentServlet?docid=D84A3 2A47-A0C3-F3BF-E2F7768BD449>.

[33] Kim, P'ir-yŏng, and Lynn Huh. Kalman Filter for Beginners: With MATLAB Examples. United States: CreateSpace, 2011. Print.

[34] Euston, Mark, Paul Coote, Robert Mahony, Jonghyuk Kim, and Tarek Hamel. "Complementary Filter for Attitude Estimation of a Fixed-Wing UAV." Print.

[35] Baldwin, Grant, Robert Mahony, Jochen Trumpf, Tarek Hamel, and Thibault Cheviron. "Complementary Filter Design on the Special Euclidean Group SE(3)." Print.

[36] Welch, Greg, and Gary Bishop. "An Introduction to the Kalman Filter." An Introduction to the Kalman Filter. Department of Computer Science, University of North Carolina at Chapel Hill, 24 July 2006. Web. 07 Jan. 2012. <http://www.cs.unc.edu/~welch/kalman/kalmanIntro.html>.

[37] Kumar, Vikas N. "Integration of Inertial Navigation System and Global Positioning System Using Kalman Filtering." Diss. INDIAN INSTITUTE OF TECHNOLOGY, 2004. Print.

[38] King, Peter H. "A Low Cost Localization Solution Using a Kalman Filter for Data Fusion." Thesis. Virginia Polytechnic Institute and State University, 2008. Print.

[39] Grewal, Mohinder S., and Angus P. Andrews. Kalman Filtering: Theory and Practice Using MATLAB. Hoboken, NJ: Wiley, 2008. Print.

[40] Cai, Guowei, Ben M. Chen, and Tong H. Lee. Unmanned Rotorcraft Systems. New York: Springer-Verlag, 2011. Print.

[41] Castillo, Pedro, R. Lozano, and Alejandro E. Dzul. "The Quad-rotor Rotorcraft." Modeling and Control of Mini-flying Machines. London: Springer, 2005. 39-59. Print.

[42] Welch, Greg, and Gary Bishop. An Introduction to the Kalman Filter. University of North Carolina at Chapel Hill Department of Computer Science, Aug. 2001. Web. <http://www.cs.unc.edu/~{welch, gb}>.

[43] Khalil, Hassan K. *Nonlinear Systems.* Upper Saddle River, NJ: Prentice Hall, 2002. Print.

[44] Saif, Abdul, Mujahid Dhaifullah, Mohammad Al-Malki, and Mustafa Shafie. “Modified Backstepping Control of Quadrotor.” International Multi-Conference on Systems, Signals, and Devices 9 (2012): n. pag. IEEE. Web.

[45] Lungu, Mihai. "Stabilization and Control of a UAV Flight Attitude Angles Using the Backstepping Method." World Academy of Science, Engineering and Technology 61 (2012): 290-97. Web.

[46] Koschorke, Johann. "Advanced Flight Control Design and Evaluation." Thesis. Delft University of Technology, 2012. Print.

[47] Colorado, J., A. Barrientos, A. Martinez, B. Lafaverges, and J. Valente. "Mini-quadrotor Attitude Control Based on Hybrid Backstepping & Frenet-Serret Theory." 2010. MS. University of Madrid, Madrid.

[48] Oliveira, Marcelo. "Modeling, Identication and Control of a Quadrotor Aircraft." Thesis. Czech Technical University of Prague, 2011. Print.

[49] "MAVLink Micro Air Vehicle Communication Protocol." - QGroundControl GCS. N.p., n.d. Web.

[50] "UIUC Propeller Data Site." UIUC Propeller Data Site. UIUC Applied Aerodynamics Group, n.d. Web.

[51] Budynas, Richard G., J. Keith. Nisbett, and Joseph Edward. Shigley. *Shigley's Mechanical Engineering Design.* Boston: McGraw-Hill, 2008. Print.

[52] Wilson, Peter, and Tim Williams. The Circuit Designer's Companion. Amsterdam: Newnes, 2012. Print.

Appendix A

Sample Set of Electrical Schematics

Other electrical schematics available upon request.

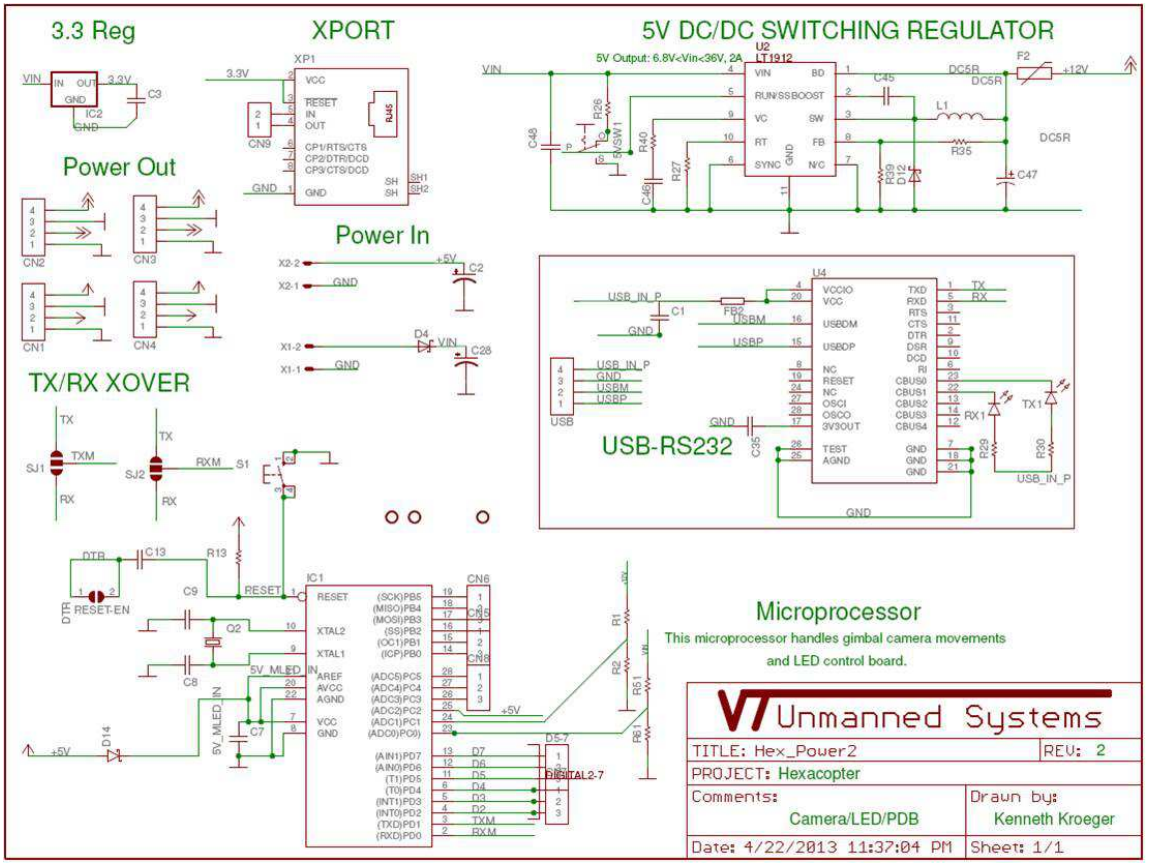


Figure A.1: Schematic of the HexaCopter gimbal and LED control circuity.

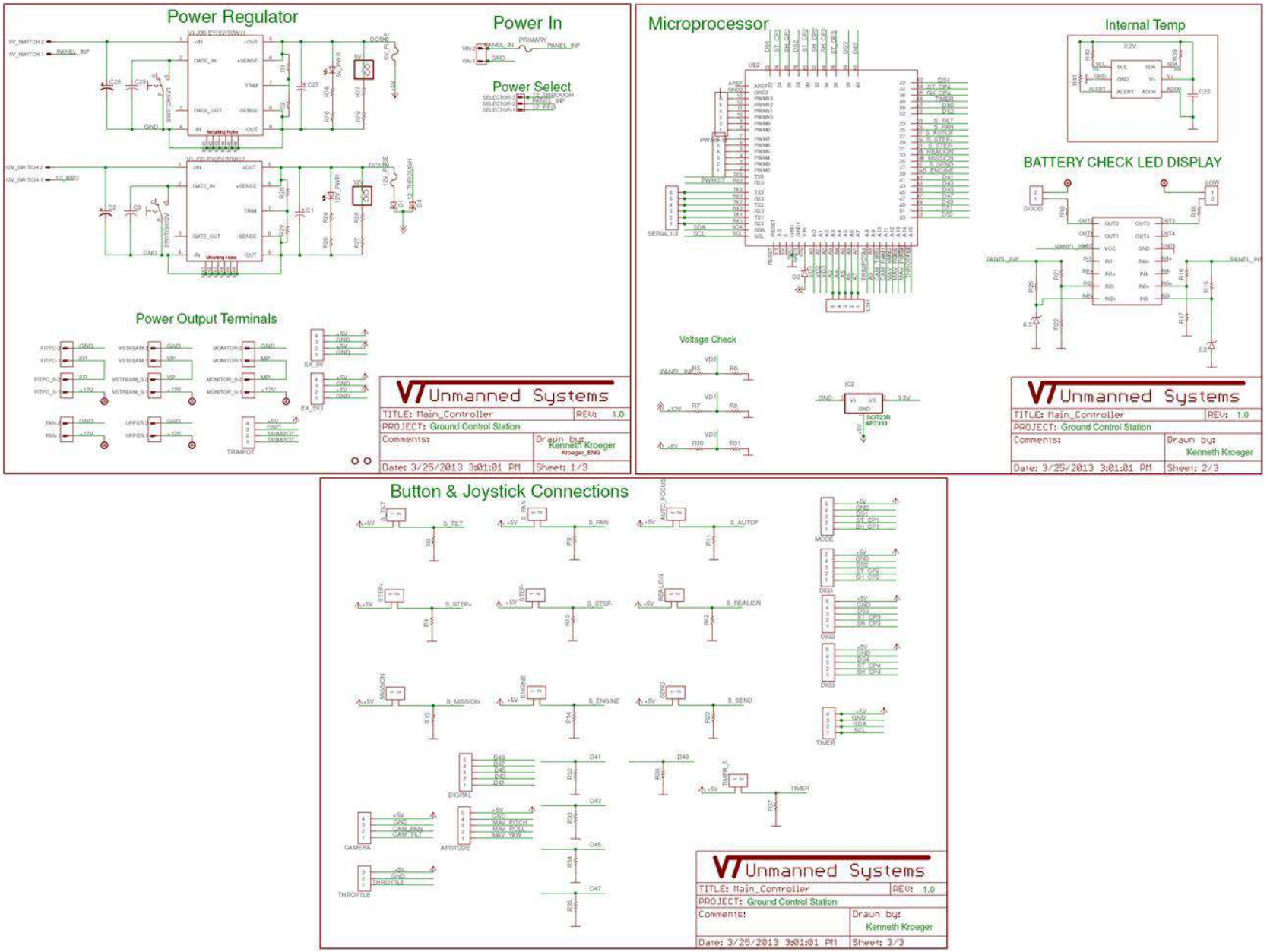


Figure A.2: Schematic of the UGCS base control.

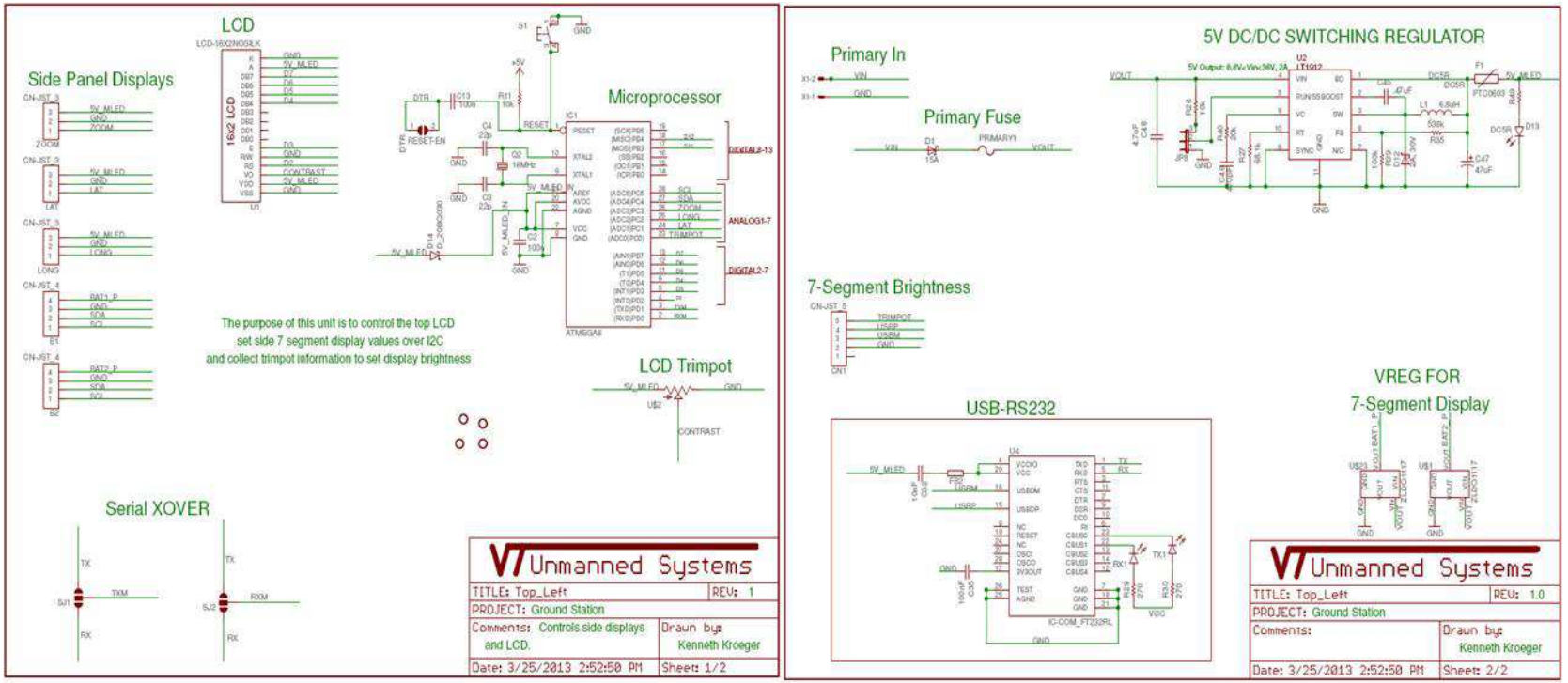


Figure A.3: Schematic of the UGCS top control.

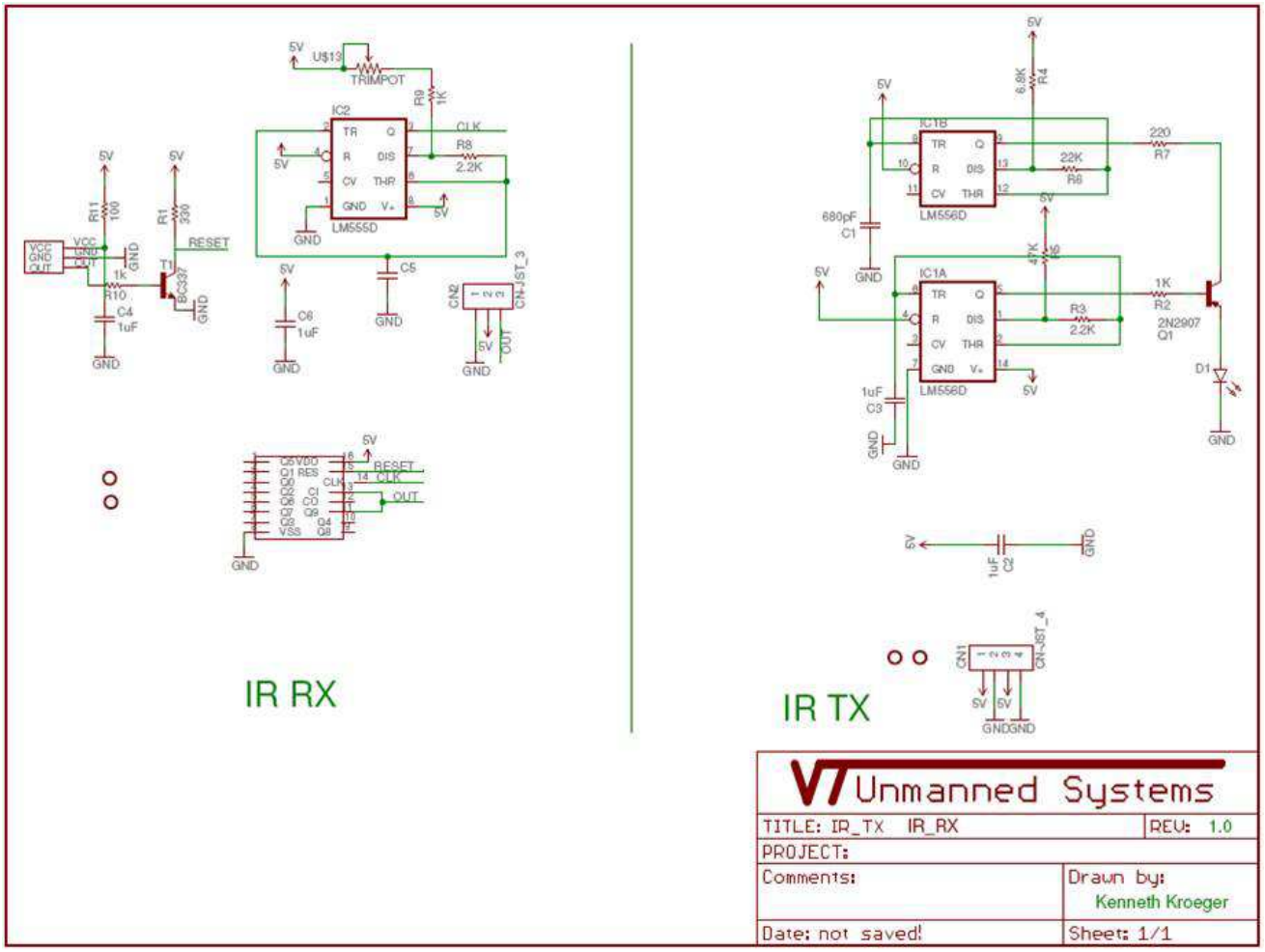


Figure A.4: Schematic of the IR TX and RX circuit developed for RPM testing of the motors.

Appendix B

Communication & Power Architecture

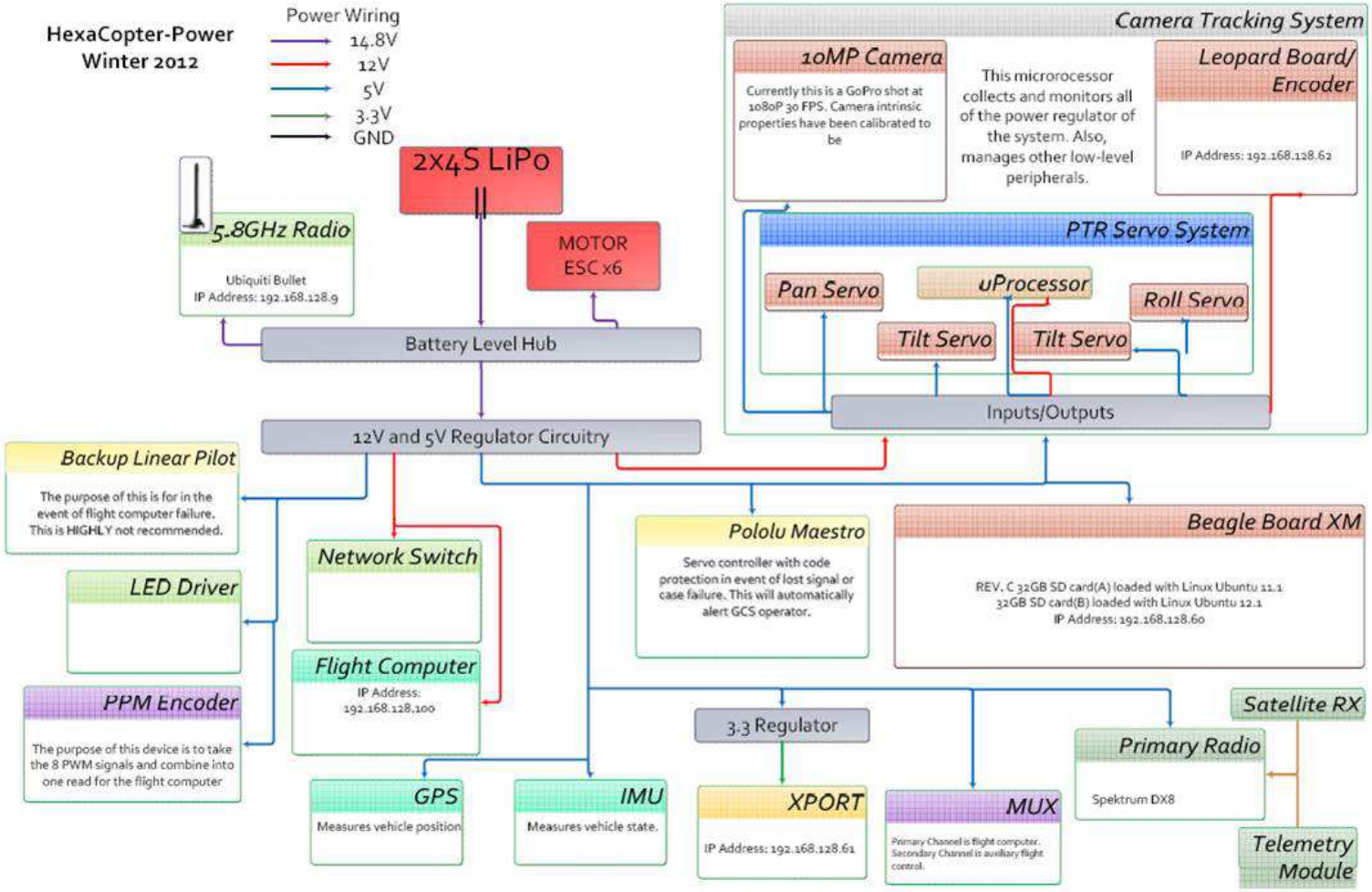


Figure B.1: HexaCopter power diagram.

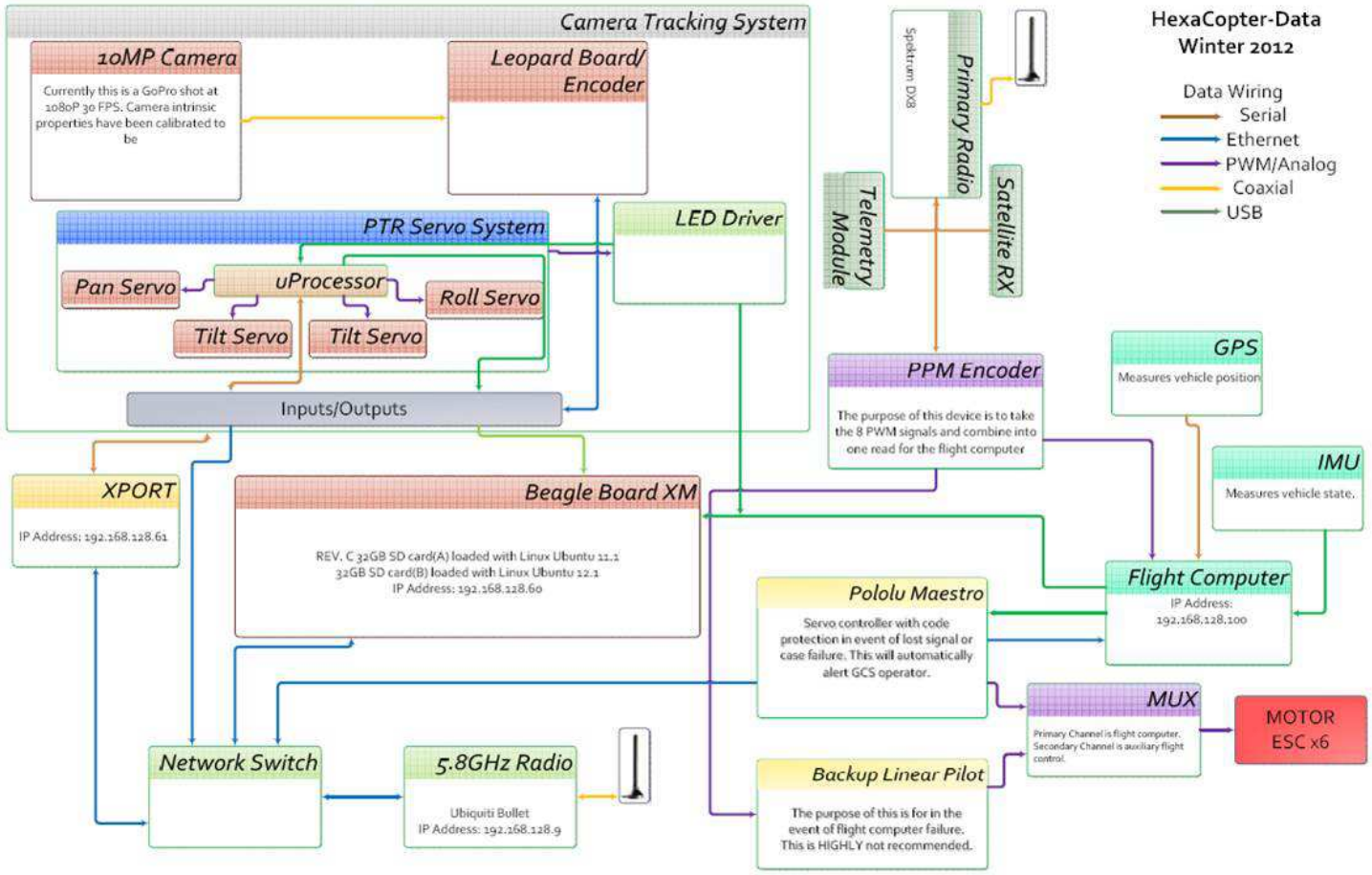


Figure B.2: HexaCopter data diagram.

Appendix C

Sample Set Simulink Files

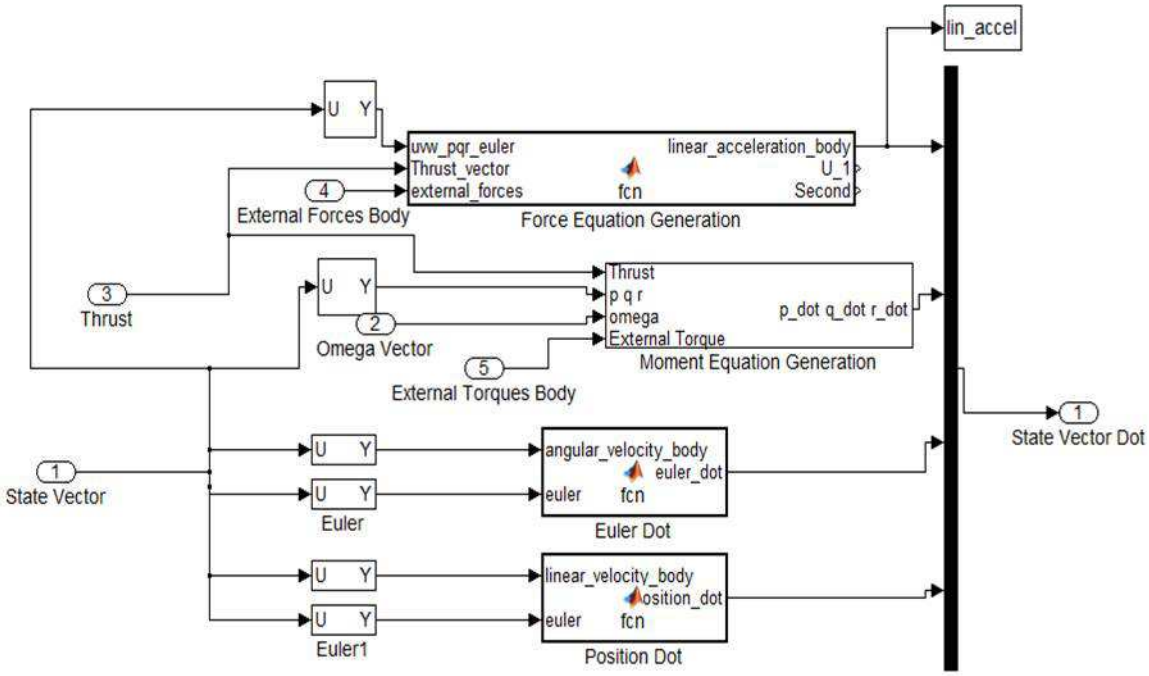


Figure C.2: Dynamic Model Blockset Internal Diagram.

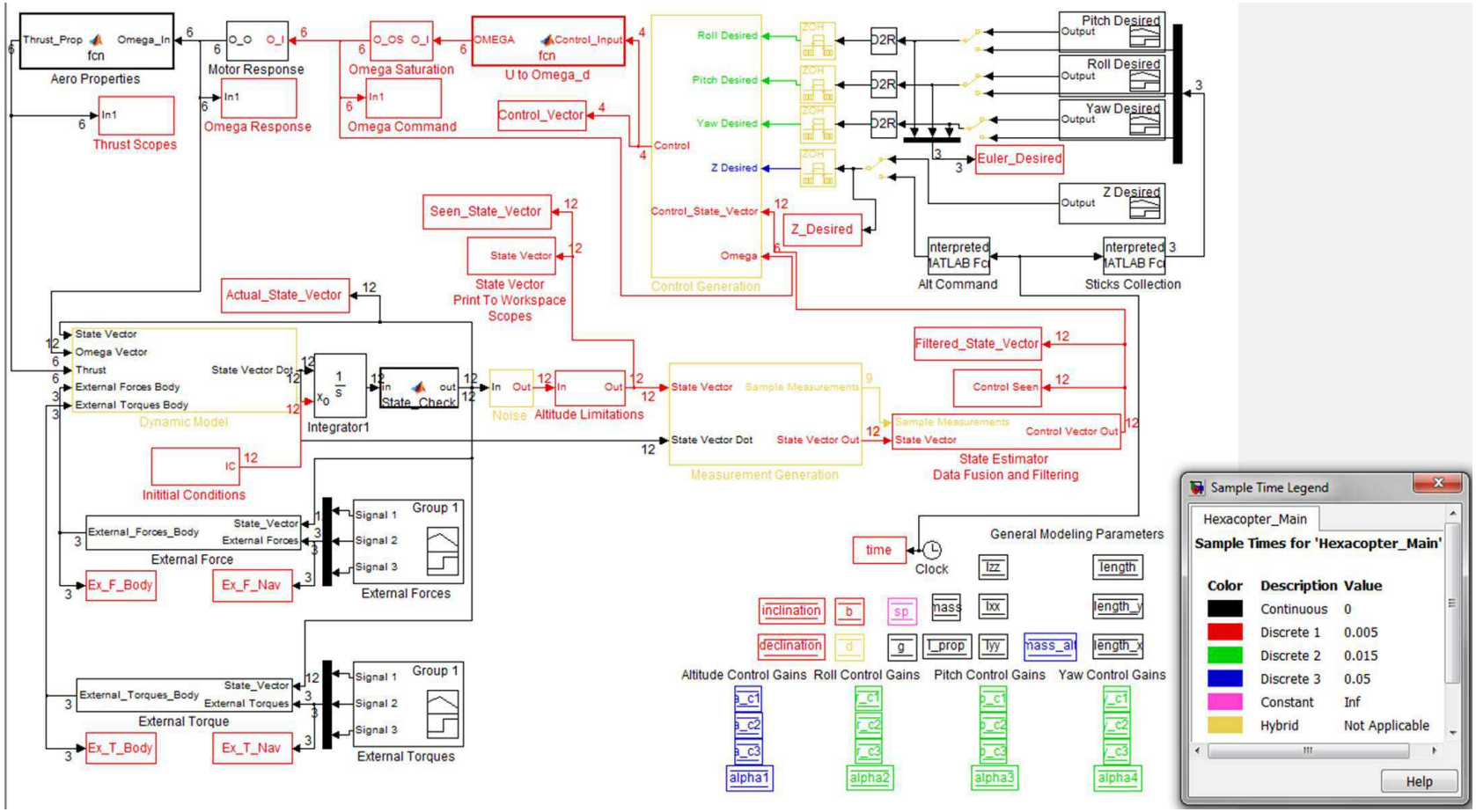


Figure C.1: Full Simulink model of HexaCopter UAV. The color coordination represents sample times of each component.

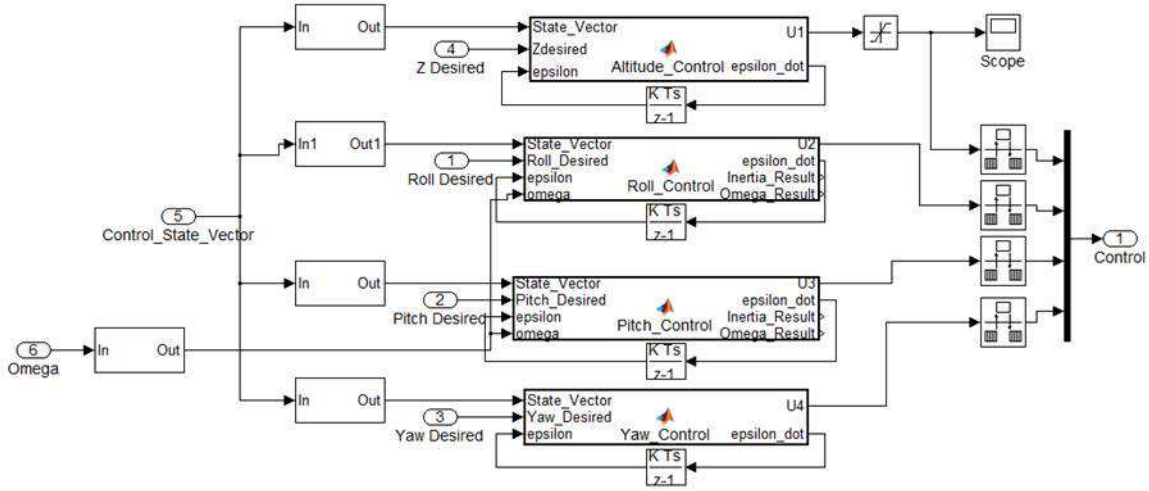


Figure C.3: Control Blockset Internal Diagram.

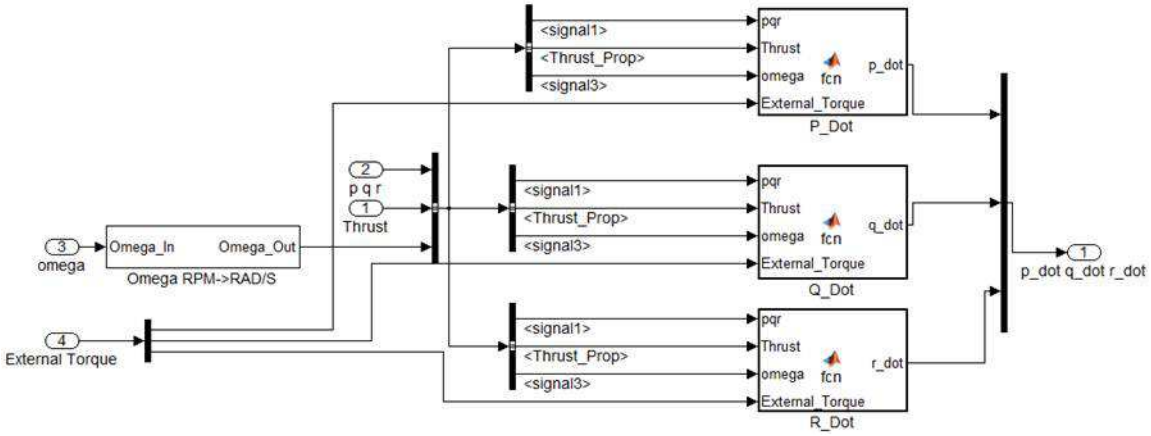


Figure C.4: Body Angular Blockset Internal Diagram.

Appendix D

Sample Set of Matlab Functions

Algorithm D.1 Example P_dot function.

```
function p_dot = fcn(pqr,Thrust,omega,External_Torque)
global Ixx; global Iyy; global Izz; global I_prop; global length; global length_x;
p=pqr(1); q=pqr(2); r=pqr(3); q_vector=[-q q -q q -q q];
Torque_prop_gyro=I_prop*(q_vector*omega); Torque_body_gyro = q*r*(Iyy-Izz);
Torque_roll = (Thrust(4)+Thrust(6))*length_x+length*Thrust(5)-(Thrust(1)+Thrust(3))*length_x-length*Thrust(2);
              p_dot=(Torque_body_gyro+Torque_prop_gyro+Torque_roll+External_Torque)/Ixx;
```

Algorithm D.2 Example determining body linear acceleration function.

```
function [linear_acceleration_body,U_1,Second] = fcn(uvw_pqr_euler,Thrust_vector,external_forces)
global g; global mass;
p=uvw_pqr_euler(1); q=uvw_pqr_euler(2); r=uvw_pqr_euler(3); u=uvw_pqr_euler(4); v=uvw_pqr_euler(5); w=uvw_pqr_euler(6);
roll=uvw_pqr_euler(7); pitch=uvw_pqr_euler(8); yaw=uvw_pqr_euler(9);
U_1=Thrust_vector(1)+Thrust_vector(2)+Thrust_vector(3)+Thrust_vector(4)+Thrust_vector(5)+Thrust_vector(6);
First=[(g*sin(pitch));(g*sin(roll)*cos(pitch));-(g*cos(roll)*cos(pitch))];
Second=[0;0;U_1/mass];
Third=[v*r-w*q;p*w-u*r;u*q-v*p];
Fourth=[external_forces(1);external_forces(2);external_forces(3)];
%linear_acceleration_body=[0;0;0];
linear_acceleration_body=First+Second+Third+Fourth;
```

Algorithm D.3 Example function determining euler_dot.

```
function euler_dot = fcn(angular_velocity_body,euler)
phi=euler(1,1);     theta=euler(2,1);     psi=euler(3,1);     p=angular_velocity_body(1);     q=angular_velocity_body(2);
r=angular_velocity_body(3);
avb=[p;q;r];
euler_dot_middle=[1     sin(phi)*tan(theta)     cos(phi)*tan(theta);0     cos(phi)     -sin(phi);0     sin(phi)/cos(theta)
cos(phi)/cos(theta)];
euler_dot=euler_dot_middle*avb;
```

Algorithm D.4 Example function determining local position_dot.

```
function position_dot = fcn(linear_velocity_body,euler)
phi=euler(1,1); theta=euler(2,1); psi=euler(3,1);
u=linear_velocity_body(1,1); v=linear_velocity_body(2,1); w=linear_velocity_body(3,1);
x_dot=u*cos(theta)*cos(psi)+(-cos(phi)*sin(psi)+sin(phi)*sin(theta)*cos(psi))*v+(sin(phi)*sin(psi)+cos(phi)*sin(theta)*cos
(psi))*w;
y_dot=u*cos(theta)*sin(psi)+(cos(phi)*cos(psi)+sin(phi)*sin(theta)*sin(psi))*v+(sin(theta)*cos(phi)*sin(psi)-sin(phi)*cos(
psi))*w; z_dot=-u*sin(theta)+v*cos(theta)*sin(phi)+w*cos(theta)*cos(phi);
position_dot=[x_dot;y_dot;z_dot];
```

Appendix E

Ideal Controller Simulation Result

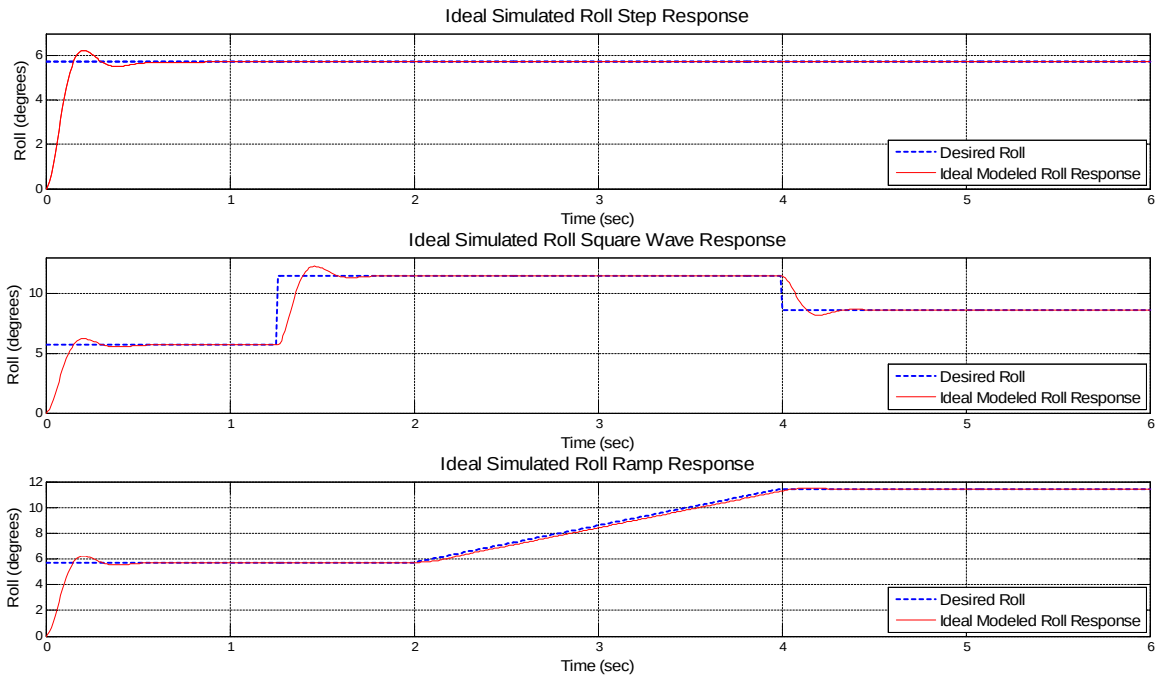


Figure E.1: Ideal simulated roll performance.

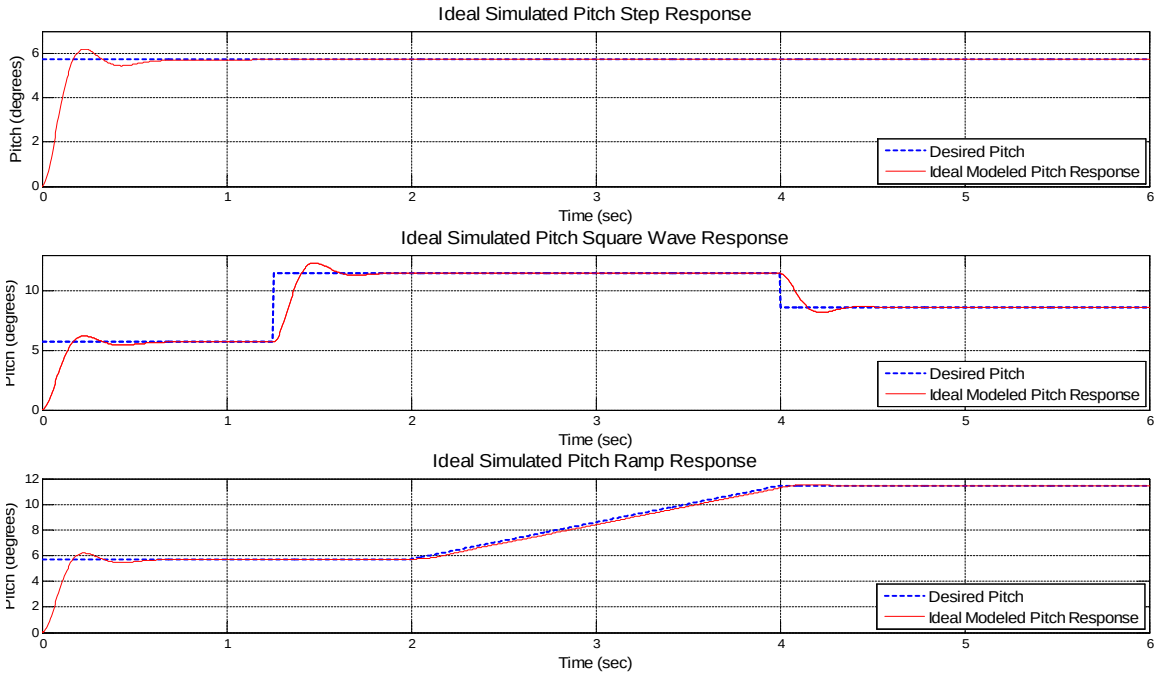


Figure E.2: Ideal simulated pitch performance.

Appendix F

Incoorporated Motor Model Simulation Result

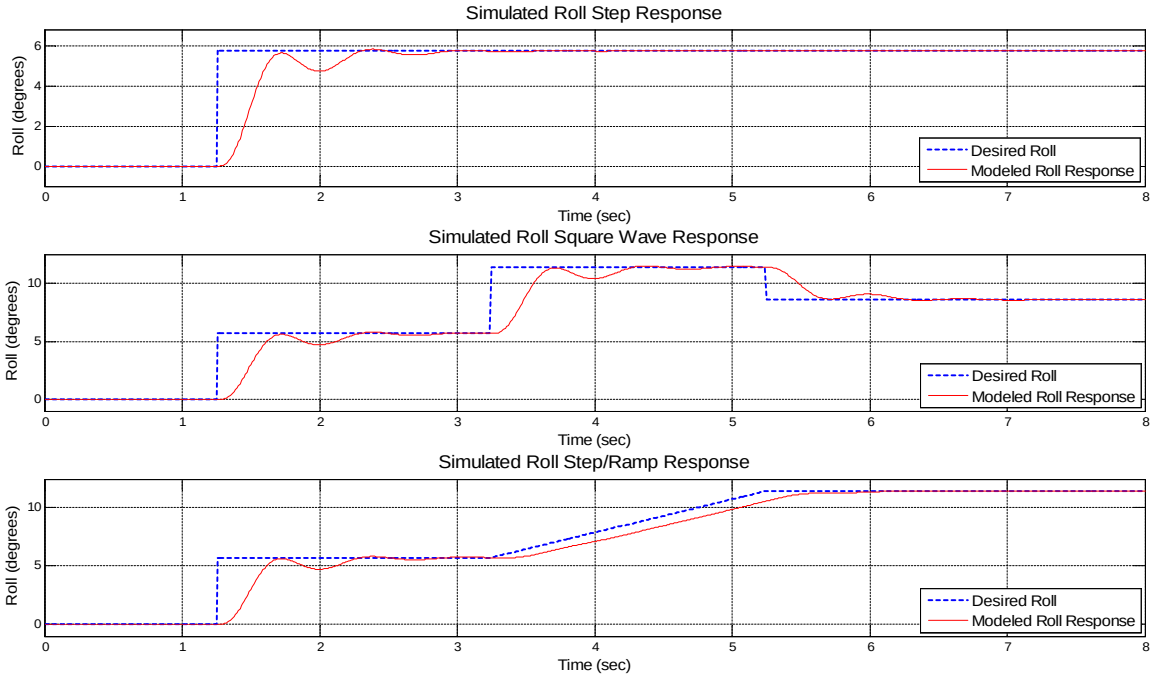


Figure F.1: Simulated roll performance with motor model incoorporated.

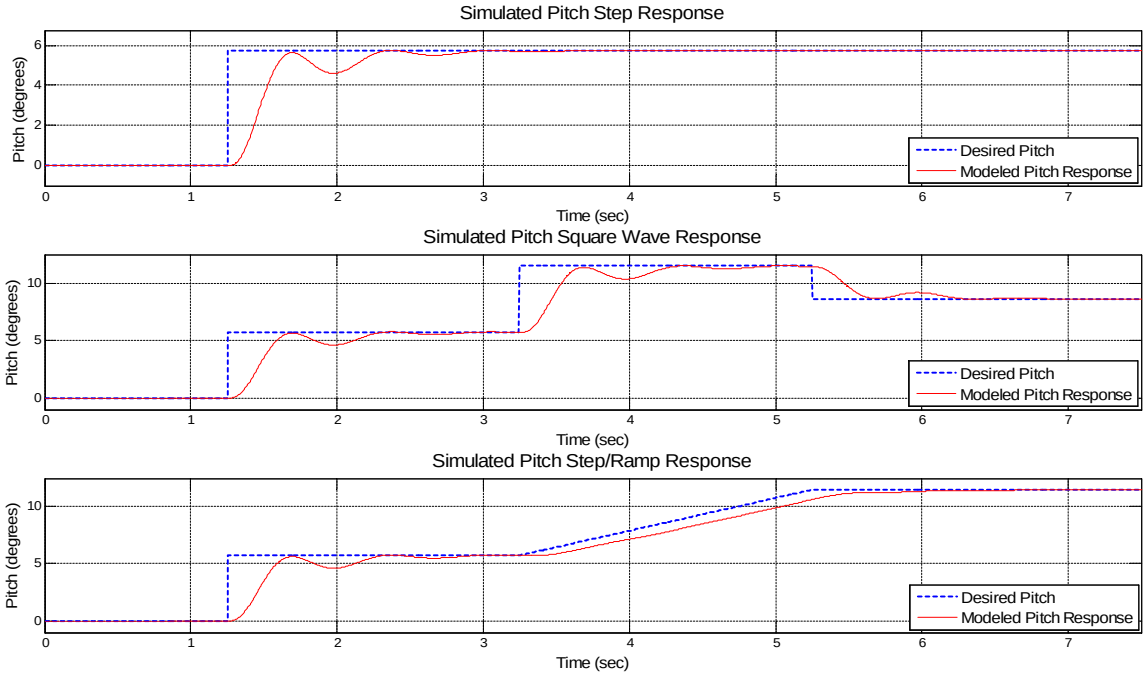


Figure F.2: Simulated pitch performance with motor model incoorporated.

Appendix G

Full Model Simulation Test Cases

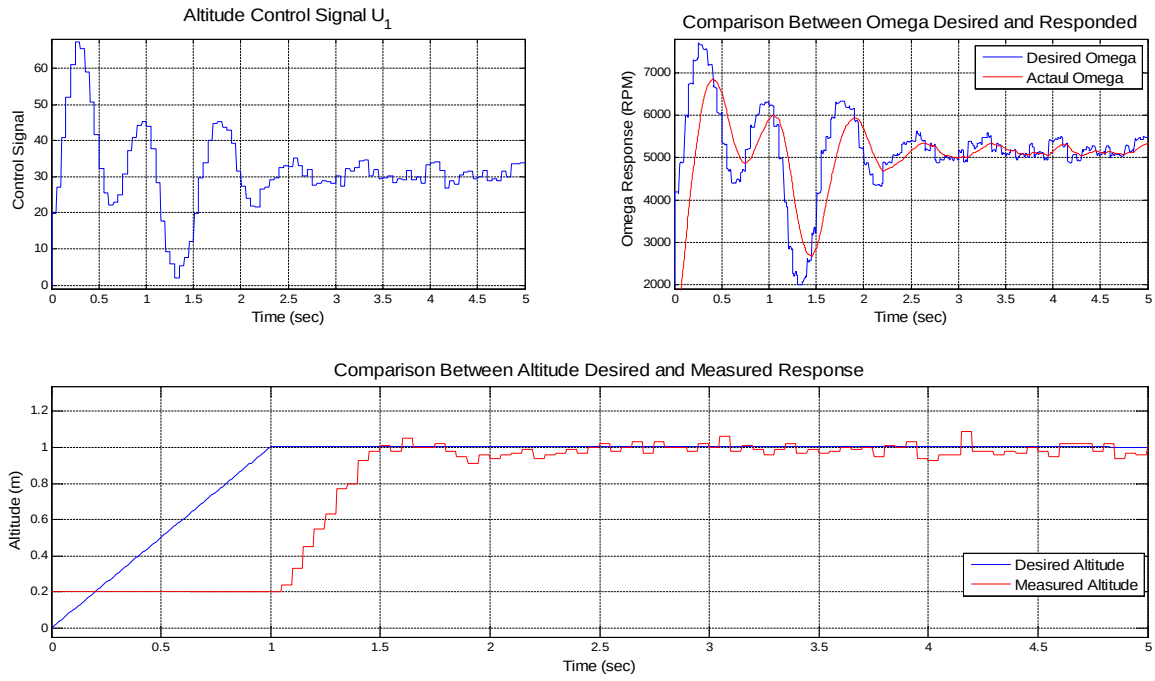


Figure G.1: Simulated performance of altitude control on the fuly realized model in Simulink.

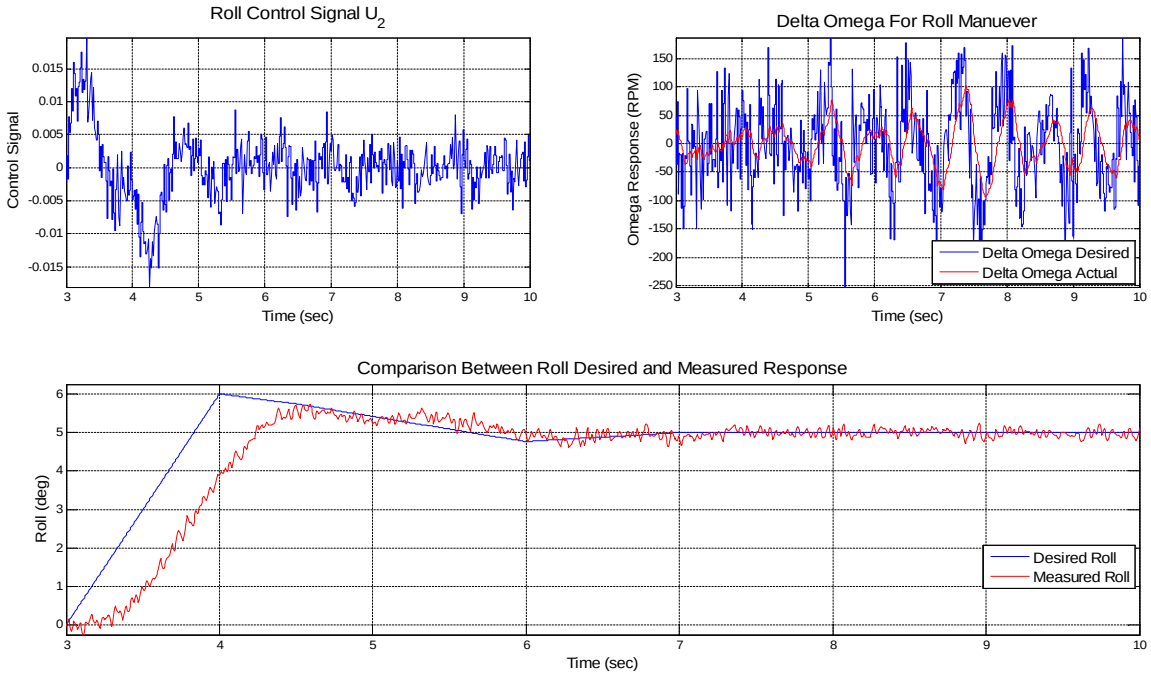


Figure G.2: Simulated performance of roll control on the fuly realized model in Simulink.

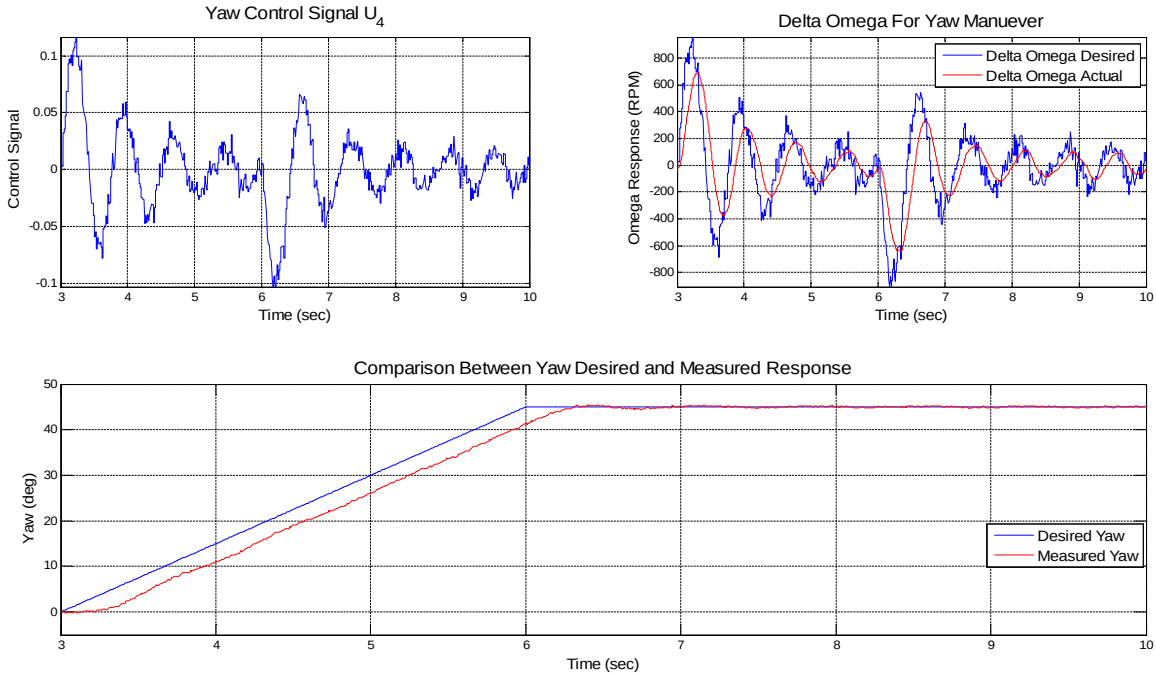


Figure G.3: Simulated performance of yaw control on the fuly realized model in Simulink.