

Polytechnic University of Puerto Rico
Electrical & Computer Engineering and Computer Science Department
FA18 (EE-4002) & WI18 (EE-4022)



Capstone Design Project
Honeywell Puerto Rico Navigation Challenge

Team Members
Amaris Vélez, #90013
María De León, #79250

Professor: Wenceslao López

Table of Contents

Sections	Pages
I. Abstract	003
II. Introduction	004-005
III. Realistic Constraints	006
IV. Multidisciplinary Aspects	007
V. Customer Memo	008-009
VI. Project Solution	
A. Alternatives Considered	010-034
B. Mathematical Justification	035-053
C. Testing Phase	054-065
D. Manufacturing and Wiring	066-069
E. System Specifications	070-082
F. Economic Analysis and Budget	083
VII. Project Results	084-087
VIII. Conclusions	088
IX. Bibliography & Reference Citation	089-091
X. Magazine Article	092-101
XI. Administrative Section and Appendix	102

Chapter 1: Abstract

This capstone project was originally proposed to complement the multidisciplinary team that was created as a result of Honeywell's First Annual Navigation Challenge where different universities from Puerto Rico were given the task of building an unmanned aerial vehicle (UAV) from scratch. The created UAV team required the following disciplines: Mechanical Engineering (ME), Computer Engineering (COE), and Electrical Engineering (EE). This report has been written by the teammates from the electrical engineering team. The main task of the electrical engineering team was to design a control system for the UAV to ensure a successful flight.

The software that was chosen to design the control system was MATLAB/Simulink. Through this software a control system for the UAV was successfully designed using the PID control technique. Thus, various Simulink models were implemented to simulate UAV flight. From another point of view, although the control logic could not be uploaded to the UAV's flight control hardware, the PID parameters for the UAV were successfully specified to the flight control hardware through the ground control station (GCS) known as Mission Planner which facilitated both manual and autonomous flight for the UAV. This report shows the complete design process for the UAV.

Este proyecto final fue propuesto originalmente con el fin de complementar el equipo multidisciplinario que fue creado como resultado del Primer Desafío Anual de Navegación de Honeywell, donde a diferentes universidades de Puerto Rico se les asignó la tarea de construir un vehículo aéreo no tripulado (VANT) desde cero. El equipo creado con el fin de diseñar un VANT requería de las siguientes disciplinas: ingeniería mecánica, ingeniería de computadoras e ingeniería eléctrica. Este informe ha sido escrito por los compañeros del equipo de ingeniería eléctrica. La tarea principal del equipo de ingeniería eléctrica fue diseñar un sistema de control para el VANT de manera que se pueda garantizar un vuelo exitoso.

El software que se eligió para diseñar el sistema de control fue MATLAB/Simulink. A través de este software, se diseñó con éxito un sistema de control para el VANT mediante la técnica de control conocida como PID. Por lo tanto, se implementaron varios modelos de Simulink para simular el vuelo del VANT. Desde otro punto de vista, aunque la lógica de control no pudo ser subida al hardware de control de vuelo del VANT, los parámetros PID para el VANT pudieron ser especificados con en el hardware de control de vuelo a través de la estación de control conocida como "Mission Planner", lo cual facilitó tanto el vuelo manual como el vuelo autónomo para el VANT. Este informe muestra el proceso completo de diseño para el VANT.

Chapter 2: Introduction

In this capstone project an unmanned aerial vehicle (UAV), also known as drone, was designed from scratch. A UAV or drone can be defined as an aircraft that does not carry a human pilot or passenger and is fully or partially autonomous. The term unmanned aerial system (UAS) is used when the selection of the GCS and communications unit is included in the design. The term UAS includes the whole system involved when operating a UAV. Thus, in truth, for this project, a UAS was developed since the vehicle must be designed along with an interface to interact with it and a communication scheme to facilitate this interaction.

The design competed with that of other universities as part of Honeywell's first annual navigation challenge competition. Initially, the UAV design had to integrate the HGuideN580 navigator that was going to be provided by Honeywell. Then, using Honeywell's navigator the UAV had to be capable of flying autonomously along a predetermined flightpath established through waypoints. According to Clough, this is considered as level 1 autonomy for the UAV since it will only have the capability of executing preplanned missions [1].

Apart from the type of autonomy, there is also a set of basic factors that must be taken into consideration for the UAV design which are weight, lift, drag, and thrust. According to these parameters, elements such as frame, motors, propellers, and batteries are identified, as well as some electrical components such as the flight controller and the electronic speed controllers (ESCs). Flight controllers are designed to assist UAV flight. However, they must be fed information from the user specifying the desired state or position to execute their function. The drone may interpret the desired position in two different ways depending on the mode in which it is operating. If it is in manual mode, then it will interpret the desired position by receiving values for throttle, roll, pitch, and yaw. If it is in autonomous mode, then it will interpret the desired position by receiving values for coordinates x, y, z. The drone must be able to interpret the desired position for every instance of time. According to this information, the flight controller will receive information from its navigator or sensing devices and send separate digital signals to each of the UAV's ESCs. Then, the function of the ESCs is to receive these digital signals known as pulse width modulations (PWMs) and control the revolutions per minute of the UAV's motors and power them at the same time. Since the UAV was designed from scratch, all these elements had to be individually selected. As a result, comparisons between products of the same type were executed and the compatibility between each of the selected components for the design was verified. This selection process is described in more detail through the project solution section in this report.

From another point of view, as it may be observed, this is a multidisciplinary project. As a result, the tasks necessary to complete the UAV design were divided into three main disciplinary teams: ME, COE, and EE. This report has been written by the EE team members whose main task was to design the UAV's control system to ensure proper maneuver for unforeseen disturbances. Knowledge of MATLAB/Simulink was particularly important for this project since the control system was designed through this software by taking into consideration the four basic elements that make up a control system: the sensor, the transmitter, the controller, and the final control element.

It is important to mention that halfway through the competition, the design requirements were changed by Honeywell since they would not be able to provide the HGuideN580 navigation units to the competing teams. Due to this occurrence, Honeywell decided to host the navigation challenge for remotely controlled drones instead of autonomous ones. Thus, all requirements relating to autonomous flight were dissolved from the competition. However, the UAV team continued to pursue the design of an autonomous UAV as shown in the following sections.

Chapter 3: Realistic Constraints

Students:

- Some of the students in the multidisciplinary competition team were not enrolled in the Capstone course. Therefore, there was the risk that the students would not be fully committed to the project. Thus, we had the challenge of motivating them to execute their tasks efficiently and effectively.
- Lack of knowledge concerning the proper design process to produce UAV from scratch.

Limited Budget:

- Honeywell provided an initial budget of \$2,000. However, the total cost of the project depended on the hardware and software tools that were used. Therefore, the request for additional equipment had to be approved by the Polytechnic University of Puerto Rico (PUPR).

Reliability:

- Mission time was less than 15 minutes and this time was dependent on the proper selection of the components to be implemented in the design.
- Communication between the GCS and the UAV was considered critical and the UAV's flight would be complicated if this communication could not be achieved.
- UAV mission was depended on how well the design took into consideration factors such as stable flight, safety controls, and a resistance to impact collisions.

Third Party:

- The flying coordinates were provided the day of the competition by Honeywell and they were subject to change.
- Honeywell specifications were subject to unexpected changes.

Safety and Health:

- The safety of the users and others around the drone was paramount. The drone design needed to meet the performance specifications as indicated in the competition rules and objectives previously stated.

Timing and Logistics:

- Most of the equipment components needed to be purchased and delivered.
- Timely installation and integration of all component was necessary to be able to perform the required tests.
- Compliance with Honeywell deadlines deliverables was mandatory.

Chapter 4: Multidisciplinary Aspects

The design, development, and manufacture of a UAV is supported by the study area of mechatronics. This area involves three engineering disciplines: ME, COE, and EE. This type of project also requires the use of systems engineering to properly identify and structure the UAV system and sub-systems to be designed. Additionally, the implementation of systems engineering will ensure adequate project progression. The following points briefly specify the roles that were given to each of the mentioned disciplines:

Systems Engineering:

This area of study was necessary due to Honeywell's requirement to present project progress according to the NASA Systems Engineering Handbook (NASA/SP-2007-6105, Rev1).

Mechanical Engineering:

This area of study was necessary to create the UAV's frame and to execute the analysis of the mass and structure of the UAV with its selected mechanical and structural components.

Computer Engineering:

This area of study was necessary to develop software through which the UAV's missions would be entered and stored. Also, users had to be able to monitor mission progress through this software. In general, the software had to ensure the following:

- Air vehicle takeoff and landing
- Interface to input mission coordinates
- Calculation of path from user input

Electrical Engineering:

This area of study involved multiple sub-disciplines in EE which are controls, power, electronics, and communications. The following points describe the application of each sub-discipline:

- Controls: Design the UAV's control system to ensure stable flight
- Power: Execute load analysis on the UAV system.
- Electronics: Evaluate electronic components to be added to protect the system.
- Communications: Establish communication link between UAV, ground control station, and radio controller.

Test Engineering:

This area of study was necessary to be able to properly assess the performance for each selected UAV component and evaluate the designed UAV system after integrating developed software with hardware.

Software:

The design of the UAV required the use of the following software:

- Fusion 360
- Matlab/Simulink
- Mission Planner

Chapter 4: Customer Memo

Polytechnical University of Puerto Rico
Electrical Engineering Department

May 6, 2019

TO WHOM IT MAY CONCERN:

I hereby confirm that the students Amaris Vélez, #90013 and María De León, #79250 were under my supervision as members of the multidisciplinary team created to represent the Polytechnic University of Puerto Rico in Honeywell's First Annual Navigation Challenge which took place during the 2018-2019 academic period.

The placements of the students in the multidisciplinary team was coupled with their capstone project. As a result, I am able to confirm that they have responsibly and successfully fulfilled the project requirements that led to the manufacture and development of an unmanned aerial vehicle.

Thank you so much for helping the institution succeed in its endeavors!

Sincerely,

Diolinete Gerena Colón
Associate Professor
Department of Mechanical Engineering

Chapter 6: Project Solution

A. Alternatives Considered

As previously stated, this project required a multidisciplinary team. As a result, each sub-team focused on selecting the UAV elements relative to their respective discipline. The electrical engineering team focused on selecting the UAV's flight technique, power supply, control technique and flight controller. The computer engineering team focused on selecting the UAV's interface to be developed. Finally, the mechanical engineering team focused on selecting the UAV's motors, propellers, and frame according to the identified UAV weight and required thrust.

Multi-copter Technology

The term multi-copter is used for any rotorcraft containing more than two rotors. Multi-copters facilitate an aircraft's flight control and they are generally implemented in the design of radio-controlled aircraft and UAVs. There are many types of multi-copter to choose from and these are categorized by the number of motors used. The most common are the tricopter, quadcopter, hexacopter, and octocopter [2]. To select the type of multi-copter to be implemented in the project, each of the mentioned multi-copter will be evaluated. It is important to recognize that there is no such thing as the best multi-copter configuration since the selection is based on the aircraft's needs. While the increase in number of used motors increases the aircraft's lift capacity and redundancy, it also decreases the aircraft's power efficiency and increases manufacturing and maintenance costs [3].

Tricopter evaluation

This type of multi-copter has 3 motors positioned 120 degrees apart in a "Y" shape as shown in Figure 6A.1. Motors 1 and 2 are positioned in the front arms and motor 4 is position in the back arm alongside motor 3 which is a servo motor whose function is to enable the yaw mechanism in the tricopter. This is a relatively cheap configuration that is very good at yawing but, it is less generally stable than other motor configurations and has the lowest lifting capability [3].

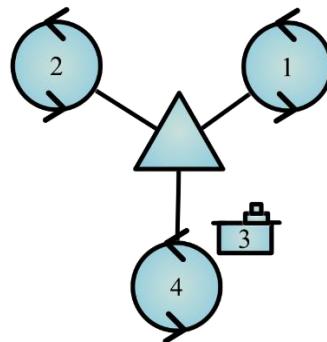


Figure 6A.1: Tricopter diagram

Quadcopter evaluation

This type of multi-copter has 4 motors positioned 90 degrees apart in a cross-style or plus-style configuration as shown in Figure 6A.2. It is composed of two sets of identical propellers. One set rotates in a clockwise manner and the other rotates in a counterclockwise manner. This is the most popular configuration due to its simplicity and flexibility in terms of frame options. Additionally, this type of multi-copter does not require servos since multi-copter with an even number of counter-rotating rotors uses rotor torque to control yaw.

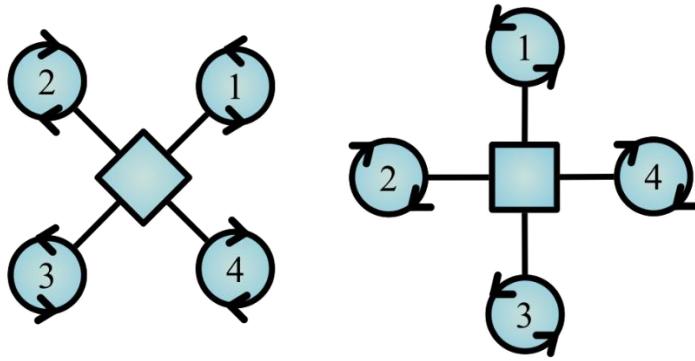


Figure 6A.2: Quadcopter diagrams showing cross-style in the left and plus-style in the right

Hexacopter evaluation

This type of multi-copter has 6 motors positioned 120 degrees apart as shown in Figure 6A.3. It is composed of three sets of counter rotating (clockwise and counterclockwise) motors. This type of multi-copter is like the quadcopter but, it has higher lifting capabilities, slightly better yaw control and redundancy so that if one motor fails, the aircraft can still land securely. From another point of view, the disadvantages of implementing a hexacopter are the increased size of the aircraft due to the positioning of the motors and the increased cost as compared to the tricopter and quadcopter.

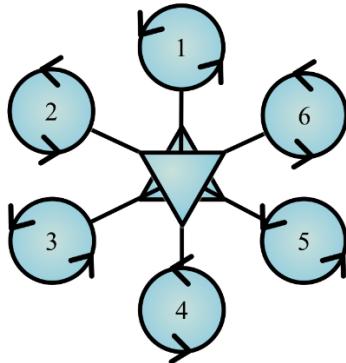


Figure 6A.3: Hexacopter diagram

Octocopter evaluation

This type of multi-copter has 8 motors as shown in Figure 6A.4. The octocopter is very similar to the hexacopter. However, due to the increased number of used motors, this type of multi-copter has the highest lifting capabilities and is considered the most reliable motor configuration because if one, two motor, or an ESC fails, the aircraft can still land securely. On the other hand, the disadvantages of implementing the octocopter are the increased power demand, increased size of the aircraft and the increased cost as compared to the tricopter, quadcopter, and hexacopter.

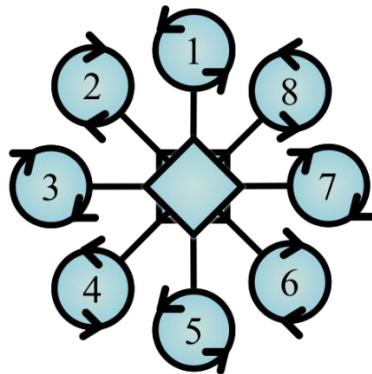


Figure 6A.4: Octocopter diagram

After evaluating each of the previously mentioned multi-copter. The quadcopter was selected as the type of multi-copter to be implemented for the project. This type of multi-copter was selected due to its mechanical simplicity and popularity among UAV designers and enthusiasts which allows to use a greater amount of references for the design of the UAV.

Selecting quadcopter configuration

Now that the quadcopter technology has been selected, one must determine the quadcopter configuration to use for the two sets of propellers. There are two styles in which these sets of propellers may be configured as it was specified in the quadcopter evaluation section: the cross-style and the plus-style [4]. The plus-style configuration changes the aircraft's attitude (roll, pitch, or yaw) by varying the speed of two of its motors. On the other hand, the cross-style configuration changes the aircraft's attitude by varying the speed of all its motors which produces higher momentum in the aircraft and thus, improved maneuverability over the plus-style configuration [4]. For this reason, the configuration that was selected for the quadcopter was the cross-style configuration.

Flight Mechanism

In the cross-style configuration, the body x and y axis for the quadcopter are tilted 45 degree with respect to the quadcopter's arms. To generate thrust in this configuration, the four motors are set to rotate with the same rotational speed along the vertical z-axis. To perform the roll maneuver, the rotational speed of motors 1 and 4 are reduced while the rotational speed of motors 2 and 3 are increased which generates a torque along the x-axis (τ_Φ). To perform the pitch maneuver, the rotational speed of motors 1 and 2 are increased while the rotational speed of motors 3 and 4 are reduced which generates a torque along the y-axis (τ_θ). Finally, to perform the yaw maneuver, the rotational speed of motors 2 and 4 are reduced while the rotational speed of motors 1 and 3 are increased which generates a torque along the z-axis (τ_Ψ). The different flight mechanisms that have been described are shown in Figure 6A.5. On the other hand, the mathematical considerations to produce these maneuvers will be presented in the Mathematical Justification section.

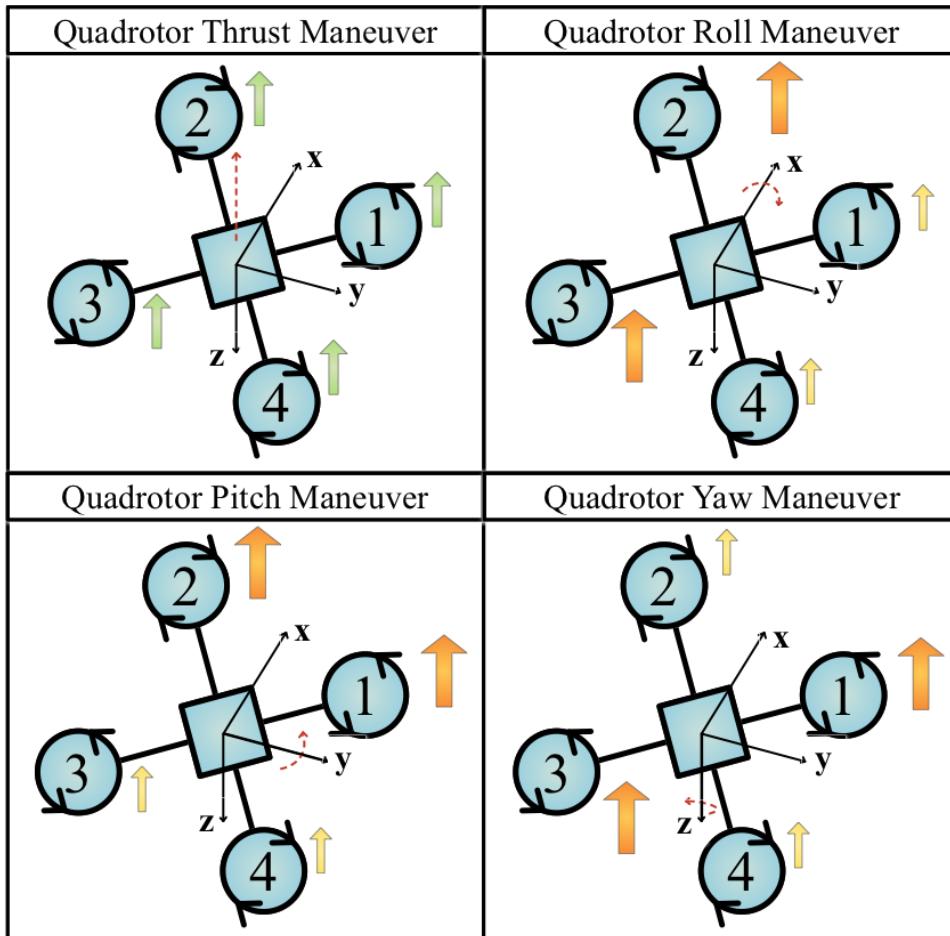


Figure 6A.5: Flight mechanisms for quadcopter with cross-style configuration

Quadcopter Flight Control Design

Every system has a set of inputs and outputs. Then, the objective of control system engineers is to identify necessary inputs to produce the desired outputs in a system known as the setpoints. However, to properly execute this operation, one must mathematically characterize the process that is present in the system. Through the mathematical model of the process one can determine the inputs to place into the system and the controller to design to obtain the desired outputs.

There are two types of control systems: Open-loop control and closed-loop control. In the open-loop control there is no feedback and thus, the system has no way of determining whether the designed controller is producing the desired outputs. On the other hand, the closed-loop control has feedback which is obtained by integrating sensors into the system. These sensors identify the outputs in the process to be

This type of control makes it possible for the system to constantly adjust according to the outputs it is producing; ensuring desired outputs established by what is known as the setpoint. A diagram for a typical control system is presented in Figure 6A.6.

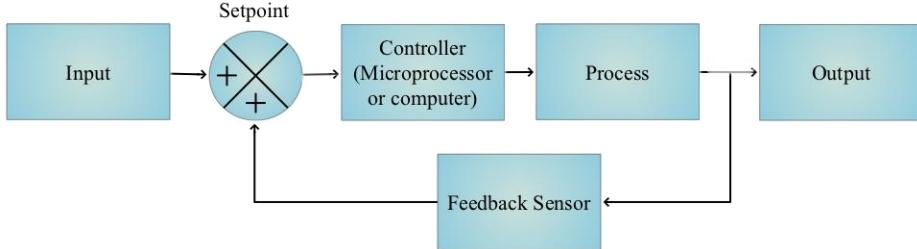


Figure 6A.6: Typical closed-loop control system

In our project the inputs for the control system come from the user interface and the remote controller depending on whether the quadcopter is set to run on autonomous or manual mode. The controller is the processing unit to be implemented in which the designed control system is installed. Then, the process is the quadcopter itself. Finally, the output is the pulse width modulation (PWM) to be inputted to each motor and produce the required thrust in each of them to achieve the desired quadcopter state. Through the quadcopter's sensors the current state of the quadcopter is measured and feedback to the control system producing its setpoint.

Selecting control technique

There are many different control techniques to choose from. Some of the implemented control techniques mentioned in references where a control system is designed for a quadcopter are the linear quadratic regulator (LQR) controller, linear quadratic gaussian (LQG) controller, fuzzy logic controller, and proportional integral derivative (PID) controller. The LQR controller is concerned with determining the control scheme for dynamic systems described by a set of linear differential equations. It can provide for optimal feedback control by implementing a cost function and specifying its parameters [5]. The cost function specifies the deviations in important measurements related to the system's state which is feedback to the controller and according to this function the LQR controller adjusts its control action. On the other hand, the LQG controller is the combination of a Kalman filter and a linear quadratic regular [6]. Thus, it functions very similarly to the LQR controller. From another point of view, there is the fuzzy logic controller which is based on a series of if-then rules to produce a “good enough” controller. The fuzzy controller is beneficial since it does not require the user to have extensive understanding of the equations that govern the system they are controlling, instead they are required to train controller [7]. The if-then rules are determined through this training process. Finally, there is the popular PID controller which is one of the most commonly used control strategies for quadcopters. It continuously measures the error signal (e) identified through its setpoint which is equal to the difference between the desired output and the actual output. Then, the control signal (u) is identified through the following equations:

$$\text{Function of a PID controller: } u(t) = K_p e(t) + K_i \int e(t) dt + K_p \frac{de}{dt}$$

Then, the tuning for this control system will be executed by iteratively changing the parameters of each PID controller and observing its respective step response. The following facts were taken into consideration while changing PID parameters:

1. Adjust P to get fast response, but you will get oscillation at steady state.
2. Adjust I to get rid of oscillation, but you will get overshoot.
3. Adjust D to lower overshoot.

Selecting modelling technique

To design the control system for a quadcopter one must have knowledge of the different control techniques that may be implemented. Additionally, one must be able to properly model the quadcopter. There are two ways in which one may pursue the modeling phase when designing a control system for a quadcopter. One may choose to model a single brushless motor and evaluate the step response of each PID controller (roll, pitch, and yaw) for a set of brushless motors [8]. On the other hand, one may also model the complete quadcopter by taking into consideration both motor and rigid body dynamics.

If one decides to simply model the brushless motors to be implemented, the following equations would be used:

$$1. \text{ Torque} = \frac{P_w \times 9.554}{n}$$

$$2. \text{ Torque constant: } k_t = \frac{\text{Torque}}{\text{Current}}$$

$$3. \text{ Phase value of the EMF constant aka electrical torque: } k_e = k_t \times 0.0605$$

$$4. \text{ Mechanical constant: } \tau_m = \frac{j \times 0.004 \times R}{k_e \times k_t}$$

$$5. \text{ Electrical constant: } \tau_e = \frac{l}{0.004 \times R}$$

Finally, once the values from equations 1-5 have been determined. The transfer function for the brushless motor may be specified as follows:

$$\text{BLDC Motor Transfer Function: } G(s) = \frac{\frac{1}{K_e}}{\tau_m \tau_e s^2 + \tau_m s + 1}$$

These equations seem straightforward, but they require that the following physical parameters concerning the motor be identified from the manufacturer or measured: RPM (n), Power (P_w), Electric resistance in one phase (R), Electric inductance (l), Maximum current (I), Moment of inertia of the rotor (j). Since the electrical inductance of the brushless motor was not identified, the researchers chose to pursue the modelling of a complete quadcopter which may show a more realistic representation of the process to be controlled.

To model a quadcopter, one must understand that the quadcopter is an underactuated system since it has 4 motors and 6 degrees of freedom (6DOF). The directions in which the quadcopter may move are divided into 2: translational directions (up/down, left/right, forward/backward) and rotational directions (roll, pitch, yaw). Rotation and thrust is then coupled to accomplish the goal of control over a quadcopter's movement. This process is shown in the mathematical justification section of this report where the complete model of the quadcopter is specified.

Quadrotor Components

Besides the quadcopter flight control design other important considerations for this project are the physical components that must be acquired to execute the physical implementation of the design. Thus, an evaluation must be done to select the following components: quadcopter control board, sensors and additional peripherals, remote control and receiver, motors, electronic speed controllers, propellers, battery and power management board.

Selection of UAV control board

Initially three processing units were considered for the physical implementation of the control system which would be designed through MATLAB. These considered processing units were the Arduino Nano development board, the NVIDIA Jetson TK1, and the ARM Cortex-M4F LaunchPad.

The Arduino is an open-source platform that facilitates the development of projects in electronics. This platform consists of a physical programmable circuit board (micro-controller) and an integrated development environment (Arduino IDE) that runs on a computer. From the Arduino IDE, code can be written and uploaded to the Arduino board. However, this can also be done through MATLAB and Simulink. There are different types of Arduino boards. The one that is being considered for this project, due to its portability, is the Arduino Nano board shown in Figure 6A.7.

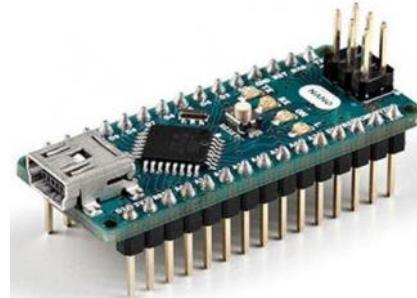


Figure 6A.7: Arduino Nano

The Arduino Nano board has the following features:

- Dimensions: 0.73" x 1.70"
- Microcontroller ATmega328
- Digital I/O Pins: 14 (6 provide PWM output)
- Analog Input Pins: 8
- Flash Memory 32 KB (ATmega328) of which 2 KB used by bootloader
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Clock Speed: 16 MHz
- Operating Voltage (logic level): 5 V
- Input Voltage (recommended): 7-12 V
- Input Voltage (limits): 6-20 V
- DC Current per I/O Pin: 40 mA
- Powering methods: via Mini-B USB connection, via pin 30 (6-20V unregulated external power supply), or via pin 27 (5V regulated external power supply)

The Jetson TX1 Developer Kit is a full-featured development platform with high computational capabilities. It has many of the advanced features that are included in modern desktops and runs in the Linux environment. It has support for many common APIs, and additionally, it is supported by NVIDIA's complete development tool chain. The board comes with various standard hardware interfaces which enables developers to work with a highly flexible and extensible platform.



Figure 6A.8: Jetson TX1 Developer Kit

The Jetson TX1 module has the following features:

- NVIDIA Maxwell™ GPU with 256 NVIDIA® CUDA® Cores
- Quad-core ARM® Cortex®-A57 Processor
- 4 GB LPDDR4 Memory
- 16 GB eMMC 5.1 Flash Storage
- 10/100/1000BASE-T Ethernet
- Power On/Off button
- Reset button
- Force Recovery button
- User-Defined button
- USB 3.0 Type A
- USB 2.0 Micro AB (supports recovery and host mode)
- HDMI
- M.2 Key E
- PCI-E x4
- Gigabit Ethernet
- Full-Size SD
- SATA Data and Power
- GPIOs, I2C, I2S, SPI
- TTL UART with Flow Control
- Display Expansion Header
- Camera Expansion Header
- Power options: External 19V AC adapter

The Tiva C Series TM4C123G LaunchPad Evaluation Board is a low-cost evaluation platform for ARM Cortex M4F based micro-controllers. The ARM Cortex M4 processor is a 32-bit high performance embedded processor. Some of its most attractive features are the USB 2.0 Device interface, hibernation module, motion control PWMs and overall cost effectiveness. It also features programmable user buttons and an RGB LED for custom applications. Additionally, it contains stackable headers that facilitate the expansion of its functionalities when interfacing to other peripherals with MSP430™ and another TI MCU BoosterPacks.

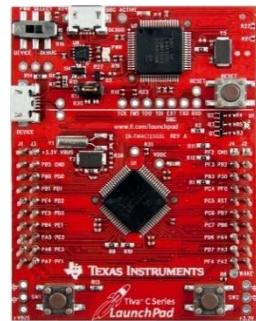


Figure 6A.9: Tiva C Series Board

The Tiva C Series TM4C123G LaunchPad Evaluation Board has the following features:

- Tiva C Series TM4C123GH6 microcontroller
- USB Micro-AB connector
- RGB user LED
- Two user switches (application/wake)
- Available I/O brought out to headers on a 0.1" grid
- On-board In-Circuit Debug Interface (ICDI)
- Switch-selectable power sources (ICDI and USB device)
- Reset switch
- Preloaded RGB quick start application
- Supported by TivaWare™ for C Series software including the USB library and the peripheral driver library
- Tiva C Series TM4C123G LaunchPad BoosterPack XL interface which features stackable headers to expand the capabilities of the 40-pin Tiva C Series LaunchPad evaluation platform

Initially, it was assumed that Honeywell would not permit the implementation of the Pixhawk or Ardupilot flight controllers. For this reason, the development boards shown in Figure 6A.8 and 6A.9 are compared with the Arduino Nano which was initially seen as the most viable board. MATLAB contained support packages for that board specifically. The final decision of selecting the Arduino Nano board was reached through decision matrix shown in Table 1. This decision matrix was presented to Honeywell during the Preliminary Design Review (PDR).

Table 6A.1: PDR decision matrix

Problem Statement	Choose a processing device to embed Simulink control system model and control motors		Weight Percentage	Arduino Nano	NVIDIA Jetson TX1	ARM Cortex-M4F LaunchPad
Criteria	Simulink Integrability	40.00%	Datum	-1	0	
	Hardware compatibility	40.00%		1	0	
	Ease of use	10.00%		-1	-1	
	Energy Consumption	3.33%		-1	0	
	Volume Constraints	3.33%		0	0	
	Weight	3.33%		0	0	
	Total	100.00 %		-0.53	-0.10	

Legend	
1	Improvement
0	Same
-1	Not Acceptable

From another point of view, the Computer Engineering team selected the Arrieta G25 board to develop the interface between the user and the Arduino. Then, the following schematic was developed:

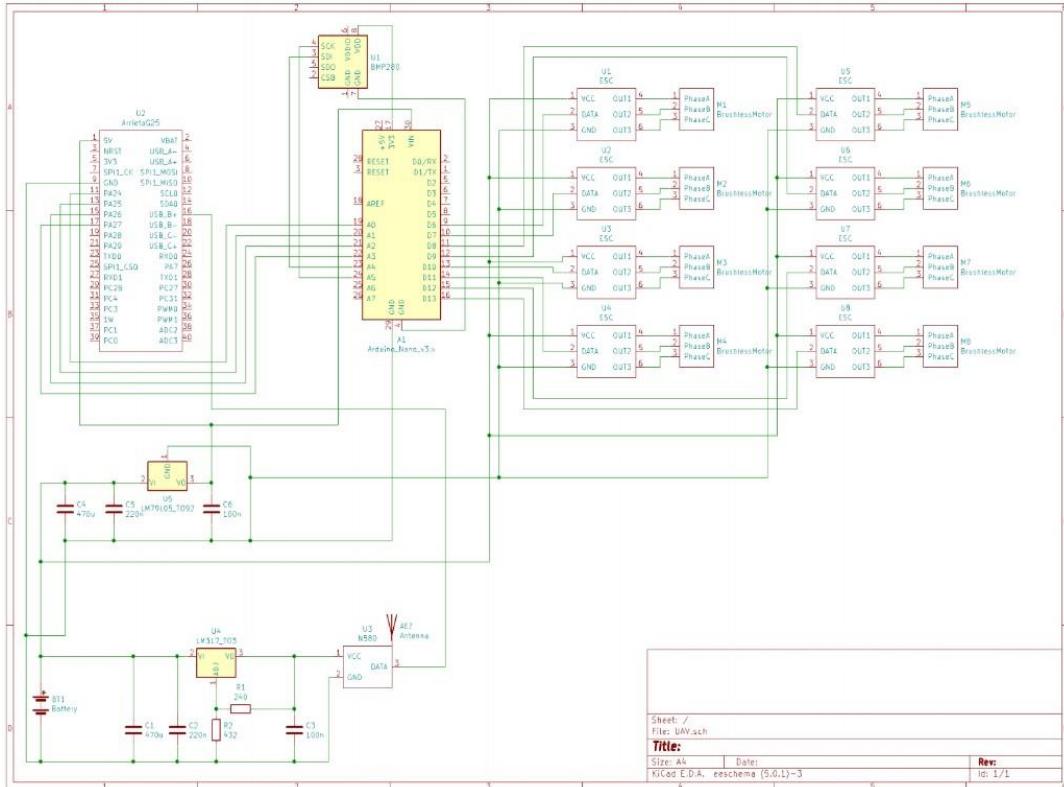


Figure 6A.10: Initial schematic for Arduino Nano and Arrieta G25 implementation.

After being informed that Honeywell would not be able to provide competitors with the HGuideN580 navigation units and that the implementation of the Pixhawk or Ardupilot flight controller was permitted for the competition, the UAV team decided to discard the implementation of the Arrieta G25 processor and Arduino Nano board in favor of the Pixhawk or Ardupilot. Both flight controllers have integrated inertial measurement units that would have been present in Honeywell's navigator. Additionally, these controllers have been specifically created for the open source development of autopilots. The UAV team had both a Pixhawk and an Ardupilot available for the physical implementation in PUPR's manufacturing room, specifically the Pixhawk 2.1 (Cube) flight controller and the Ardupilot v2.6 flight controller. Thus, each flight controller was individually evaluated.

The Cube flight controller is an autopilot intended for manufacturers of commercial systems. It is based on the Pixhawk-project **FMUv3** and can run on either the PX4 or APM firmware on the NuttX OS. The Cube has a separated IMU and FMU systems to reduce interference between them. Then, to ensure precision, it has a triple redundant IMU system: 3xAccelerometer, 3xGyro, 3xMagnetometer, and 2xBarometer.



Figure 6A.11: Cube Flight Controller

The Cube flight controller has the following features:

- Processor
 - 32-bit STM32F427 Cortex-M4F® core with FPU
 - 168 MHz / 252 MIPS
 - 256 KB RAM
 - 2 MB Flash (fully accessible)
 - 32-bit STM32F103 failsafe co-processor
- Sensors
 - On-board: MPU9250 or ICM 20xxx integrated accelerometer / gyro.
 - On-board: MS5611 Barometer
 - Vibration isolated board: L3GD20 gyro
 - Vibration isolated board: LSM303D accelerometer and magnetometer
 - Vibration isolated board: MPU9250 or ICM 20xxx
 - Vibration isolated board: MS5611 Barometer
 - Everything is connected via SPI
- Interfaces
 - 5x UART (serial ports), one high-power capable, 2x with HW flow control
 - 2x CAN (one with internal 3.3V transceiver, one on expansion connector)
 - Spektrum DSM / DSM2 / DSM-X® Satellite compatible input
 - Futaba S.BUS® compatible input and output

- PPM sum signal input
- RSSI (PWM or voltage) input
- I2C
- SPI
- 3.3 and 6.6V ADC inputs
- Internal microUSB port and external microUSB port extension
- Support by MATLAB/Simulink through Pixhawk pilot support package and embedded coder

Ardupilot Mega (APM) is a professional quality IMU autopilot that is based on the Arduino Mega platform. It is a full autopilot capable for autonomous stabilization, waypoint-based navigation and two-way telemetry with Xbee wireless modules. It consists of a main processor board and an IMU shield. Additionally, it supports 8 remote control channels with 4 serial ports.

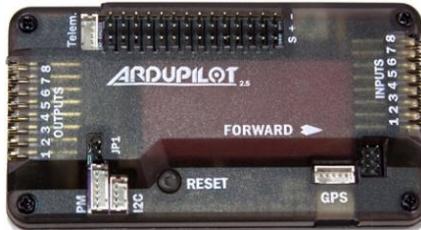


Figure 6A.12: Ardupilot Mega v2.6

The Ardupilot Flight Controller has the following features:

- Includes 3-axis gyro, accelerometer and magnetometer
- Onboard 4 Megabyte DataFlash chip for automatic datalogging
- Optional off-board GPS, uBlox LEA-6H module with compass
- Invensense's 6 DOF Accelerometer/Gyro MPU-6000.
- MS5611-01BA03 barometric pressure sensor
- Atmel's ATMEGA2560 and ATMEGA32U-2 chips for processing and USB functions
- High-resolution altimeter and high-performance barometer
- Arduino Compatible
- Support by MATLAB/Simulink through Ardupilot toolbox

Among the two flight controllers available, the Pixhawk, specifically was initially selected due to its higher processing capability.

Selection of UAV sensors and peripherals

After being informed that the HGuideN580 navigator would not be provided a research was executed to select a set of sensors that would enable the quadcopter to receive a state estimation similar to that which it would have received from Honeywell's navigator that had an integrated IMU and GPS. In terms of the IMU, a selection was not necessary since the selected flight controller had an integrated IMU. Thus, the only sensor to be selected was the GPS. Additionally, a telemetry module had to be selected to calibrate the flight controller's integrated sensors.

The evaluated GPS modules were mainly those recommended in the PX4 user guide such as the: Zubax GNSS 2, Here GNSS GPS (M8N), Here+ RTK GNSS/GPS, and XT-XINTE GPS. The

majority of the considered GPS modules are categorized as Global Navigation Satellite System (GNSS) as well. However, an additional GPS module that is compatible with Ardupilot was evaluated to leave open the possibility of switching from the Pixhawk to the Ardupilot flight controller. To select the GPS module the features for each of the considered modules were identified.

The Zubak GNSS 2 offers the following features:

- Supported by Pixhawk Autopilot Series
- GPS Chipset: U-blox MAX-M8Q
- GNSS Receiver Channels: 72
- Satellite Navigation Systems in View
 - GPS
 - GLONASS
 - Galileo.
- Navigation Sensitivity: -167dBm
- Acquisition Time (seconds): Cold – 26; Aided Start – 2; Reacquisition – 1
- Onboard Components
 - Barometric Pressure Sensor
 - Super Capacitor
 - 3-Axis Magnetometer

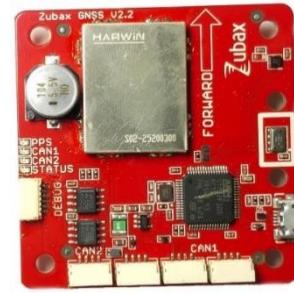


Figure 6A.13: Zubak GNSS 2

The Here GNSS GPS (M8N) offers the following features:

- Supported by Pixhawk Autopilot Series
- Security and integrity protection
- Supports all satellite augmentation systems
- Advanced jamming and spoofing detection
- GPS Chipset: U-blox M8 engine
- GNSS Receiver Channels: 72
- Satellite Navigation Systems in View
 - GPS
 - GLONASS
 - Galileo
 - BeiDou.
- Navigation Sensitivity: -167dBm
- Acquisition Time (seconds): Cold – 26; Aided Start – 2; Reacquisition – 1
- Onboard Components
 - HMC5983 MAG
 - LIS3MDL Mag



Figure 6A.14:
Here GNSS GPS

Then, the Here+ RTK GNSS/GPS shown in Figure 6A.15 was evaluated. This module includes the Real Time Kinematic (RTK) feature which was also present in Honeywell's navigator. This feature greatly increases the accuracy of the GNSS/GPS systems and is useful when users would like to monitor with precision their vehicle current position through a ground control station.



Figure 6A.15: Here+ RTK GNSS/GPS

In general, the Here+ RTK GNSS/GPS offers the following features:

- Centimeter-level GNSS positioning for the mass market
- Integrated Real Time Kinematics (RTK) for fast time-to-market
- Smallest, lightest, and energy-efficient RTK module
- Complete and versatile solution due to base and rover variants
- MPU9250 IMU
- MS5611 Barometer

To enable the RTK feature the following is needed:

- Pair of RTK GPS devices which are the base and air module (one connected to the laptop another to the vehicle)
- Laptop with QGroundControl
- Vehicle with a Wi-Fi or Telemetry radio link to the laptop

Finally, as was mentioned before, the specifications for the XT-XINTE GPS (shown in Figure 6A.16) were identified in the case that the Pixhawk is substituted with the Ardupilot.



Figure 6A.16: XT-XINTE GPS

The XT-XINTE GPS offers the following features:

- U-blox M8N module
- Navigation Sensitivity: 167 dBm navigation sensitivity
- Navigation update rate: Up to 10 Hz
- Cold starts: 26s
- Supported by Pixhawk 2.4 and Ardupilot

From another point of view, for the telemetry module selection there were two options for its implementation. These options were to either use a telemetry Wi-Fi module (ESP 8266 shown in Figure 6A.17) or a telemetry radio module (Readytosky 3DR Radio Telemetry Kit shown in Figure 6A.18). Both offer the same feature of wireless connection, through MAVLink communication protocol, between a ground control station and a vehicle running PX4 or APM firmware. The differences between the two telemetry options are observed in terms of range and data rates. Wi-Fi telemetry modules provide for higher data rates than telemetry radio modules, but telemetry radio modules have a longer connection range than Wi-Fi telemetry module [9].

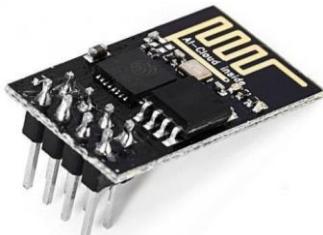


Figure 6A.17: ESP 8266 (Wi-Fi module)



Figure 6A.18: 3DR Radio Telemetry Kit

Since the UAV is not being designed to fly long distances the Wi-Fi telemetry module was selected. However, it is important to recognize that this implementation will require flashing the ESP 8266 Wi-Fi module and then soldering its output to a cable that fits in the flight controller's inputs. If this procedure fails then, the telemetry radio module will be implemented.

Selection of UAV remote control and receiver

The remote controller along with its compatible receiver is required to manually control quadcopter movement (speed, direction, throttle, yaw, pitch, roll). Through the remote controller the different quadcopter's flight modes such as takeoff, land, return to land, stabilize, mission, etc. can be enabled. From another point of view, an important feature of the remote controller is the number of channels it supports. This number defines how many physical controls can be used on the remote controller to send commands to the quadcopter. Since the quadcopter is an aerial vehicle, the remote controller to be selected must support a minimum of 4 channels to have control over roll, pitch, yaw, and thrust. The remaining channels in the remote controller can then be used to control other mechanisms or activate different flight modes provided by the autopilot.

Additionally, since the Cube flight controller is being used, the selected remote-control system (transmitter + receiver) must be compatible with the selected flight controller. According to the Pixhawk User Guide, the Pixhawk flight controller is compatible with the following RC systems:

- All Spektrum DSM RC
- All Futaba S.BUS and S.BUS2 RC
- All FrSky PPM and S.Bus models
- Graupner HoTT
- All PPM models from other manufacturers

Taking compatibility into consideration the following remote controllers which came with their respective compatible receivers where considered: FrySky ACCST Taranis X9D plus RC transmitter w/X8R receiver, Turnigy 9X RC transmitter w/iA8 receiver, and Futaba T6J RC transmitter w/R2006GS receiver.

The FrySky ACCST Taranis X9D plus remote controller shown in Figure 6A.19 is one of the most popular RC units according to the Pixhawk user guide. It has a 2.4Ghz band and 16 comm. channels. The remote controller includes 2 programmable level and 8 programmable switches. The frame rate for this remote controller is 9ms. Additionally, this remote controller contains failsafe mechanisms and has 60 model memory.

From another point of view, the Turnigy 9X remote controller shown in Figure 6A.20 is the best budget transmitter according to DroneUplift's list for the best radio controllers for 2018. It has a 2.4Ghz band and 9 comm. channels. The specifications for this remote controller did not offer details concerning the amount of programmable levels or switches it contained nor did it offer information concerning the remote controller's frame rate. Additionally, this remote controller does not contain failsafe mechanisms and has 8 model memory.

Finally, the Futaba remote controller shown in Figure 6A.21 was recommended by a mechanical engineer that had experience with UAV research and development. It has a 2.4Ghz band and 6 comm. channels. The remote controller has 1 programmable level and 3 programmable switches. The frame rate for this remote controller is 6.8ms. Additionally, this remote controller contains failsafe mechanisms and has 15 model memory. This remote controller differed from the other evaluated controller mostly due to its helicopter features such as: throttle hold, hover pitch, and gyro mixing.



Figure 6A.19: FrySky ACCST Taranis X9D plus RC transmitter w/X8R receiver



Figure 6A.20: Turnigy 9X RC transmitter w/iA8 receiver



Figure 6A.21: Futaba T6J RC transmitter w/R2006GS receiver

The selected remote-control system was the Futaba T6J RC transmitter along with its compatible receiver (R2006GS). This control system was selected because it presented the highest response speed with its frame rate of 6.8ms. Additionally, its unique helicopter features are believed to facilitate the manual operation of the UAV. From another point of view, it is important to note that the selected remote controller contained the lowest amount of comm. channels. The minimum number of channels that a remote controller should have is 4 to be able to successfully control a UAV's throttle, roll, pitch, and yaw. Additional channels may be used to send additional information to the UAV such as the flight mode in which to operate or whether the motors may be armed or not. For this project, only one additional channel is needed which will be used to identify the flight mode in which the drone will operate.

Selection of UAV motors, electronic speed controllers, and propellers

After selecting all the electrical components for the UAV design, the ME team went through the process of selecting the motors, electronic speed controllers, and propellers which are selected in combination. The type of electronic speed controller and propeller to be implemented depends on the selected motor. Then, the specific motor to be implemented depends on the total weight of the quadrotor. For this reason, the ME team executed a mass analysis procedure to estimate the system's total weight. This mass analysis was mainly executed through a MATLAB program called UAV sizing program. This program provided as output the estimated drone weight, frame weight, motor weight, power consumption, and battery weight for 4 types of frames: tricopter, quadcopter, hexacopter, and octocopter as shown in Figure 6A.22. Additionally, an estimation of the cost to manufacture each type of frame with different types of materials was outputted.

UAV Sizing Program:				
	Y_Conf	Quadcopter	Hexacopter	Octacopter
Drone Weight (lbf)	6.4732	6.4136	6.3964	6.4265
Frame weight (lbf)	1.4888	1.4751	1.4712	1.4781
Number of motors	3	4	6	8
Motors weight (lbf)	0.72318	0.69185	0.68284	0.69867
Power in Watts	1277.2	1265.4	1262	1268
Battery Weight (lbf)	1.129	1.1188	1.1158	1.121

Cost in dollars				
	Y_Conf	Quadcopter	Hexacopter	Octacopter
Carbon fiber	287.43	287.43	287.43	287.43
Fiber glass	231.61	231.61	231.61	231.61
PLA	20.99	20.99	20.99	20.99
ABS	21.99	21.99	21.99	21.99

Figure 6A.22: UAV sizing program output

The program was able to provide these outputs by taking into consideration factors such as: frame type, weight of electronic components, disc loading, flight time, air density, and figure of merit for the drone. From another point of view, the outputs shown in Figure 1 were obtained before being informed that HGuideN580 navigation units would not be made available. Thus, the adjustments made in accordance to that new piece of information reduced the quadcopter's estimated weight to 4.85 lb. which is equal to 2199 g. Then, to determine the motor to be implemented one must ensure that the motor can hover the quadcopter at 50% throttle. Thus, the estimated quadcopter's weight is doubled resulting in a weight of 4398 g and by adding a safety factor of 20% the total weight is 5277 g. Therefore, the thrust system must be capable of producing at least 5600 g of thrust. Thus, a thrust of 1400 grams must be produced by each motor. As a result, the Cobra CM-2213/26 950 KV motor shown was selected. Note that this specific motor is called the CM-2213/26. The 22 refers to the motor diameter and the 13 refers to the height in mm. The higher the motor shaft, the more torque to motor will produce and the more weight your system can carry [10]. Additionally, the KV value is usually referred to as the number of rpms that the motor can produce per volt. However, this isn't technically correct since the KV value is not an indicator for how powerful the motor, how much current it can handle, or how efficient it is. To determine this information, thrust tests need to be executed. The KV value is at best a piece of information used to understand a motor's current requirements to produce a certain amount of torque. A small KV value requires less current to increase propeller speed but, shows a loss in efficiency at high rpm. Inversely, a large KV value requires more current to increase propeller speed but, can achieve high rpms efficiently [10].

Once the motor has been selected, can proceed to select the ESC and the propeller. The ESC is easily determined by looking for the motor's maximum continuous current in the datasheet offered by the manufacturer. The maximum continuous current for the selected motor is 14 A. Thus, an ESC capable of withstanding maximum continuous current of 14 A or more must be selected. AS a result, the Cobra 20A Opto Multirotor ESC was selected. Then, the propeller is selected by looking at the propeller data chart for the selected motor offered by the manufacturer. In this data chart the manufacturer recommends different types of propeller dimensions from different manufacturers. The considered propeller dimensions for this project were from the manufacturer named Advanced Precision Composites (APC) Since this is the brand that is most commonly used for small UAV development. For this reason, only the recommended propeller dimensions from APC are shown in Figure 6A.23.

Cobra CM-2213/26 Motor Propeller Data										
Magnets 14-Pole	Motor Wind 26-Turn Delta		Motor Kv 950 RPM/Volt		No-Load Current $I_o = 0.42$ Amps @ 12v	Motor Resistance $R_m = 0.230$ Ohms		I Max 14 Amps	P Max (3S) 155 W	
Stator 12-Slot	Outside Diameter 27.0 mm, 1.063 in.		Body Length 28.0 mm, 1.102 in.		Total Shaft Length 30.1 mm, 1.185 in.	Shaft Diameter 3.17 mm, 0.125 in.	Motor Weight 65 gm, 2.29 oz			
Test Data From Sample Motor		Input I_o Value	10.0 V 0.39 A	12.0 V 0.42 A	14.0V 0.46 A	16.0V 0.50 A	Measured Kv value 967 RPM/Volt @ 10v	Measured Rm Value 0.230 Ohms		
Prop Manf.	Prop Size	Li-Po Cells	Input Voltage	Motor Amps	Input Watts	Prop RPM	Pitch Speed in MPH	Thrust Grams	Thrust Ounces	Thrust Eff. Grams/W
APC	8x4.5-MR	3	11.1	7.40	82.1	8,579	36.6	541	19.08	6.59
APC	9x4.5-MR	3	11.1	9.63	106.9	7,964	33.9	692	24.41	6.47
APC	10x4.5-MR	3	11.1	11.86	131.6	7,395	31.5	826	29.14	6.27
APC	11x4.5-MR	3	11.1	13.79	153.1	6,861	29.2	943	33.26	6.16

Figure 6A.23: Extracted information from Cobra CM-2213/26 Motor Propeller Data [11]

After evaluating the data in Figure 6A.22, the APC 11x4.5-MR propeller was selected. Note that this specific propeller is called the 11x4.5-MR. The 11 refers to the propeller diameter in inches and the 4.5 refers to the propeller pitch in inches. A 4.5 propeller pitch will cause the quadrotor to move forward 4.5 inches after one full rotation. While a higher propeller pitch produces faster top speeds and draws more current, a lower propeller pitch provides for more torque and maneuverability while drawing less power [10].

Selection of UAV materials

This selection phase was also executed by the ME team. In terms of UAV materials, these were chosen according to the material's: availability, cost efficiency, manufacturability, thermal properties, resistance to shear stress, bending moment, signal interference, and water resistance. The initially evaluated materials where carbon fiber, weighting, fiberglass, ABS (filament for 3D printing), and PLA (filament for 3D printing). Among these 5 types of material PLA was selected due to its low cost and practicality given that the UAV team had manufacturing room equipped with 3D printers. Other materials that were later included to the design were Nylon X and Nylon G filaments and Lexan.

Selection of UAV power supply and power management board

Most UAVs use batteries as their power supply since fuel engine-based UAVs are expensive, heavily regulated and application specific. In the beginnings of RC aircraft and drone design, Nickel Cadmium (NiCad) and Nickle Metal Hydride (NiMH) were used. However, now the most commonly used batteries on UAVs are lithium based due to the different advantages that they have over NiCad and NiMH batteries such as their higher capacity, higher discharge rate, and lower weight. There are different types of lithium-based batteries. The types that are frequently considered for UAV design are mainly two:

1. Lithium Ion (Li-ion)

- Twice the high energy density of lithium polymer batteries
- Can provide from 40 to 60 minutes of flight time in copters
- Generally packed in a cylindrical form with metal casing
- Capable of operating to over a thousand cycles
- Low discharge rate



Figure 6A.24: Lithium ion battery

2. Lithium Polymer (LiPo)

- Available in various configurations: C-Rating, milli ampere-hour, and number of cells
- Generally, comes in a square package that can easily fit in any UAV
- Can provide from 10 to 30 minutes of flight time in copters

- Lower energy density than lithium ion
- Capable of operating up to 300 cycles
- High energy discharge rate



Figure 6A.25: Lithium polymer battery

Due to its variety in configuration, convenient packaging, and high discharge rate, the lithium polymer battery was found to be the best type of battery to be implemented on the UAV to be designed. However, this selection does not finalize the consideration of alternatives for the UAV's power supply. As it has been mentioned before, LiPo batteries are offered in a wide variety of configurations according to its c-rating, milli amperes per hour, and number of cells.

The C-rating of a battery is directly related to a battery's discharge rate. The higher the c-rating the lower the internal resistance in a battery, and the higher the discharge rate. This also makes the battery more expensive. From another point of view, it is important to note that sometimes batteries will be sold showing a range of C-rating. An example of a C-rating range that is sometimes offered in batteries is 20C-30C. This range means that the battery will provide a constant C-rating of 20C and can provide burst of 30C. Thus, they should not be used on drones that need a constant c-rating greater than 20C. The LiPo battery used in this project was selected to have a C-rating of 35C.

There is also the consideration of the battery's milli ampere-hour (mAh) rating. The mAh unit is an electric charge unit specifying the charge transferred by a steady current of one mili ampere flowing for one hour. Thus, this rating is directly related to the flight time that the battery will be capable of providing. The longer the flight time, the larger this rating must be. The LiPo battery used in this project was selected to have a 4200 mAh rating.

On the other hand, there is the consideration of the number of cells which is directly related to the amount of voltage provided by the battery. The greater the number of cells, the greater the amount of voltage. The following table shows the direct relation between the number of cells in the battery and the battery's nominal voltage.

Table 6A.2: Relation between number of cells and nominal battery voltage

Number of Cells	Nominal Battery Voltage
1S	3.7 V
2S	7.4 V
3S	11.1 V
4S	14.8 V

5S	18.5 V
6S	22.2 V

The minimum number of cells (3S) that the battery powering the selected motors should have is specified in the selected motor's propeller data chart shown in Figure 6A.23. As a result, the number of cells for the battery was selected to be 4S containing one more cell to ensure power compliance.

After selecting the specifications for the LiPo battery, the flight time for the case in which each motor is drawing maximum current (14 A) may be measured by doing the following division:

$$\frac{\text{Ampere hour}}{\text{total current draw}} = \frac{4.2 \text{ Ah}}{56 \text{ A}} = 0.075 \text{ h} = 4.5 \text{ min.}$$

For the testing phase however, the typical current demand of the selected motor will be determined to measure with better precision the flight time that the selected LiPo battery will provide.

From another point of view, the power management board is another component to take into consideration to facilitate the distribution of power among the different electronic components implemented for the UAV design. For this project, the selected power distribution board, also known as power management module, was the HolyBro PM07 specifically designed for the Pixhawk Flight Controller. This module was selected since a flight controller from the Pixhawk brand had already been selected.

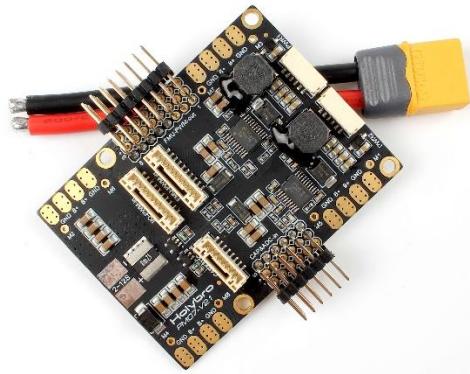


Figure 6A.26: HolyBro PM07 Power Management (PM) Module

Quadrrotor Software

Since the use of the Arduino Nano was discarded in favor of the Pixhawk and Ardupilot flight controller, another evaluation had to be executed. These flight controllers must run on a firmware to be functional. Additionally, ground control stations must be used to calibrate the flight controller's sensors. Thus, as part of the design one must proceed to select the firmware and the ground control station to be implemented.

Selection of flight controller firmware

While the Ardupilot flight controller can run only on the APM firmware, also known as flight stack, Pixhawk's Cube flight controller can run on either the APM or the PX4 firmware. Thus, there are two implementation options concerning the firmware for the selected flight controller.

The PX4 firmware is composed of three layers the flight stack (contains individual programs for applications such as flight control and state estimation), the middleware (facilitates communication between programs installed in the flight controller and the drivers), and the drivers as shown in Figure 6A.27. This architecture allows for a modular design since each of the three layers can run independently. This also allows for the individual development of programs that may then be integrated to the flight controller by installing them on the PX4 flight stack. Additionally, each application connects to other processes and drivers using a Publisher-Subscriber framework. This allows for efficient communication between processes and simplifies the process of adding new applications to the flight controller.

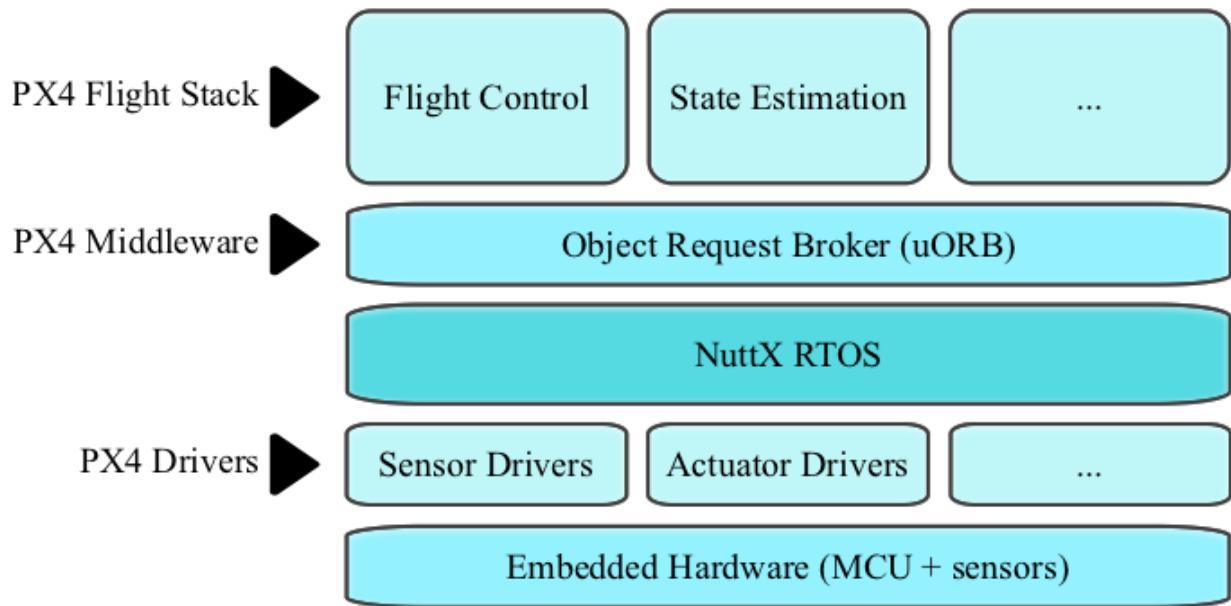


Figure 6A.27: PX4 firmware architecture on Pixhawk flight controller

On the other hand, there is the APM firmware to consider which was originally designed for the Ardupilot flight controller but has since been ported as a single application to the PX4 flight control architecture so that it may be run on any Pixhawk flight controller through the PX4 middleware layer as seen in Figure 6A.28. By selecting to implement the APM firmware, the APM application will substitute the PX4 flight stack for the APM flight stack to control the flight controller's drivers. From the user's perspective, this allows the PX4 flight controller to behave like the legacy APM hardware.

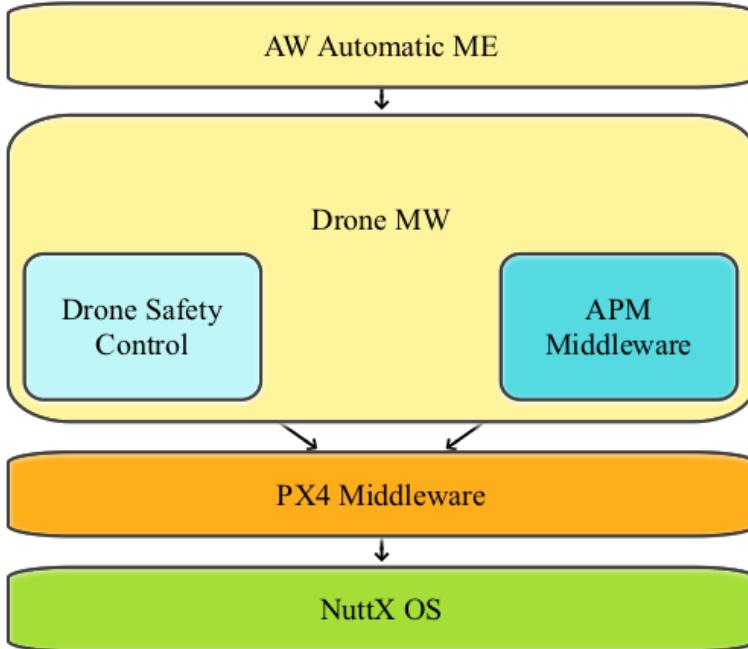


Figure 6A.28: APM firmware architecture on Pixhawk flight controller

After evaluating each firmware, the PX4 firmware was ultimately selected since it was specifically designed for the selected flight controller and the greater flexibility that offered by its flight stack which was not composed of a single application contrary to the Ardupilot firmware. Most importantly, the PX4 firmware had greater compatibility with MATLAB/Simulink since MathWorks had already initiated an open source community effort to promote the implementation of the Pixhawk autopilot through its PX4 Pilot Support Package (PX4 PSP). The PX4 PSP provides the means to implement controllers and models designed in Simulink onto the PX4 flight control hardware. It contains a library of PX4 Simulink blocks along with PX4 example Simulink models that can be used for developing the plant or controller of a vehicle. However, it is important to notice that although the PX4 PSP functions as a good guide to create Simulink models that may be uploaded to the PX4 flight control hardware optimize its control logic, the PX4 PSP is outdated and cannot be used to upload control logic to the Cube flight controller specifically. Then, to upload the control logic to the Cube flight controller, MathWorks's Embedded Coder Support Package for PX4 Autopilots is used which contains a library of PX4 Simulink blocks similar to that of the PX4 PSP.

Selection of UAV Ground Control Station (GCS)

A ground control station is an application that runs on a ground-based computer and allows users to communicate with their UAV via telemetry. It can display real-time data concerning the UAV performance and position. Commands such as executing a mission may be pre-programmed or sent to the flight controller via the GCS. Additionally, GCSs may be used to calibrate the selected remote controller and the UAV's integrated sensors. There are different ground control stations to choose from. However, the two most commonly used are the Mission Planner, and QGroundControl. Both have very similar features such as:

- Loading appropriate firmware into the flight controller hardware.
- Setting up, configuring, and tuning the flight controller hardware and additionally implemented devices such as the remote controller and sensors.
- Planning, saving and loading autonomous missions into the flight controller hardware.
- Downloading and analyzing mission logs created by the flight controller hardware.
- Monitoring UAV flight status (requires telemetry).
- Recording, viewing, and analyzing telemetry logs.

While Mission Planner only supports flight controllers running the APM firmware, QGroundControl supports flight controllers running either the APM or the PX4 firmware. For this reason, QGroundControl was the selected GCS.

B. Mathematical Justification

Deriving the mathematical model of the quadcopter system was the first step to designing the quadcopter's control system. This step required proper knowledge of the physical laws and mathematical principles that describe the quadcopter's behavior. Thus, the necessary information to derive the mathematical model is described below. On the other hand, it is important to note that the following general assumptions were made concerning the quadcopter system while executing the derivation:

- The quadcopter structure is rigid and symmetrical with a center of mass aligned with the center of the body frame of the vehicle.
- The thrust and drag of each motor is proportional to the square of the motor velocity.
- The propellers are rigid and therefore blade flapping is negligible (deformation of propeller blades due to high velocities and flexible material).
- The Earth is flat and non-rotating (difference of gravity by altitude or the spin of the earth is negligible).
- Ground effect is negligible.

1. Quadcopter Configuration

A quadcopter, also referred to as quadcopter helicopter, is an aerial vehicle with four BLDC motors. Each motor is mounted in an extended arm and attached to a fixed-pitch propeller. The propellers are arranged to rotate on pairs; one pair rotates clockwise, and another pair rotates counter-clockwise. The rotors are directed upwards and placed in a square formation with equal distance from the center of mass of quadcopter.

Quadcopters are operated by applying the concepts of angular velocities, variable torques, thrusts and forces as discussed by Sattar and Ismail [12]. Concerning forces, an important concept to understand is that all forces come in pairs. For this reason, as the rotor pushes down on the air, the air pushes up on the rotor. This is the basic idea behind lift, which comes down to controlling the upward and downward force. The faster the rotors spin, the greater the lift, and vice-versa. There are two basic flight configurations that can be adopted by a quadcopter with the rotation direction of the four propellers: the plus configuration and cross configuration as shown in Figure 6B.1.

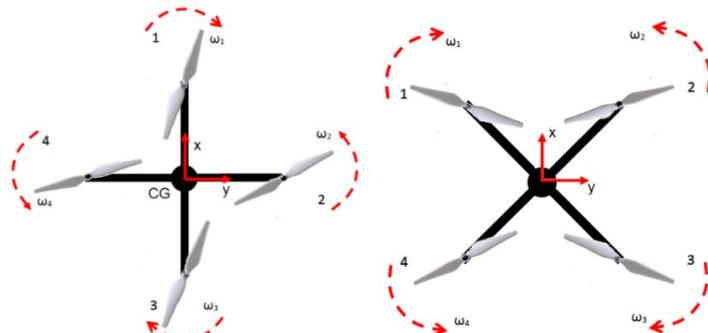


Figure 6B.1: Quadcopter in plus (+) and cross (X) configurations

In the plus configuration the quadcopter attitude (i.e., roll, pitch or yaw) is changed by varying the rotational speed for two propellers, but in cross configuration this is accomplished by varying the rotational speed of all four propellers. The latter configuration provides for higher momentum and better maneuverability performance as presented by Wei Zhong [13]. In both configurations, the distance of the quadcopter's center of mass from the axis of rotation is used to determine the necessary torque to be generated by the motors and control the quadcopter. Considering a quadcopter with each arm of length r away from the center of mass of the vehicle and using one axis of motion (roll or pitch) the following can be stated about the plus and cross configurations:

- For the plus configuration: The thrust forces are applied at a distance r .
- For the cross configuration: The thrust forces are applied at distance of $r \cdot \cos(\pi/4)$ approximately $0.71 \cdot r$ since the arms are at a 45-degree angle from the axis of rotation.

Thus, the main advantage of the cross configuration is that torque is generated with all four motors, and this provides for a torque that is greater by the square of the torque produced by the plus configuration [14]. On the other hand, quadcopters are able to perform flight maneuvers flight by having flight controller hardware that sends PWM signals to the brushless DC motors and vary the rotational speed of the propellers. Quadcopters are also successfully maneuvered and balanced by ensuring that the propellers rotate towards the quadcopter's main body or barycenter [15]. This is achieved through the following motor setup:

- Front left – clockwise motor (CW)
- Front right – counter-clockwise motor (CCW)
- Back left – counter-clockwise motor (CCW)
- Back right – clockwise motor (CW)

2. Coordinate System

There are two reference systems that must be defined when determining the quadcopter's position. One of the reference systems is fixed and the other is mobile as discussed by Sabatino [15]. By assuming a flat fixed Earth model, quadcopter movement can occur around and above a flat ground in a rectilinear fixed coordinate system. Newton's first law is applied in the previously mentioned fixed system which is also known as inertial coordinate system (Inertial Frame). In this system the frame is centered on the origin O and set on the ground pointing towards North, East, and Down (NED) as shown in Figure 6B.2.

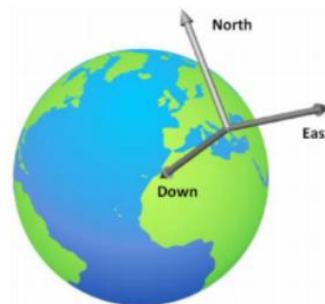


Figure 6B.2: Origin, north, east and down (ONED) inertial reference system.

A rigid body, unlike a particle, occupies a volume of space and has a particular shape. This shape is defined in terms of a fixed and unchanging space called body space. As a result, a rigid body can only undergo rotation and translation. By assuming that the quadcopter is a rigid body, the mobile reference system will refer to the fixed quadcopter frame which is called the Body-fixed coordinate system or Body frame. The origin of the body frame is assumed to be at the center of gravity (CG). Generally, the origin of the body frame is referred to as *OABC* system, where ABC stands for *Aircraft Body Center*. Figure 6B.3 illustrates the relation between the two coordinate systems: ONED and OABC.

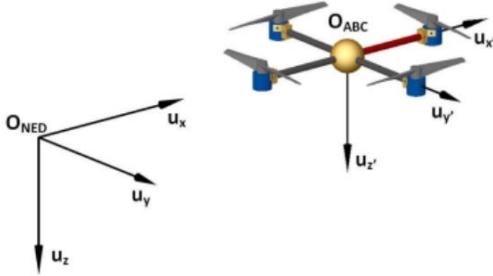


Figure 6B.3: Fixed and mobile reference systems

3. 6DOF

The location of all points on a rigid bodied vehicle in a 3-D world can be described with 6 coordinates known as 6 degrees of freedom or 6DOF [16]. The first 3 are the directional (translational) coordinates which represent the distance of the object's center of mass from some origin in the 3-D world. The last 3 coordinates are the rotational directions represent through the following angles: roll, pitch and yaw.

3.1 Vertical Motion

The propellers rotate at the same speed to generate a thrust force that will overcome the quadcopter's weight and lift the quadcopter from the ground (hover) or produce a vertical maneuver (ascend or descend). The three types of vertical motions that a quadcopter may produce are shown below:

- Hovering: The net thrust of the four rotors pushing the drone up must be equal to the gravitational force pulling it down.
- Ascending: The thrust of the four rotors is increased so that there is a non-zero upward force greater than the weight.
- Descending: The thrust of the four rotors is decreased so that the net force is downward.

3.2 Horizontal Motion (Forward, Backwards, and Sideways)

To perform a flight maneuver in the horizontal plane or a side-to-side motion, the quadcopter would need to generate a rotating moment by pitching or yawing its body. This is accomplished by varying the rotational speed of each set of propellers.

Forward flight is produced by increasing the rpm (revolutions per minute) of rotors 2 and 3 (rear motors shown in Figure 6B.5) and decreasing the rpm of rotors 1 and 4 (front motors shown in Figure 6B.5). The total thrust force will remain equal to the weight, so the drone will stay at the same vertical level. Also, since one of the rear rotors is spinning counterclockwise and the other clockwise, the increased rotation of those motors will still produce zero angular momentum. The same applies to the front motors therefore resulting in the drone not rotating.

3.3 Rotation Movement: Roll, Pitch and Yaw

There are two possible conventions when discussing a rotation: rotation of the axes, and rotation of the object (body) relative to fixed axes. Euler's rotation theorem by Leonhard Euler, states that in three-dimensional space, any displacement of a rigid body, such that a point on the rigid body remains fixed, is equivalent to a single rotation about some axis that runs through the fixed point. This implies that the composition of two rotations is also a rotation. In addition, any orientation can be achieved by composing three elemental rotations, i.e. rotations about the axes of a coordinate system.

As a result, any movement can be achieved by combining three elemental rotations and describing the orientation of a rigid body. In general, the rotation around the forward direction axis is called roll rotation while the rotation around the lateral direction axis is called pitch rotation and the rotation around the upward direction is called yaw rotation. In other words, the rotation angles are simply known as yaw, pitch and roll respectively or Euler angles.

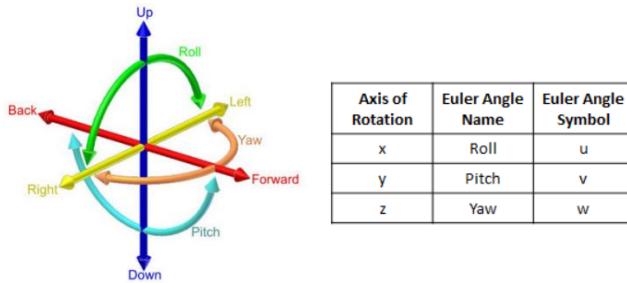


Figure 6B.4: Euler angle axes [17], names and symbol convention [18]

3.3.1 Translational or rotational maneuver

- The quadcopter in cross configuration changes the rotational speed of all four propellers by the same amount, to generate a thrust and accelerate the quadcopter along the vertical z-axis.

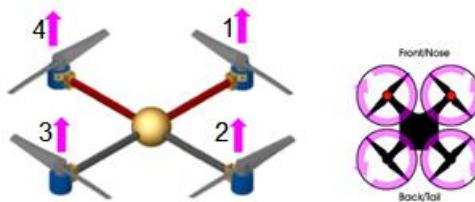


Figure 6B.5: Thrust [19]

3.3.2 Roll maneuver

- The rotation around the front-to-back axis is called roll. Roll moves the quadcopter on the longitudinal axis, so it would tilt side to side. This causes the quadcopter to fly sideward, either to left or right depending on the tilt – banking left or right. For example, the rotational speed of propellers 1 and 2 are increased, while the rotational speed of propellers 3 and 4 are reduced to generate a torque along the x-axis and roll to the left and vice versa.

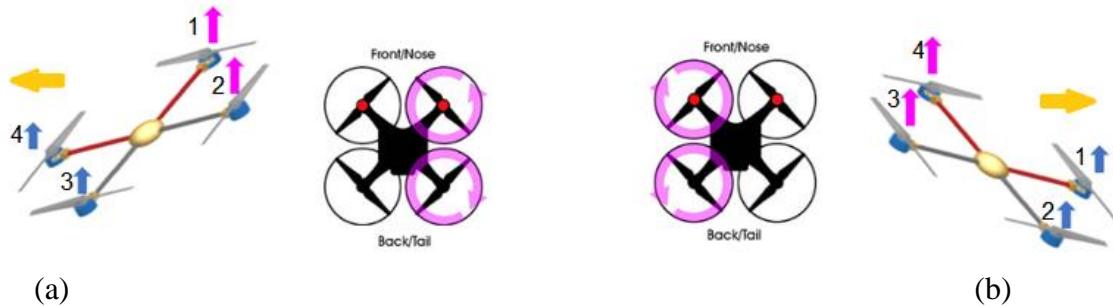


Figure 6B.6: Roll maneuver (a) to the left (b) to the right [19]

3.3.3 Pitch maneuver

- The rotation around the side-to-side axis is called pitch which makes the quadcopter tilt up and down from front to back. This causes the vehicle to move forward or backwards depending on which way it is tilted. For example, the rotational speed of front propellers (1 and 4) are increased, while the rotational speed of back propellers (3 and 2) are reduced to generate a torque along the y-axis.

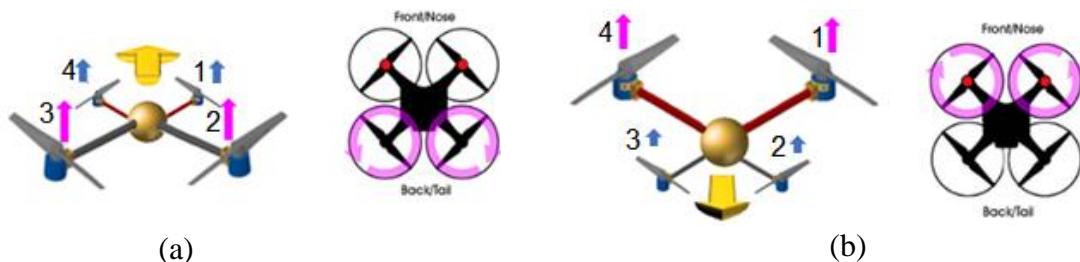


Figure 6B.7: Pitch maneuver [19]:

- (a) rotational speed of back/rear propellers is increased, and quadcopter moves forward
- (b) rotational speed of front propellers is increased, and quadcopter moves backwards

3.3.4 Yaw maneuver

- The rotation around the vertical axis is called yaw. It moves the quadcopter around in a clockwise or counterclockwise rotation as it stays leveled to the ground. For example, applying different rotational speed to the counter rotating pair of propellers, a torque along the z-axis is generated.

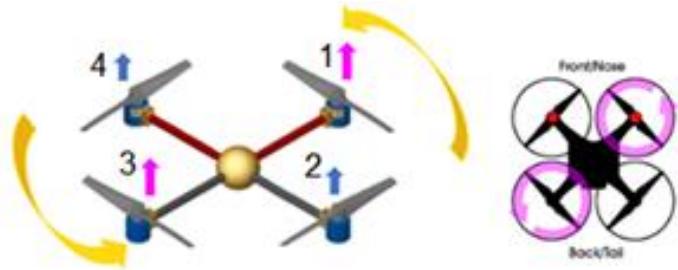


Figure 6B.8: Yaw maneuver: Turning around z-axis [19]

In summary, the quadcopter's translational and rotational movements can be expressed through six variables as follows:

- x and y represent the position of the quadcopter in space
- z defines the altitude of the quadcopter
- ϕ or roll angle represents angle about the x-axis
- θ or pitch angle represents angle about the y-axis
- ψ or yaw angle represents angle about the z-axis

4. Kinematics

The kinematics of the quadcopter is well described in [20] - [22]. The rigid body kinematics uses the relationships governing the displacement, velocity, acceleration, and rotational motion of the body. To describe the motion of a 6DOF rigid body, two reference frames were defined: The inertial frame and the body frame. The distance between the inertial frame and the body frame (r) describes the absolute position of the center of mass of the quadcopter:

$$r = [x \ y \ z]^T \quad (4.1)$$

The orientation of the quadcopter is described using roll, pitch and yaw angles ($\phi \ \theta \ \psi$) representing rotations about the X, Y and Z-axes respectively. Since the position of the quadcopter is given in the inertial frame and the velocity and angular velocity are defined in the quadcopter body frame, a transformation from one frame to the another is needed. A rotation matrix \mathbf{R} will be used to execute the previously mentioned transformation. The rotation matrix will also allow to have a relation between the states measured in the body frame (like thrust forces produced by the propellers) and the states measured in the inertial frame (e.g. the gravitational forces and the quadcopter's position). From another point of view, the rotation \mathbf{R} from the body frame to the inertial frame describes the orientation of the quadcopter. Assuming the order of rotation to be roll (ϕ), pitch (θ) then yaw (ψ), the rotation matrix R is derived based on the sequence of principle rotations as follows:

- Rotation about the x-axis by angle ϕ :

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix}$$

- Rotation about the y-axis by angle θ :

$$R_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$$

- Rotation about the z-axis by angle ψ :

$$R_z(\psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The transformation of the overall rotation movement (R_T) of the body frame is then:

$$R_T = R_x(\phi)R_y(\theta)R_z(\psi)$$

or

$$R_T = \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (4.2)$$

The information about the angular velocity of the quadcopter is typically obtained from an on-board Inertial Measurement Unit (IMU). This component will also give the velocity in the body coordinate frame [23]:

- u = the body frame velocity measured along x-axis
- v = the body frame velocity measured along y-axis
- w = the body frame velocity measured along z-axis

Special attention should be given to the difference between the body rate measured p, q, r in Body Fixed Frame and the Euler angle rates $\dot{\phi}, \dot{\theta}, \dot{\psi}$ expressed in Earth Fixed Frame. The transformation matrix from $[\dot{\phi} \dot{\theta} \dot{\psi}]^T$ to $[p \ q \ r]^T$ is given by:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin \phi \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \sin \theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (4.3)$$

To relate the Euler rates $\dot{\eta} = [\dot{\phi} \dot{\theta} \dot{\psi}]^T$ that are measured in the inertial frame and angular body rates $\omega = [p \ q \ r]^T$, a transformation is needed as follows:

$$\omega = R_r \dot{\eta} \quad (4.4)$$

$$R_r = \begin{bmatrix} 1 & 0 & -\sin \phi \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \sin \theta \end{bmatrix} \quad (4.5)$$

Around the hover position, small angle assumption is made where $\cos \phi = 1$, $\cos \theta = 1$ and $\sin \phi = \sin \theta = 0$. The matrix R_r can be simplified to an identity matrix I . The derivation for the previous transformation is well elaborated by Nabil ElKholy [22].

The quadcopter state variables are well described in articles [23] - [24]. Therefore, the generalized quadcopter state variables are:

X = the inertial (north) position of the quadcopter along x-axis in inertial frame,

Y = the inertial (east) position of the quadcopter along y-axis in inertial frame

h = the altitude of the aircraft measured along z-axis in inertial frame

u = the body frame velocity measured along x-axis

v = the body frame velocity measured along y-axis

w = the body frame velocity measured along z-axis

ϕ = the roll angle defined with respect to body frame

θ = the pitch angle defined with respect to body frame

ψ = the yaw angle defined with respect to body frame

p = the roll angular rate measured along x-axis in body frame

q = the pitch angular rate measured along y-axis in body frame

r = the yaw angular rate measured along z-axis in body frame

5. Quaternions

In addition to Euler angles, 3x3 matrices, axis/angles used to express rotations in 3D space, there is Quaternions. As mentioned in [25] - [28], a quaternion is a 4-dimensional complex number and is commonly used to represent a rotation in 3-dimensional space. A quaternion is somewhat like an axis/angle where the x, y, and z components are the axis, and the w component is the angle. Quaternions have the following advantages over other methods:

- Can be converted to and from an axis/angle without any loss of information.
- Can be converted to a rotation matrix.
- Do not present the problem of gimbal lock like Euler angles since quaternions can perform rotations on all axes at once.
- More efficient for vector multiplication than Matrices which require more storage space in memory.
- Axis/angles work similarly to quaternions but are less convenient for interpolation and vector multiplication than quaternions are.

6. Dynamics: Forces and moments acting on the Quadcopter

Dynamics is a branch of mechanics which studies the effects of forces and torques on the motion of a body or system of bodies. There are several techniques which can be used to derive the equations of a rigid body with 6DOF. A Newton-Euler formulation is used to derive the dynamics of the quadcopter in terms of force and momentum. Newton's equations describe the translational motion of the center of mass (altitude and x and y position), while Euler's equations describe the rotational motion about the center of mass (roll, pitch and yaw). Therefore, the Newton-Euler unique equation that defines the total influence of the net forces and moments on the quadcopter is:

$$\begin{bmatrix} mI_{3x3} & 0 \\ 0 & J \end{bmatrix} \begin{pmatrix} \dot{v}^b \\ \dot{\omega}^b \end{pmatrix} + \begin{bmatrix} \omega^b \times (mv^b) \\ \omega^b \times (J\omega^b) \end{bmatrix} = \begin{pmatrix} F^b \\ \tau^b \end{pmatrix} \quad (6.1)$$

where:

- m is the mass of the quadcopter in kg
- I_{3x3} denotes 3x3 identity matrix for inertia
- 0_{3x3} denotes 3x3 zero matrix
- J is the moment of inertia matrix for the quadcopter
- v^b change of body linear velocity $[u, v, w]^T$
- ω^b change of body angular rates $[p, q, r]^T$

6.1 Gravitational Force

The gravitational force also exerts force on the quadcopter by acting on the center of gravity in the inertial frame. This is given by:

$$F_g^I = (0 \ 0 \ mg)^T \quad (6.2)$$

Then, the gravity force transferred to the body frame is:

$$F_g^b = R^T F_g^I = \begin{bmatrix} -mg \sin \theta \\ mg \cos \theta \sin \phi \\ mg \cos \theta \cos \phi \end{bmatrix} = mg \begin{bmatrix} -\sin \theta \\ \cos \theta \sin \phi \\ \cos \theta \cos \phi \end{bmatrix} \quad (6.3)$$

6.2 Inertia

The inertia matrix for the quadcopter is a diagonal matrix. The off-diagonal elements, $J_{xy} = J_{xz} = J_{yz} = 0$, which are the product of inertia, are zero due to the symmetry of the airframe's mass distribution of the quadcopter with respect to the Body frame Timmons [29].

$$J = \begin{bmatrix} J_{xx} & -J_{yx} & -J_{zx} \\ -J_{xy} & J_{yy} & -J_{zy} \\ -J_{xz} & -J_{yz} & J_{zz} \end{bmatrix} \Rightarrow J = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \quad (6.4)$$

J_{xx}, J_{yy} and J_{zz} are the area moments of inertia about the principle axes in the body frame and are calculated as follows:

$$\begin{aligned} J_{xx} = J_{yy} &= \frac{\frac{2m_e r^2}{5}}{5} + 2d^2 m_c \\ J_{xx} &= \frac{\frac{2m_e r^2}{5}}{5} + 4d^2 m_c \end{aligned} \quad (6.5)$$

where:

m_c = Mass of the quadcopter's center (assuming a spherical dense center, with radius r)

m_e = The mass at the end of each cross arm, where the propellers are located

6.3 Gyroscopic Moment

A gyroscope works on the principle that angular momentum changes in the direction of torque. The gyroscopic moment of a rotor is a physical effect that acts on the quadcopter in the body coordinate frame in which gyroscopic torques or moments attempt to align the spin axis of the rotor along the inertial z-axis, [22], [13], and [29]. The Gyroscopic moments (M_G) caused by the rotor's moment of inertia, the rotor's angular velocity, and the angular rate of change of the plane of rotation of the propeller (body attitude rate) is:

$$M_G = \omega \times [0 \ 0 \ J_r \Omega_r]^T \quad (6.6)$$

where:

J_r is moment of inertia that put the propeller

Ω_r is the angular velocity of the propeller

$[0 \ 0 \ J_r \Omega_r]$ is angular momentum vector

ω is angular rate of change of the plane of rotation of the propeller

Following the path of rotation of the propeller, through the right-hand rule the thumb indicates angular velocity vector (same direction as the angular momentum vector $J_r \Omega_r$). Then, the propeller is rotating in the XY plane and by the right-hand rule, cross X to Y, the thumb is pointing down to Z-axis.

As the propeller is spinning at a constant angular velocity Ω_r , if we twist the XY plane of rotation of that rotating propeller about the y-axis at an angular velocity and positive ω along the wide array of the y-axis, it results in a ω vector in the y direction and a torque vector M_G down the direction of x-axis. In other words, if we impart an angular velocity about the y-axis, a torque is generated that wants it to rotate around de x-axis.

That is the gyroscopic effect of the spinning propeller as its plane of rotation is changes with an angular velocity ω . However, since the moment of inertial of the rotor-propeller combination is a constant and small value, therefore negligible, [13] and [29].

6.4 Lift Force and Aerodynamic Moment

Two physical forces and moments on the body of the Quadcopter, as an effect of rotation produced by the rotors are: The aerodynamic forces or lift force and the aerodynamic moments. Figure 6B.9 shows the forces and moments acting on the quadcopter.

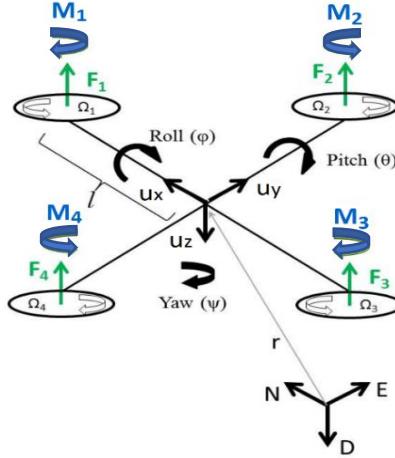


Figure 6B.9: Forces and moments acting on quadcopter

A propeller creates an aerodynamic thrust force out of the supplied power. The performance of propellers can be described by dimensionless (normalized) coefficients. A propeller can be described in terms of *advance ratio*, *thrust coefficient*, and *power coefficient*. Based on the publications of the National Advisory Committee for Aeronautics (NACA), thrust coefficient C_T , and power coefficient C_P are defined as:

$$C_T = \frac{T}{\rho n^2 D^4} \quad (6.7)$$

$$C_P = \frac{P}{\rho n^3 D^5} \quad (6.8)$$

Therefore, the thrust and power forces of the propeller may be written as:

$$T = C_T \rho n^2 D^4 \quad (6.9)$$

$$P = C_P \rho n^3 D^5 \quad (6.10)$$

where:

- v is the velocity in m/s
- D is the propeller diameter in m
- n represents the revolutions per second 1/s
- ρ is the density of air kg/m³
- P is the power W
- T is the thrust N

The magnitude of the thrust force is not constant for a given propeller but depends on the velocity of the incoming air and the rotational velocity of the propeller itself. Since for the case of quadcopters, the maximum altitude is usually limited, the air density can be considered constant. The aerodynamic thrust force F_i is proportional to square of the propeller's angular velocity Ω_i , and acts along the z-axis (u_z) as shown in Figure 9. It is defined as:

$$F_i = K_f \Omega_i^2 \quad (6.11)$$

where:

- K_f is the aerodynamic force constant
- Ω_i is the angular velocity of rotor i

Each rotor causes an upwards thrust force F_i and generates a moment M_i with direction opposite to the direction of rotation of the corresponding rotor i . It is defined as:

$$M_i = K_M \Omega_i^2 \quad (6.12)$$

where:

- K_M is the moment constant

Starting with the moments about the body frame's x-axis, by using the right-hand rule in association with the axes of the body frame, F_2 multiplied by the moment arm l generates a negative moment about the y-axis, while in the same manner, F_4 generates a positive moment. Thus, the total moment about the x-axis can be expressed as:

$$M_x = -F_2 l + F_4 l = -(K_f \Omega_2^2)l + (K_f \Omega_4^2)l = K_f l(-\Omega_2^2 + \Omega_4^2) \quad (6.13)$$

For the moments about the body frame's y-axis, also using the right-hand-rule, the thrust of rotor 1 generates a positive moment, while the thrust of rotor 3 generates a negative moment about the y-axis. The total moment can be expressed as:

$$M_y = F_1 l - F_3 l = (K_f \Omega_1^2)l - (K_f \Omega_3^2)l = K_f l(\Omega_1^2 - \Omega_3^2) \quad (6.14)$$

For the moments about the body frame's z-axis, the thrust of the rotors does not cause a moment. However, moment caused by the rotors' rotation is per Equation 6.9. Then, by using the right-hand-rule, the moment about the body frame's z-axis can be expressed as:

$$\begin{aligned} M_z &= M_1 - M_2 + M_3 - M_4 \\ &= (K_M \Omega_1^2) - (K_M \Omega_2^2) + (K_M \Omega_3^2) - (K_M \Omega_4^2) \\ &= K_M (\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{aligned} \quad (6.15)$$

Combining equations (6.10), (6.11) and (6.12) in vector form, we get:

$$M_B = \begin{bmatrix} K_f l(-\Omega_2^2 + \Omega_4^2) \\ K_f l(\Omega_1^2 - \Omega_3^2) \\ K_M (\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{bmatrix} \quad (6.16)$$

where:

- l is the moment arm, which is the distance between the axis of rotation of each rotor to the origin of the body reference frame which should coincide with the center of the quadcopter.

The roll and pitch torque ($\tau\phi, \tau\theta$) acting on the body frame are produced by changing the motors' thrust. However, the yaw torque, $\tau\psi$, is generated by differential thrust between the two pairs of counter-rotating motors. This momentum is produced by the rotors' reactive torque, namely Q , defined as:

$$Q_i = d\Omega_i \quad (6.17)$$

where:

- d is a rotor induced torque drag coefficient dependent on the propeller specifications
- Ω_i is the propeller's angular velocity

Therefore, the acting torque $\tau^b = [\tau_\phi, \tau_\theta, \tau_\psi]$ on body fixed frame are:

$$\tau_\phi = \frac{\sqrt{2}}{2} l [(F_2 + F_3) - (F_1 + F_4)] \quad (6.18)$$

$$\tau_\theta = \frac{\sqrt{2}}{2} l [(F_1 + F_2) - (F_3 + F_4)] \quad (6.19)$$

$$\tau_\psi = Q_1 - Q_2 + Q_3 - Q_4 \quad (6.20)$$

6.5 Nongravitational Forces

When the quadcopter is in a horizontal orientation (i.e. it is not rolling or pitching), the only nongravitational forces acting on it is the thrust produced by the rotation of the propellers which is proportional to the square of the angular velocity of the propeller (see Equation 6.9). Therefore, the nongravitational forces acting on the quadcopter, F_B , can be expressed as:

$$F_B = \begin{bmatrix} 0 \\ 0 \\ -K_f(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{bmatrix} \quad (6.21)$$

Since there are no forces in the X and Y directions, the first two rows of the force vector are zeros. The last row is the thrust forces produced by the four propellers. The negative sign is because the thrust is upwards while the positive z-axis in the body framed is pointing downwards. To apply the equation in any orientation of the quadcopter, F_B is multiplied by the rotation matrix R to transform the thrust forces of the rotors from the body frame to the inertial frame.

6.6 Drag Forces

Due to the friction of the moving quadcopter body with air, another force acts on the body frame of the quadcopter resisting the motion. As the velocity of travel of the quadcopter increases, the drag forces in turn increase. Drag force as a function of a drag coefficient C_D is described as:

$$F_d = -\frac{1}{2} \rho C_d A v^2$$

$$C_d = \frac{1}{2\rho A v^2} \quad (6.22)$$

where:

- F_d is drag force in Newtons
- ρ is the density of air in kg/m³
- C_d is the drag coefficient
- A is the blade area
- v is the velocity in m/s

The drag forces F_a can be approximated by:

$$F_a = K_t \dot{r} \quad (6.23)$$

where:

K_t is an aerodynamic effect (constant matrix called the aerodynamic translation coefficient matrix) that can be neglected since the helicopter's motion is planned to be relatively slow [31]

\dot{r} is the time derivative of the position vector r .

6.7 Drag Moments

The same as the drag force, due to the air friction, there is a drag moment M_a acting on the quadcopter body which can be approximated by:

$$M_a = K_r \dot{\eta} \quad (6.24)$$

where:

K_r is a constant matrix called the aerodynamic rotation coefficient matrix

$\dot{\eta}$ represents the Euler rates

As mentioned in Gopalakrishnan [20], for the quadcopter model and simulation, the drag moment is neglected since the effect of these forces inside a room is small compared to the thrust produced by the quadcopter.

6.8 Total forces

The total acting force in body frame as result of all the forces discussed in the previous sections is:

$$F^b = F_a + F_g^b - \sum_{i=1}^4 F_i u_z \quad (6.25)$$

6.9 Rotational equations of motion

The rotational equations of motion are derived in the body frame to have the inertia matrix independent on time. Using the Newton-Euler method general formalism, the equation is:

$$J\dot{\omega} + \omega \times J\omega + M_G = M_B \quad (6.26)$$

where:

$J\dot{\omega}$ and $\omega \times J\omega$, represent the rate of change of angular momentum in the body frame

J is the quadcopter's diagonal inertia matrix

ω is the angular body rates

M_G represents the gyroscopic moments due to rotors' inertia J_r

M_B represents the moments acting on the quadcopter in the body frame

Considering all moments acting on the quadcopter, the rotational equation of the quadcopter's motion can be written as seen in [22], [13], and [26].

$$J\dot{\omega} + \omega \times J\omega + \omega \times [0 \ 0 \ J_r \Omega_r]^T = M_B - M_a \quad (6.27)$$

where:

J is the rotors' inertia

Ω_r is the rotors' relative speed. $\Omega_r = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4$

M_a is the drag moment acting on the quadcopter body

6.10 Translational Equations of Motion

The translation equations of motion for the quadcopter are based on Newton's second law and are derived in the Earth inertial frame. Considering the extra forces acting on the quadcopter body, the translational equation of motion is

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + RF_B - F_a \quad (6.28)$$

where:

$r = [x \ y \ z]^T$ is the quadcopter's distance from the inertial frame

m is the quadcopter's mass

g is the gravitational acceleration $g = 9.81 \text{ m/s}^2$

F_B is the nongravitational forces acting on the quadcopter in the body frame

F_a is the drag force in Newtons

7. From mathematical model to state space model

The mathematical model for the quadcopter will be formulated into a state space model to facilitate the handling of the control problem.

7.1 State Vector X

The state vector defines the position of the quadcopter in space and its angular and linear velocities. Considering the Quadcopter 6DOF, the state vector of the quadcopter is mapped as follows:

$$X = [\phi \dot{\phi} \theta \dot{\theta} \psi \dot{\psi} \ z \dot{z} \ x \dot{x} \ y \dot{y}]^T \quad (7.1)$$

7.2. Control Input Vector U

A control input vector, U , consisting of four inputs; $U1$ through $U4$ is defined as:

$$U = [U_1 \ U_2 \ U_3 \ U_4]^T \quad (7.2)$$

where:

$$U_1 = K_f(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \quad (7.3)$$

$$U_2 = K_f(-\Omega_2^2 + \Omega_4^2) \quad (7.4)$$

$$U_3 = K_f(\Omega_1^2 - \Omega_3^2) \quad (7.5)$$

$$U_4 = K_M(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \quad (7.6)$$

Equations (7.3) through (7.6) can be arranged in a matrix form to result in:

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} K_f & K_f & K_f & K_f \\ 0 & -K_f & 0 & K_f \\ K_f & 0 & -K_f & 0 \\ K_M & -K_M & K_M & -K_M \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \quad (7.7)$$

U_1 is the resulting upwards force of the four rotors which is responsible for the altitude of the quadcopter z and its rate of change \dot{z} , thus generating the desired altitude of the quadcopter.

U_2 is the difference in thrust between rotors 2 and 4 which is responsible for the roll rotation ϕ and its rate of change $\dot{\phi}$, that will generate the desired roll angle.

U_3 represents the difference in thrust between rotors 1 and 3 thus generating the pitch rotation θ and its rate of change $\dot{\theta}$.

U_4 is the difference in torque between the two clockwise turning rotors and the two counterclockwise turning rotors generating the yaw rotation ψ and its rate of change $\dot{\psi}$, that will generate the desired heading.

If the rotor velocities are needed to be calculated from the control inputs, it can be obtained by first inverting the matrix in Equation 7.8 giving:

$$\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4K_f} & 0 & \frac{1}{2K_f} & \frac{1}{4K_M} \\ \frac{1}{4K_f} & -\frac{1}{2K_f} & 0 & -\frac{1}{4K_M} \\ \frac{1}{4K_f} & 0 & -\frac{1}{2K_f} & \frac{1}{4K_M} \\ \frac{1}{4K_f} & \frac{1}{2K_f} & 0 & -\frac{1}{4K_M} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \quad (7.8)$$

then taking the square root, the rotors' velocities will be:

$$\Omega_1 = \sqrt{\frac{1}{4K_f} U_1 + \frac{1}{2K_f} U_3 + \frac{1}{4K_M} U_4} \quad (7.9)$$

$$\Omega_2 = \sqrt{\frac{1}{4K_f} U_1 - \frac{1}{2K_f} U_2 - \frac{1}{4K_M} U_4} \quad (7.10)$$

$$\Omega_3 = \sqrt{\frac{1}{4K_f} U_1 - \frac{1}{2K_f} U_3 + \frac{1}{4K_M} U_4} \quad (7.11)$$

$$\Omega_4 = \sqrt{\frac{1}{4K_f} U_1 + \frac{1}{2K_f} U_2 - \frac{1}{4K_M} U_4} \quad (7.12)$$

7.3. Rotational Equation of Motion

Substituting equations (7.3) through (7.6) in equation (6.16), the equation of the total moments acting on the quadcopter becomes:

$$M_B = \begin{bmatrix} lU_2 \\ lU_3 \\ U_4 \end{bmatrix} \quad (7.10)$$

Substituting (7.10) into the rotational equation of motion (6.24) and expanding each term with their prior definition in this chapter, the following relation can be derived:

$$\begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ J_r \Omega_r \end{bmatrix} = \begin{bmatrix} lU_2 \\ lU_3 \\ U_4 \end{bmatrix} \quad (7.11)$$

Expanding and rewriting equation 7.11 to have the angular accelerations in terms of the other variables [22], we have:

$$\ddot{\phi} = \frac{l}{I_{xx}} U_2 - \frac{J_r}{I_{xx}} \dot{\theta} \Omega_r + \frac{I_{yy}}{I_{xx}} \dot{\theta} \dot{\psi} - \frac{I_{zz}}{I_{xx}} \dot{\theta} \dot{\psi} \quad (7.12)$$

$$\ddot{\theta} = \frac{l}{I_{yy}} U_3 - \frac{J_r}{I_{yy}} \dot{\phi} \Omega_r + \frac{I_{zz}}{I_{yy}} \dot{\phi} \dot{\psi} - \frac{I_{xx}}{I_{yy}} \dot{\phi} \dot{\psi} \quad (7.13)$$

$$\ddot{\psi} = \frac{l}{I_{zz}} U_4 + \frac{I_{xx}}{I_{zz}} \dot{\theta} \dot{\phi} - \frac{I_{yy}}{I_{zz}} \dot{\theta} \dot{\phi} \quad (7.14)$$

7.4 Translational Equation of Motion

Substituting equation (7.3) in equation (6.21), the equation of the total moments acting on the quadcopter becomes:

$$F_B = \begin{bmatrix} 0 \\ 0 \\ -U_1 \end{bmatrix} \quad (7.18)$$

Embedding equation (7.18) into the translational equation of motion (6.28) and the following expression is obtained by expanding the terms:

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -U_1 \end{bmatrix} \quad (7.19)$$

Rewriting Equation (7.19) to have the accelerations in terms of the other variables [22]:

$$\ddot{x} = \frac{-U_1}{m} (\sin \phi \sin \psi + \cos \phi \cos \psi \sin \theta) \quad (7.20)$$

$$\ddot{y} = \frac{-U_1}{m} (\cos \phi \sin \psi \sin \theta - \cos \psi \sin \phi) \quad (7.21)$$

$$\ddot{z} = g \frac{-U_1}{m} (\cos \phi \cos \theta) \quad (7.22)$$

Rewriting in terms of the state variable X :

$$\ddot{x} = \frac{-U_1}{m} (\sin x_1 \sin x_5 + \cos x_1 \cos x_5 \sin x_3) \quad (7.23)$$

$$\ddot{y} = \frac{-U_1}{m} (\cos x_1 \sin x_5 \sin x_3 - \cos x_5 \sin x_1) \quad (7.24)$$

$$\ddot{z} = g \frac{-U_1}{m} (\cos x_1 \cos x_3) \quad (7.25)$$

Consequently, the resulting translational subsystem is dependent on the translational state variables and the rotational state variables causing it to be underactuated Nabil ElKholy [22]. The complete mathematical nonlinear quadcopter model [32] is:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \cos \theta \cos \psi & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & \cos \psi \sin \theta \sin \phi + \sin \psi \cos \phi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \psi & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} + \begin{bmatrix} -g \sin \theta \\ g \cos \theta \sin \phi \\ g \cos \theta \cos \phi \end{bmatrix} + \frac{1}{m} \begin{bmatrix} 0 \\ 0 \\ -F^b \end{bmatrix}$$

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

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{J_{yy} - J_{zz}}{J_{xx}} qr \\ \frac{J_{zz} - J_{xx}}{J_{yy}} pr \\ \frac{J_{xx} - J_{yy}}{J_{zz}} pq \end{bmatrix} + \begin{bmatrix} \frac{1}{J_{xx}} \tau \phi \\ \frac{1}{J_{yy}} \tau \theta \\ \frac{1}{J_{zz}} \tau \psi \end{bmatrix}$$

8. Power Considerations

The quadcopter's battery is an essential component in the quadcopter system since flight would not be possible without it. Furthermore, the quadcopter's flight time is directly proportional to the battery's capacity. Besides flight time, it is also important to take into consideration the quadcopter's power demand. This can be determined through Equation 8.1 which is the generalized power equation for a helicopter while hovering [33].

$$P = \frac{T^{3/2}}{FM\sqrt{2\rho A}} \quad (8.1)$$

where:

P is hover power

T is weight of the UAV

ρ is density

A is rotor/propeller area

FM is Figure of Merit

Equation (8.1) provides the real value of power that produces usable thrust. It contains a value of efficiency of power conversion known as the figure of merit (FM). The FM is a quantity used to characterize the performance of the rotor. This quantity typically varies from 0.7 to 0.8 in the case that the quadcopter is hovering [12].

C. Testing Phase

This section of the project may be divided into various segment such as: Power demand test, dynamometer test, MATLAB/Simulink tests, ground control station tests, and additional Honeywell requirement validation tests. The following content describes how each segment of the testing phase was executed.

Power demand test

Objective

Determine the power consumption per weight produced by the motor throttle necessary to lift $\frac{1}{4}$ of the quadcopter' weight.

Equipment

- 1xPower Analyzer
- 1xLiPo battery
- 1xESC
- 1xMotor
- 1xMotor Mount
- 1xPropeller
- 1xScale
- 1xRemote Controller
- 1xRemote Controller Receiver
- 4xAA Batteries
- Tape
- Velcro

Procedure or description

1. The motor mount containing motor with propeller was attached to the scale with Velcro and tape.
2. The necessary connections were executed as shown in Figure 6C.2 according to the diagram presented in Figure 6C.1.

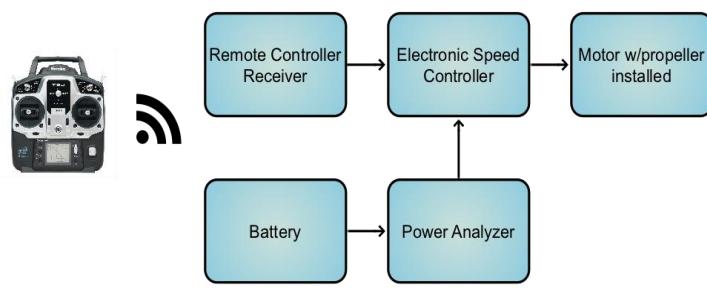


Figure 6C.1: Diagram for measuring power demand test

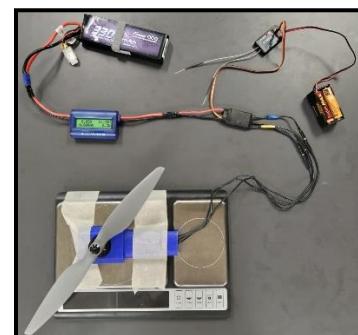
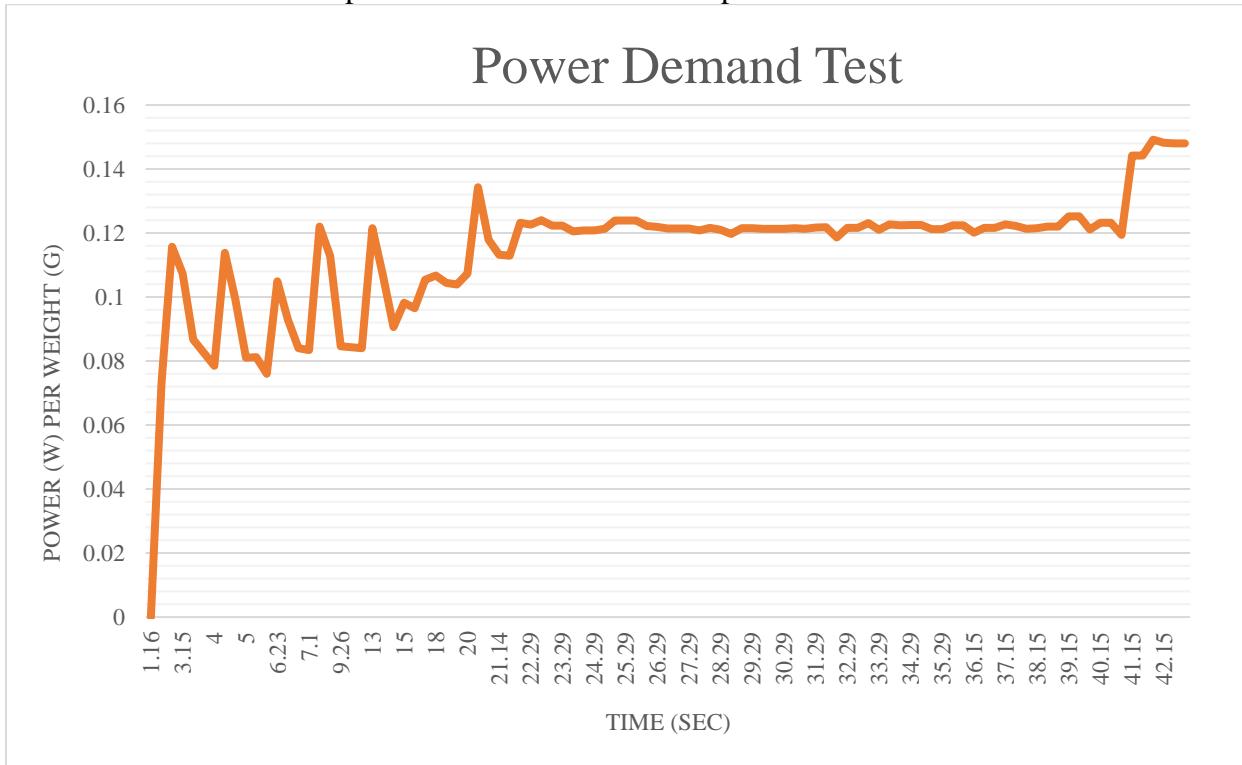


Figure 6C.2: Executed connections for measuring power demand

- 2a. The power analyzer's output cables were soldered to the ESC's power input cables.
- 2b. The motor's input phase cables were connected to their corresponding output phase cables in the ESC.
- 2c. The ESC's PWM input cable was connected to the remote controller receiver and the remote controller receiver was energized.

3. The remote controller was powered on and, while recording the measurements from the scale and the power analyzer with a video recorder, the motor's throttle was increased until it reached the throttle that was necessary to lift $\frac{1}{4}$ of the quadcopter' weight.
4. The power analyzer's recorded measurements were transferred to a table from which Graph 6C.1 was developed to verify the motor's power consumption per weight produced by the motor's throttle.

Graph 6C.1: Results from motor power demand test



From the data that was obtained while the motor's throttle was maintained at an approximately constant rate (from the time period of 22.29 s to 40.15 s), it may be stated that the motor consumes approximately 0.12 W/g.

Dynamometer test

Objective

Determine the thrust and drag factor for the selected motor-propeller configuration.

Equipment

- 1xLiPo battery
- 1xESC
- 1xMotor
- 1xMotor Mount
- 1xPropeller
- 1xDynamometer
- 1xLaptop w/RC benchmark software

Procedure or description

1. The necessary connections were executed as shown in Figure 6C.4 according to the diagram presented in Figure 6C.3.

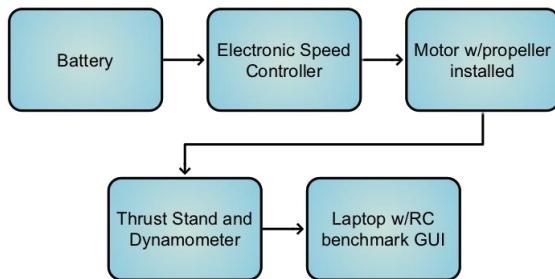


Figure 6C.3: Diagram for dynamometer test



Figure 6C.4: Executed connections for dynamometer test

2. The RC benchmark graphical user interface was opened, and the benchmark was initiated.
3. The thrust and power coefficient were calculated from the outputted excel table and the values were recorded in Table 6C.2.

Table 6C.2: Characterizing motor-propeller configuration

Thrust coefficient (C_t)	Power coefficient (C_p)
0.102	0.0449
Thrust factor (b)	Drag factor (d)
1.9366×10^{-5}	3.7909×10^{-7}

MATLAB/Simulink Tests

Objective

Determine a quadcopter model from which quadcopter flight may be successfully simulated.

Equipment

1xLaptop w/MATLAB & Simulink software

Procedure or description

The following section describe the three main Simulink models that were used as reference to determine the most appropriate control technique, simulate UAV flight, and develop the final UAV Simulink model for the designed quadcopter which is shown in the Project Results section.

Simulink Reference #1: Quadrotor Dynamic Model

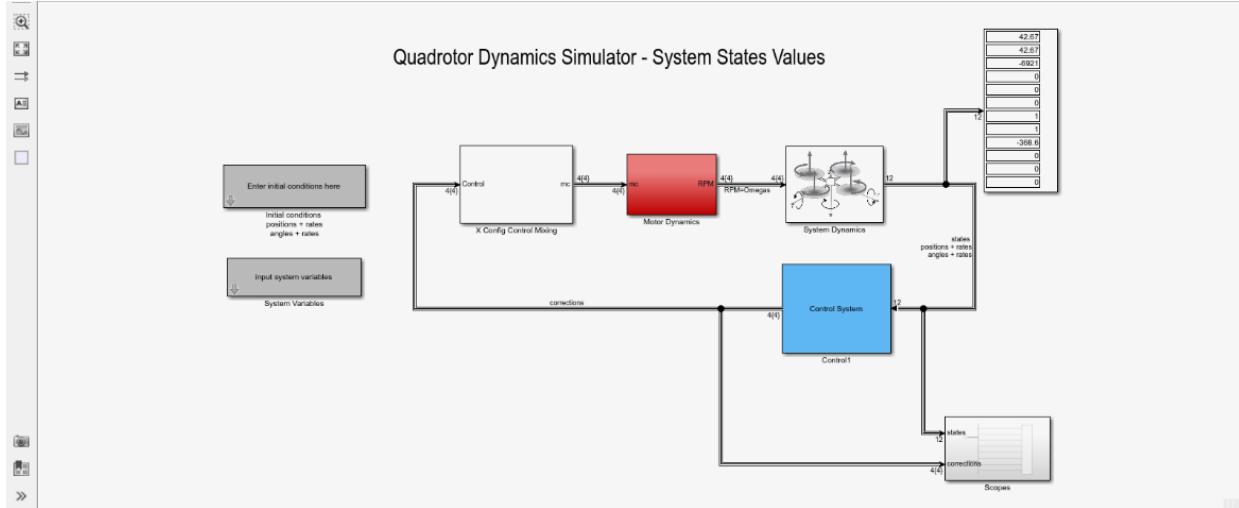


Figure 6C.1: Simulation of Quadrotor Dynamic Model

The performance of the Quadrotor Dynamic Model, shown in Figure 6C.1, did not result as expected. Consequentially, the Simulink model was modified by using Simulink's PID Tuner application to adjust the values for the proportional, integral, and derivative gains in each of the PID controllers to be used to control a quadcopter system. The PID gain values were adjusted until the desired performance was achieved.

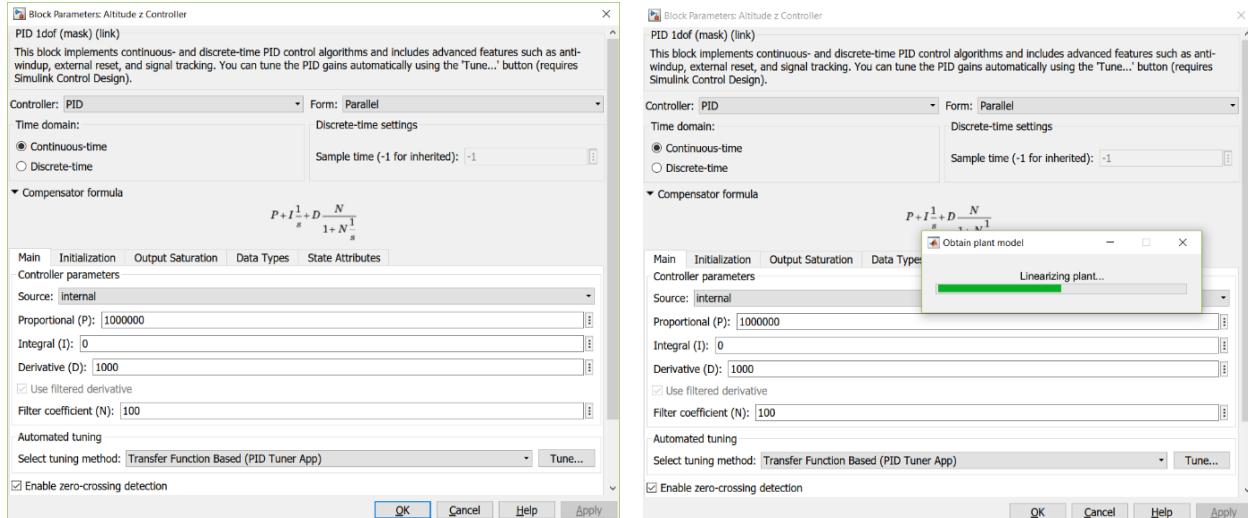


Figure 6C.2 Attitude PID block tuning

Figure 6C.2 shows the Simulink PID block and the “Tune” option which invokes the tuner application. The PID Tuner automatically linearizes the plant at the operating point specified by the Simulink model’s initial conditions. Consequentially, the tuner parameters were adjusted to find the set of gains that ensured the best performance of the control system. However, the gains that were obtained from the tuner were either too high or too low. Furthermore, when the “tuned” values were applied to the quadcopter’s PID controllers, the result from the quadcopter Simulink model would always be an unstable system as shown in Figures 6C.3 thru 6C.7.

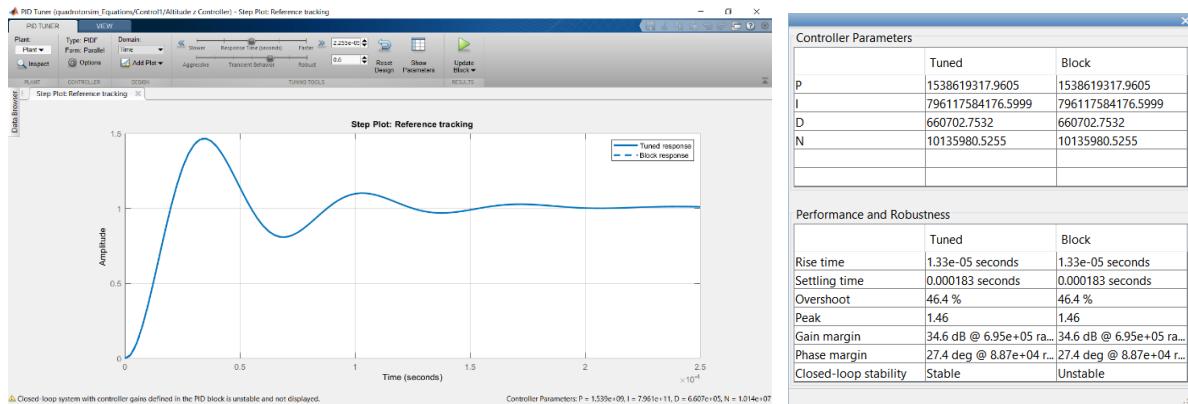


Figure 6C.3: Attitude PID tuning result

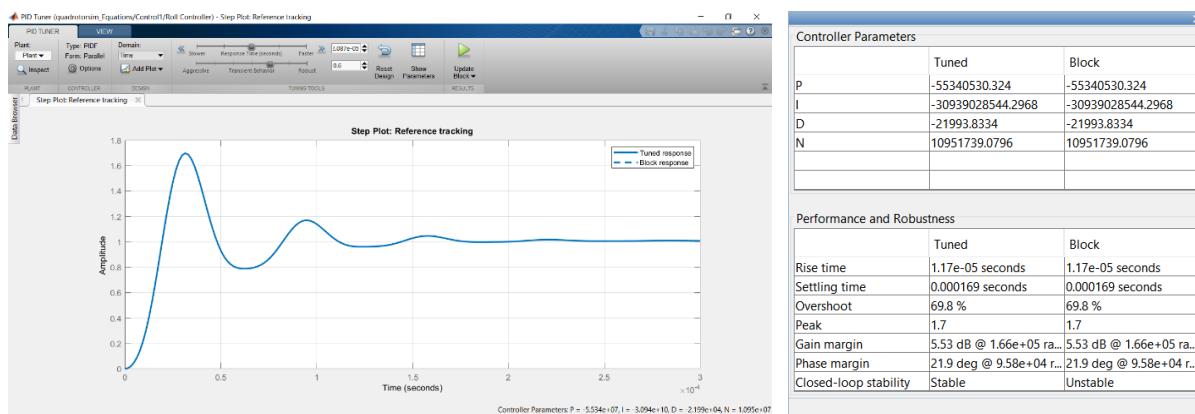


Figure 6C.4: Roll PID tuning result

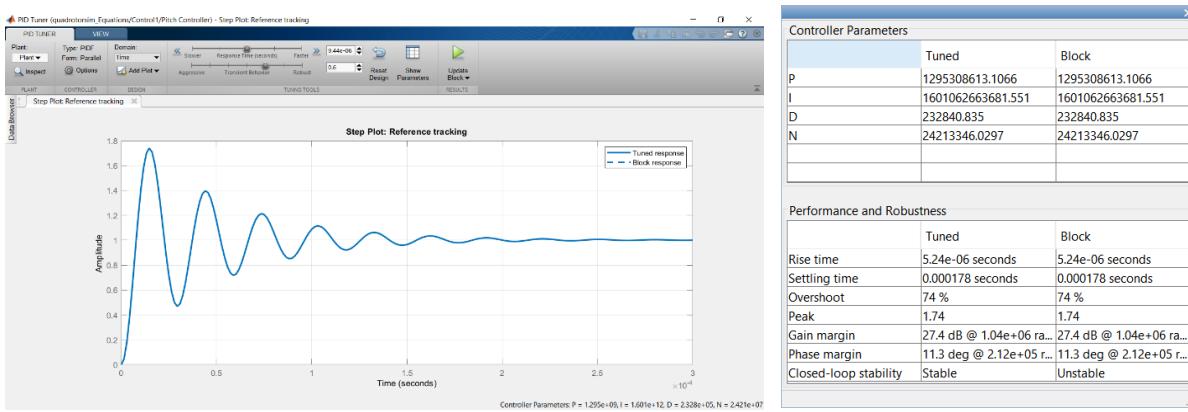


Figure 6C.5: Pitch PID tuning result

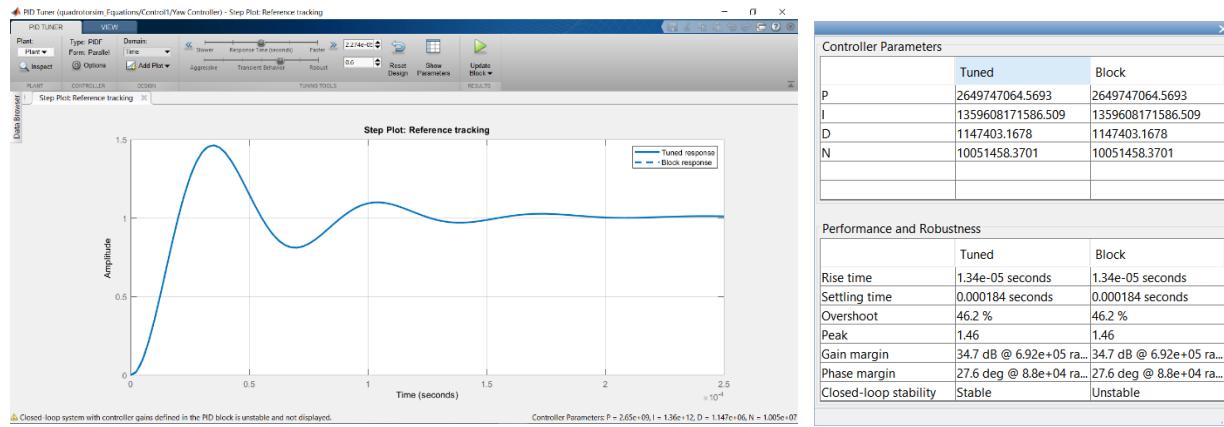


Figure 6C.6: Yaw PID tuning result

After a stable system was obtained, the simulation was executed to evaluate the step responses for the desired and estimated (corrected) altitude (z), roll, pitch and yaw values as shown in Figure 6C.7. The simulation took about 3 hours to complete for a sample of 3 seconds.

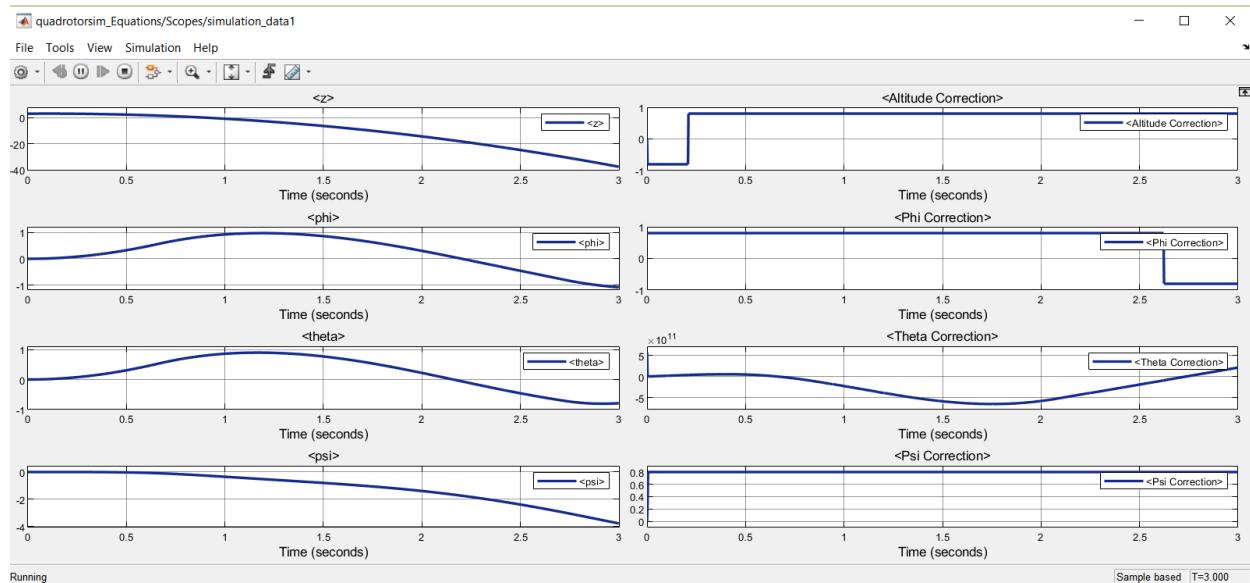


Figure 6C.7: Simulation result after tuning: linearized model response

Simulink Reference #2: sl_quadrotor model

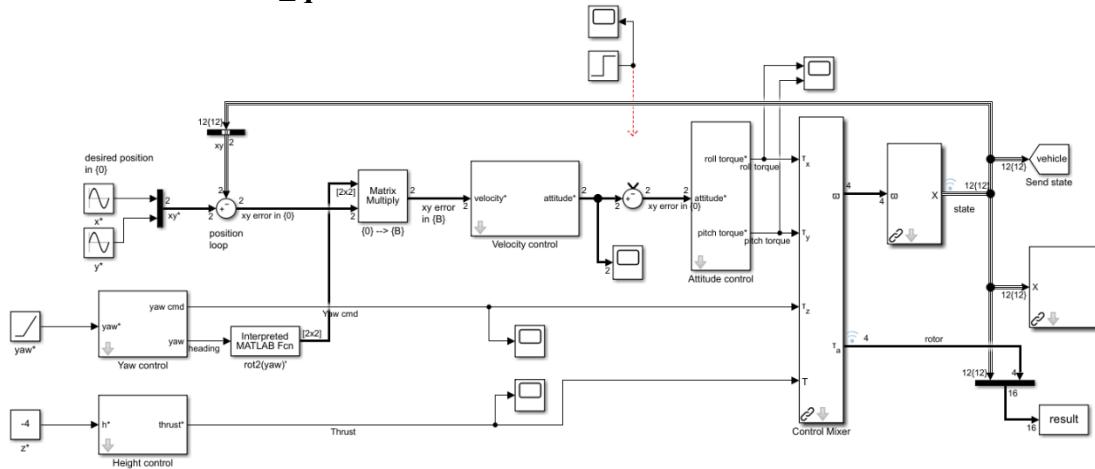


Figure 6C.8: Simulation of sl_quadrotor model without disturbance

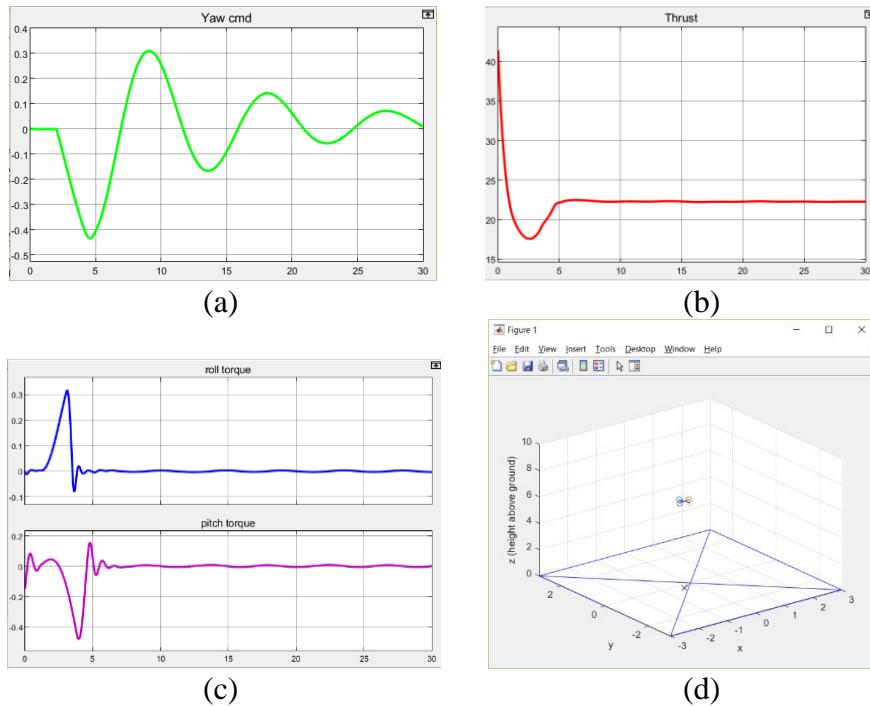


Figure 6C.9: sl_quadrotor model simulation without disturbance responses (a), (b), (c) and plot

The sl_quadrotor model was set up with the identified physical parameters of the designed quadrotor. Then, the sl_quadrotor model was run to execute the tuning procedure. The step responses for each PD controller are shown in Figure 6C.9 (a), (b) and (c). The quadrotor trajectory plot is shown in Figure 6C.9 (d). To further study the Simulink model representing the quadrotor system, a disturbance was added, and the test results are shown in Figures 6C.10 and 6C.11. The step responses were significantly similar to those obtained while the model had no integrated disturbances. Thus, it may be stated that the model was adjusted correctly.

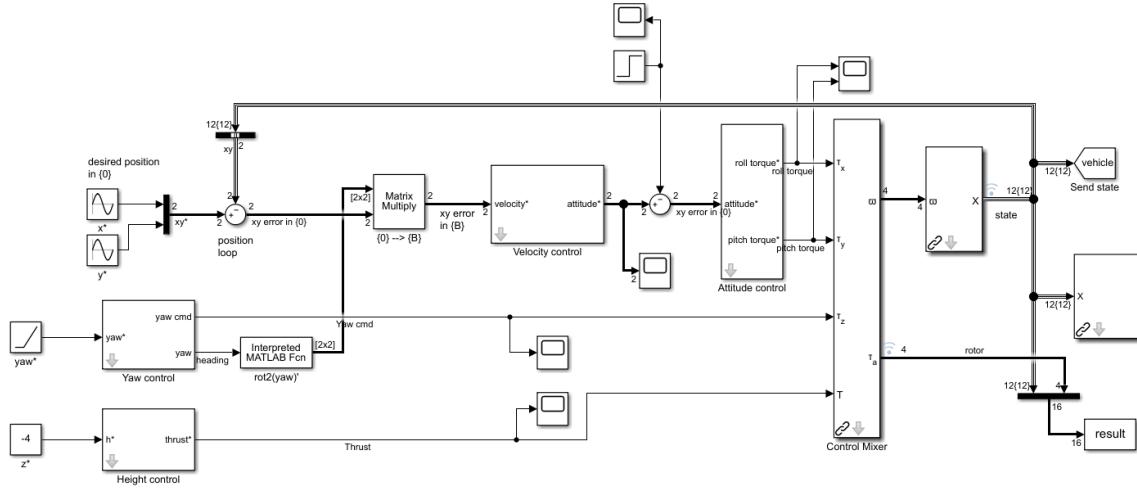


Figure 6C.10: Simulation of sl_quadrotor model with disturbance

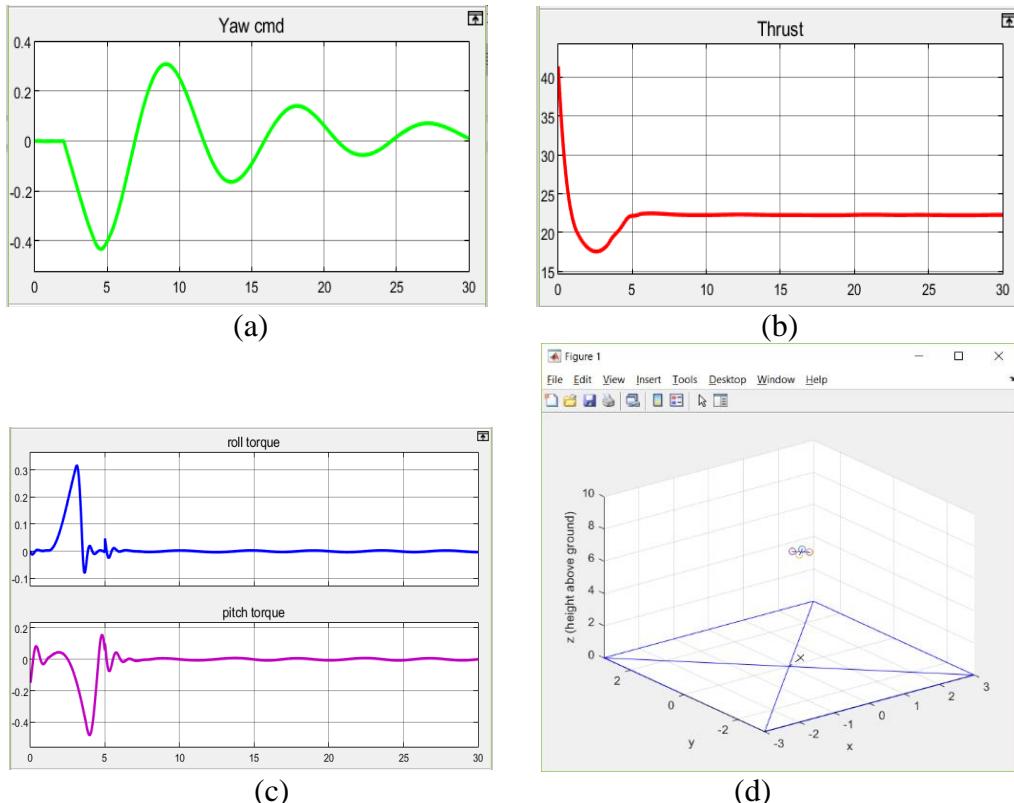


Figure 6C.11: sl_quadrotor model simulation with disturbance responses (a), (b), (c) and plot (d)

This model was tested for the purpose of acquiring a model that was able to serve as a guide to plotting a random UAV trajectory. The plot of this model was limited to a predefined circular trajectory which represent a disadvantage to our requirements.

Simulink Reference #3: QuadrotorSimulink model

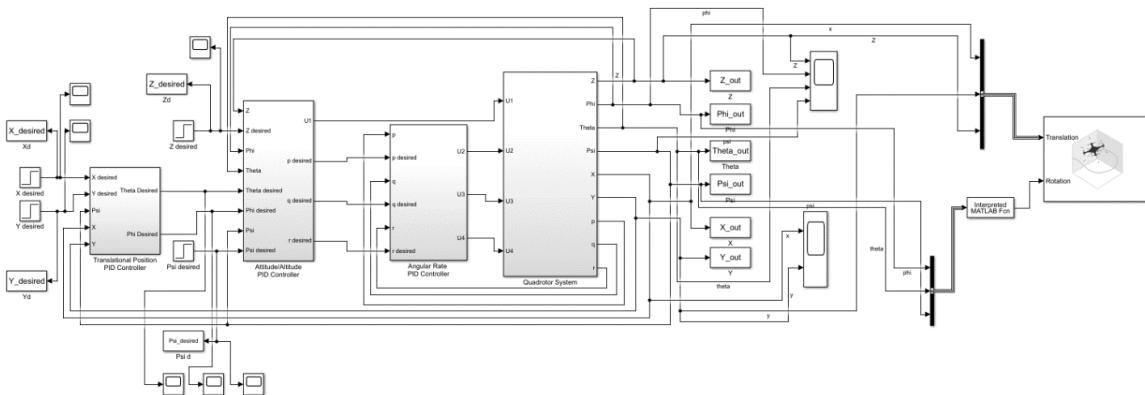


Figure 6C.12: Simulation of QuadrotorSimulink model

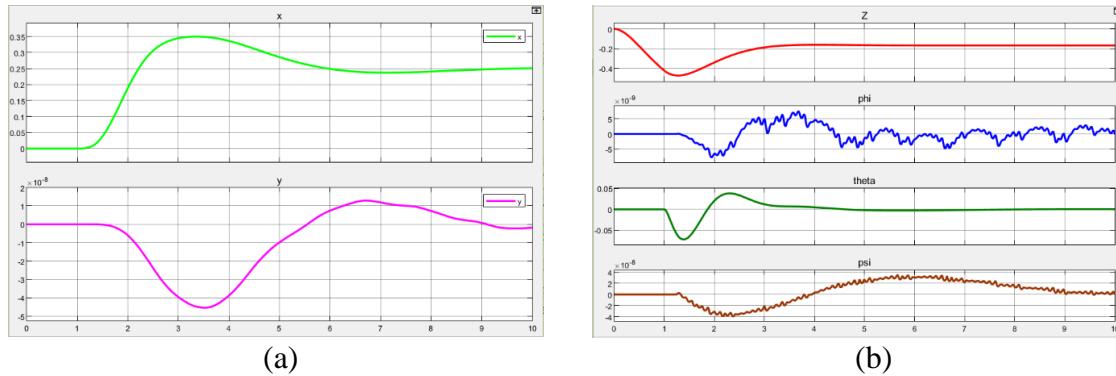


Figure 6C.13: QuadrotorSimulink model (a) XY translational response and (b) Z-altitude and rotational response

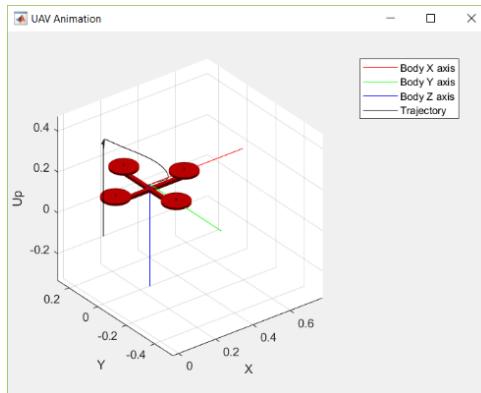


Figure 6C.14: QuadrotorSimulink model plot

Similar to the sl_quadrotor model, this model included a plot of the quadrotor trajectory. The difference is that the Simulink UAV Animation block was used instead of the plot3 function. The advantage of this model was that the UAV Animation block represented the UAV flight path based on the translations and rotations estimated by the model. However, the results were limited since the model did not permit waypoints to be inputted.

4. Ground Control Station Tests

Objective

Establish connectivity between ground control station and quadcopter and then, proceed to calibrate quadcopter.

Equipment

- 1xESC
- 1xMotor
- 1xMotor Mount
- 1xRemote Controller
- 1xRemote Controller Receiver
- 1xWiFi telemetry module
- 1xGPS module
- 1xLaptop/w GCS software
- 2xRadio telemetry module

Procedure or description

Technical difficulties arose while executing the ground control station tests. Initially, QGroundControl was used to calibrate the quadcopter control board and remote controller. While the wired connection between the Pixhawk and QGroundControl proved to be reliable, the wireless connection was problematic.

The first problem arose while testing the Wi-Fi telemetry module ESP8266. To successfully implement this module with the Pixhawk control board, which uses MAVLink to communicate with the GCS, the proper firmware had to be uploaded to the module so that it could function as an ESP8266 Wi-Fi access point and MAVLink bridge. However, the firmware could not be uploaded to the Wi-Fi module. Thus, the Wi-Fi module was substituted with a set of two radio telemetry modules which already came with the necessary firmware installed. One telemetry module was called the air module and was connected to the control board. The other telemetry module was called the ground module and was connected to the laptop running the GCS. On the other hand, another problem emerged while implementing the radio telemetry modules with the QGroundControl station seen in Figure 6C.15.

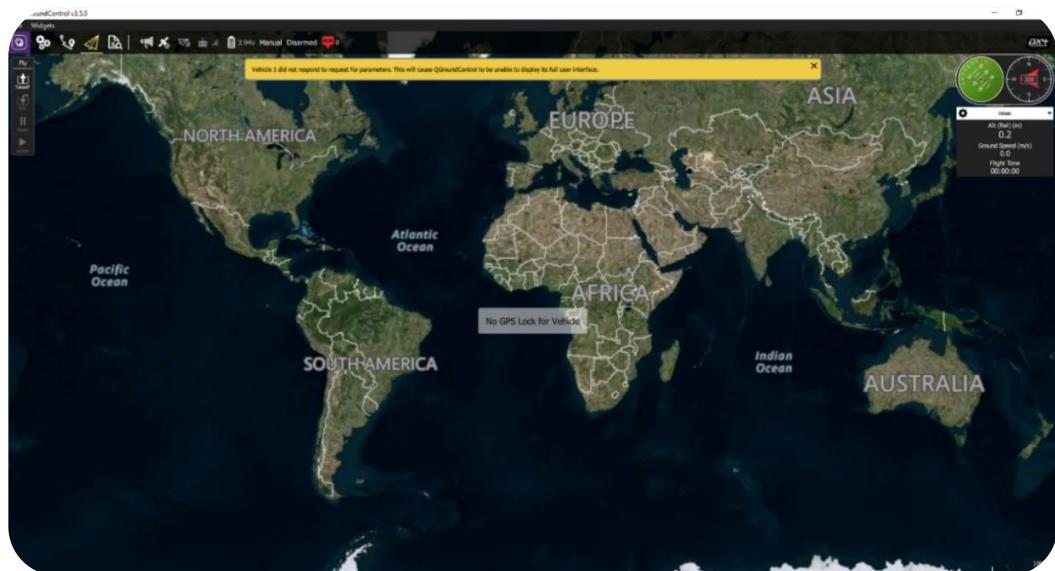


Figure 6C.15: QGroundControl graphical user interface

The QGroundControl GCS automatically installed a firmware into the radio telemetry ground module which was different from the firmware installed in the radio telemetry air module. Thus, the air module was left unable to communicate with the ground module. For this reason, the Mission Planner GCS seen in Figure 6C.16 was implemented instead.



Figure 6C.16: Mission Planner graphical user interface

Through the Mission Planer GCS, the radio telemetry modules were successfully installed. However, the Mission Planner was designed to be used with the different control board from the one that was currently being implemented (Pixhawk Cube controller). As a result, the Pixhawk Cube flight controller was substituted with the Ardupilot v2.6 flight controller and the Here+ GPS designed for Pixhawk was substituted with the XT-XINITE GPS which was compatible with the Ardupilot.

5. Additional Honeywell Requirement Validation Tests

From another point of view, through the testing phase it was important to ensure compliance with Honeywell's requirements. After eliminating the autonomy requirement from the competition, the revised requirements could be summarized as follows:

1. Vehicle, motor, and all electronics shall operate using internal batteries and selected internal battery shall comply with ESC/motor and Pixhawk flight controller power demand.
2. All electronics on vehicle shall be properly insulated
3. At power up, vehicle shall not turn on its motors until commanded by pilot/user and power button shall be operational and easily accessible for the user
4. All wires will be properly secured to the vehicle structure and kept away from its propellers and fuselage will have provisions to secure wiring and cable harnesses
5. Fuselage shall provide enough weatherproofing for internal electronics to protect said electronics from rain, dust, and dirt
6. Batteries must be installed in such a way as to avoid direct sunlight and batteries must be installed in such a way as to have an adequate heat transfer to avoid overheating during operation

Requirement 1 was validated through the power demand test previously described. Then, requirements 2-6 were validated through visual inspection ensuring the following: drone was properly insulated by implementing heat shrink tubes or electric tape on soldered wires, installed power button functioned properly, battery was placed below the drone to avoid sunlight (shown in Figure 6C.17) and electronics were effectively protected by rain, dust and dirt through its Lexan Casing (shown in Figure 6C.18).



Figure 6C.17: Visual inspection for requirements 2-6



Figure 6C.18: Visual inspection for UAV casing

D. Manufacturing and Wiring

The designed UAV was completely built from scratch excluding its propellers. Thus, this section demonstrates the process in which the UAV was manufactured and how its electrical components were wired together. The ME team made use of the 3D printers available at PUPR to create many of the physical components that would be part of the UAV such as the: arms, arm mounts, motor mounts, landing gear, and propeller guards. The materials used for 3D printing were Polylactic Acid (PLA), Nylon G, and Nylon X. Additionally, the ME team made use of the laser cutter available at PUPR to shape Lexan material into UAV physical components such as the: casing and base plates. The manufacturing plan that was initially devised is shown in Table 6D.1.

Table 6D.1: UAV manufacturing plan

Name of part	Printing time	Manufacturing time	Finished parts	Needed parts	Start date	Due date
Legs (NylonX)	3h	12h	4	4	3/1/2019	3/23/2019
Arms (NylonG)	2.5h	10h	4	4	3/1/2019	3/23/2019
Arm Mounts (PLA)	2h	16h	8	8	3/1/2019	3/23/2019
Motor Mounts (NylonG)	6h	24h	1	4	3/1/2019	3/23/2019
Propeller Guards (NylonX)	13h	52h	0	4	3/1/2019	3/23/2019
Top/Bottom Case (NylonG)	24h	48h	0	2	3/1/2019	3/23/2019
Top/Bottom Base (Lexan)*	0h	0h	0	2	3/1/2019	3/23/2019
Total Amounts		162h	17	28		

After culminating the manufacturing process, additional work was done with the 3D printers to create two pieces to place on the sides of the UAV's top and bottom base to hold the power button required by Honeywell. Once all the UAV's physical components were printed or shaped for their intended use, the EE team proceeded to the wiring phase in which all the UAV's electrical components were put together. These components included the: motors, electronic speed controllers (ESC), flight controller, GPS, telemetry module, remote controller receiver, power management board, and LiPo battery. The following procedure was followed:

1. A motor was placed in each of the 4 motor mounts which connected to the UAV's hollow arms where the motor cables were stored.
2. An ESC was placed between the top and bottom base of the UAV and each output phase cable was connected to its respective input phase cable in a motor. Each motor was connected to an ESC in this manner.
3. A power management board was placed between the top and bottom base of the UAV and each ESC power and ground cable was soldered into a dedicated motor power and ground contact respectively from the power management board.
4. A flight controller was placed on top of the UAV's top base and its power input was connected to the power management board's dedicated flight controller power output.
5. The different sensors and peripherals (GPS, telemetry module, and remote controller) were connected to their respective ports in the flight controller.
6. Each ESC PWM input cable was connected to its respective PWM output pins in the flight controller.
7. Finally, the UAV's casing was installed in the top base of the UAV.

The necessary assembly to build the quadcopter system is shown in more detail through the assembly chart shown below in Figure 6D.1. Additionally, in the following page the final wiring scheme for the quadcopter system is shown in Figure 6D.2.

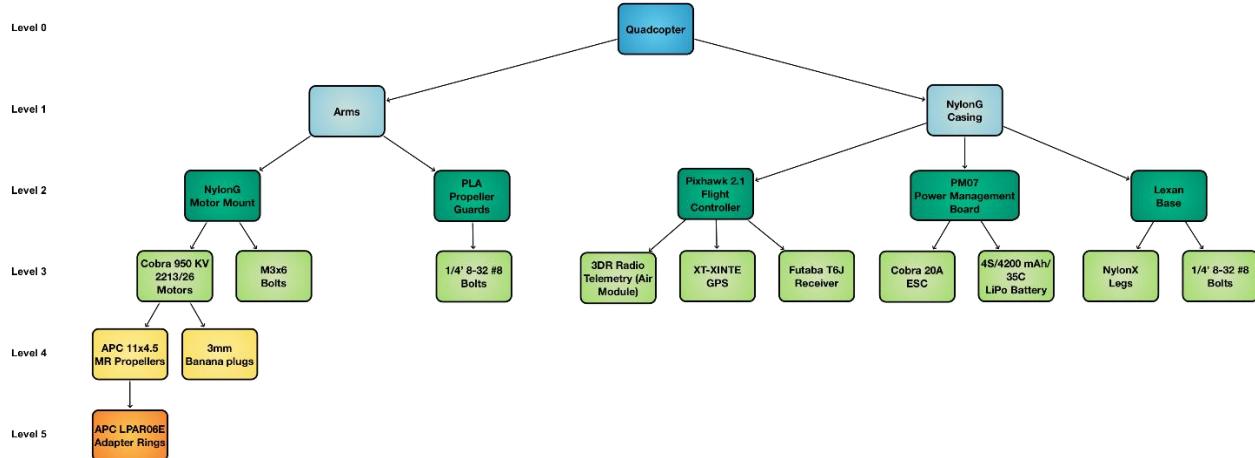


Figure 6D.1: Assembly chart for quadcopter system

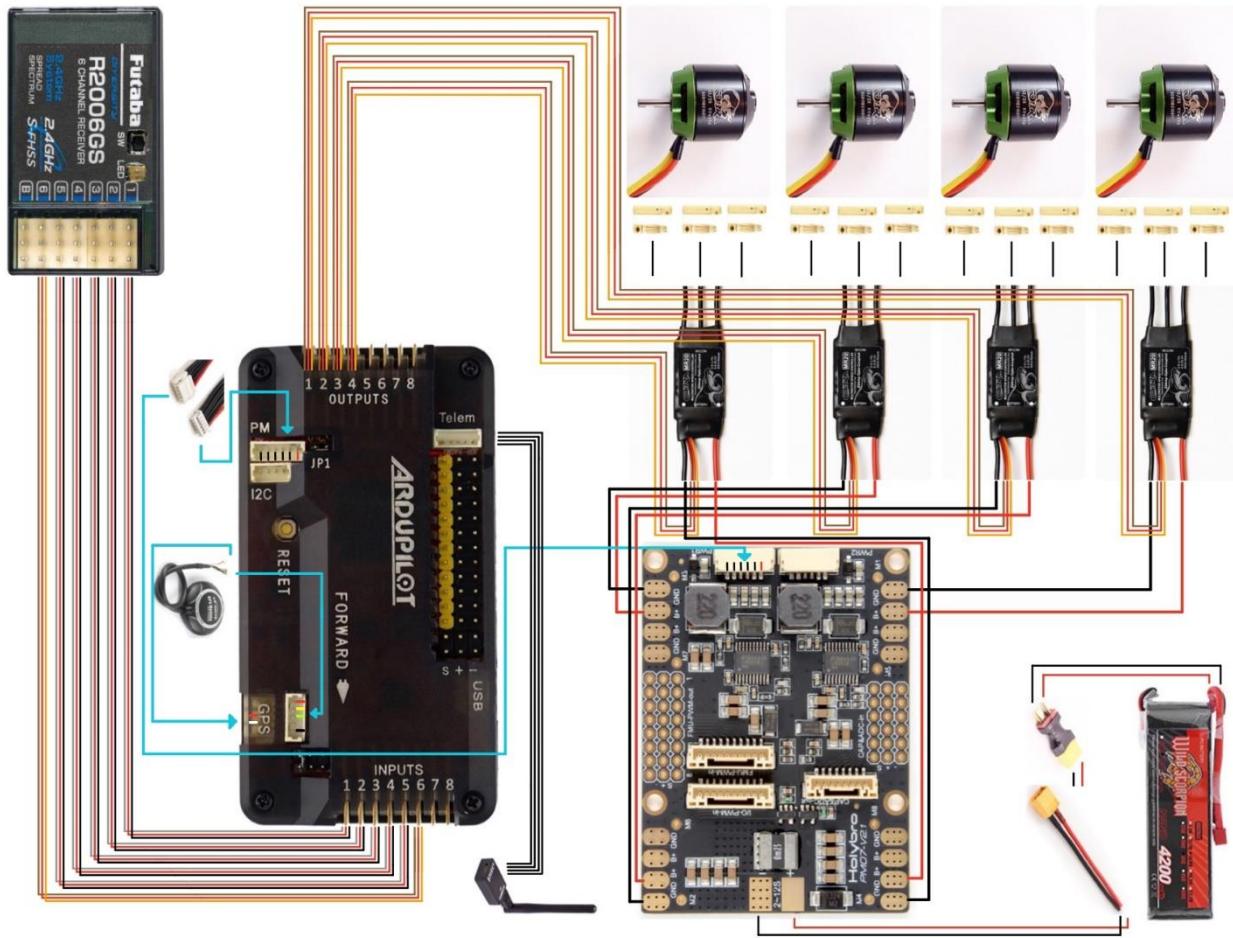


Figure 6D.2: Final wiring scheme

E. System Specifications

The quadcopter components and requirements are shown in Table 6E.1:

Table 6E.1: PUPR Quadcopter System Requirements

Component	Power Requirement	Weight
ArduPilot Mega v2.6 Flight Control Board	Operating Voltage: +5V	28g
X94 3DR Radio Telemetry Kit	Supply voltage:3.7-6V DC	19g
XT-XINTE GPS	Working voltage is 5V	29g
Cobra CM-2213/26 Multirotor Motor	950 RPM per Volt 20A (Max), 300W (Max)	65g
Cobra DL20A Electronic Speed Controller	Operating Voltage: 6-16V; Maximum Continuous Current: 20A	12.2g
Power Supply	LIPO 4S 14.8V 4200MAH 35C	419g
Power Distribution Board	12V and 5V outputs	13g
Futaba 6J 6-Channel S-FHSS Remote Controller System	6-Channel 2.4GHz S-FHSS and FHSS Modes	N/A

The measured and calculated parameters for the designed and manufactured quadcopter system are shown in Table 6E.2:

Table 6E.2: PUPR Quadcopter Model Parameters

Variable	Description	Value	Unit
R	Internal resistance of the motor	0.23	Ohms
K_e	Motor constant	15	V/rad/sec
m	Quadrotor Mass	2.267962	kg
I_{xx}	Quadrotor Moment of Inertia along x axis	0.01928	kg.m^2
I_{yy}	Quadrotor Moment of Inertia along y axis	0.03764	kg.m^2
I_{zz}	Quadrotor Moment of Inertia along z axis	0.02041	kg.m^2
b	Thrust Factor	1.9366E-05	N/A
d	Drag factor	3.79090E-07	N/A
J_r	Rotor Inertia	0.0002662	kg.m^2
l	Length from the rotor to the center of mass	0.260858	m
x_0	Initial X position	1	meters
y_0	Initial Y position	1	meters
z_0	Initial Z position	1	meters
ϕ_0	Initial Roll angle	0	deg
θ_0	Initial Pitch angle	0	deg
ψ_0	Initial Yaw angle	0	deg
\dot{x}_0	Initial Velocity in X direction	1	m/sec
\dot{y}_0	Initial Velocity in Y direction	1	m/sec
\dot{z}_0	Initial Velocity in Z direction	1	m/sec
$\dot{\phi}_0$	Initial Roll rate	0	deg/sec
$\dot{\theta}_0$	Initial Roll rate	0	deg/sec
$\dot{\psi}_0$	Initial Roll rate	0	deg/sec
C_p	Power coefficient	0.0449	N/A
C_t	Thrust coefficient	0.102	N/A
ρ	Air density	1.23	kg.m^3
D_p	Diameter of propeller	0.1143	m

Model Design

Many paths may lead to a functioning Simulink model that presents the simulation of a process along with its controller. Depending on the number of parameters that are considered, the model can be either simple or complex. In the simplest case, the mathematical model of the process has already been derived and the controller is designed through Simulink by observing the processes' response to different inputs.

In this project, the path that led to the development of the final UAV Simulink model, shown in Figure 7A.1, was composed of the following steps: deriving the equations that represent actuator dynamics and quadcopter movement, obtaining the manufactured quadcopter's physical parameters, and observing the different open-source UAV Simulink models that were found to serve as a reference for the final UAV Simulink model that was designed.

The main factors that were taken into consideration while observing the different Simulink models that served as reference for the design of the final UAV Simulink model were the following: implemented control technique and control system parameters, implemented equations representing actuator dynamics and quadcopter movement, implemented sensor readings, and implemented method for state estimation. Additional factors that were observed in some of the reference models were the following: determined UAV trajectory, modeled possible disturbances to UAV flight and implemented Kalman filter.

From another point of view, while designing the UAV Simulink model, it was important to recognize that quadcopters are underactuated systems. They have 6 degrees of freedom in motion (3 translational and 3 rotational) while having only 4 control inputs (the speeds of each motor) which makes it very complex to obtain stabilization in the system. Consequentially, the Simulink models were used to design the most appropriate controller that provided for stable flight when the quadcopter was following its established flight path composed of the desired x, y, z positions and heading.

Designing the controller through Simulink enabled the quadcopter to safely move to the desired position while maintaining stable roll, pitch, and yaw angles. Another factor that was considered while designing the controller on the Simulink model was the function of the ESCs. These components are used to control the angular velocities on each rotor. As a result, the following direct control inputs were derived, as shown in the Mathematical Justification section, and implemented in Simulink [24]:

- U_1 : the resulting thrust of the four rotors
- U_2 : the difference of thrust between the motors on the x axis which results in roll angle changes and subsequent movement in the lateral x direction.
- U_3 : the difference of thrust between the motors on the y axis which results in pitch angle changes and subsequent movement in the lateral y direction.
- U_4 : the difference of torque between the clockwise and counterclockwise rotors which results in a moment that rotates the quadrotor around the vertical z axis.

The equations for the net thrust and moments facilitated the process of solving for the desired motor speeds to be sent to the ESCs according to the control inputs and this was applied in each of the studied Simulink models. The models that were studied for the development of the UAV controller may be divided into two categories: reference models and prototype models. The reference models were those which were found from previous projects on quadcopter design and were modified according to the designed quadcopter's physical parameters. Once modified, the reference models were used to create a Simulink model successfully simulated UAV flight and design the control system. On the other hand, the prototype models were those which were developed to be uploaded into the flight controller hardware while implementing the control system that was designed through the reference models.

Reference Simulink Models

1. Quadrotor dynamics modelling using Simulink (Quadrotorsim) [34]

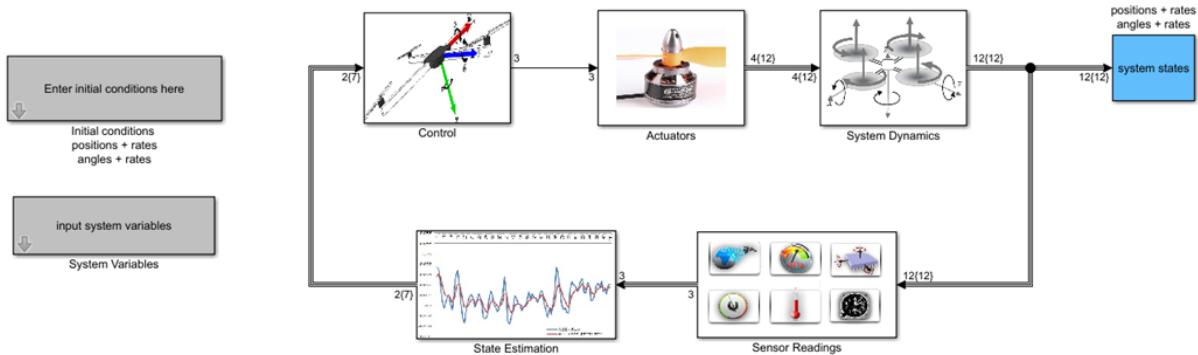


Figure 6E.1: Quadrotorsim Simulink model

The Quadrotorsim Simulink model represents the translational and rotational dynamics of a quadrotor. This model was modified and was used for: system feasibility studies, system performance assessment and trade-offs, and control law performance evaluation. The following processes were modelled:

- Quadrotor dynamics
- Motor dynamics
- Kalman filter for state estimation
- Simple sensor model/ Analog to Digital conversion (ADC)

2. Quadcopter Project (asbQuadcopter) [35]

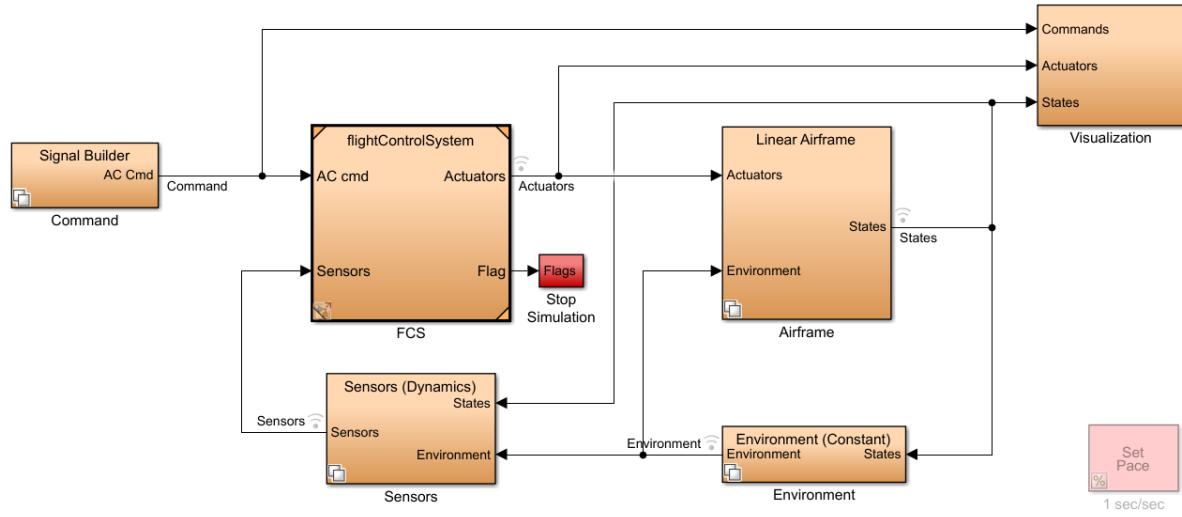


Figure 6E.2: asbQuadcopter Simulink model

The asbQuadcopter Simulink model implemented a flight control system that made use of a complementary filter to estimate attitude and Kalman filters to estimate position and velocity. The flight control system in this model was composed of three control sub-systems: a PID controller for pitch/roll control, a PD controller for yaw control, and a PD controller for position control in the North-East-Down coordinates. This Simulink model provided for the implementation of several combinations of estimators and controllers to be evaluated for the final Simulink model design. On the other hand, the inputs (desired pitch, roll, yaw, North (X), East (Y), Down (Z) coordinates) for this model were adjusted through the VSS_COMMAND variable in the Simulink workspace. Additionally, there were different ways in which the values for these inputs could be received (e.g. signal editor block, joystick, previously saved data, or spreadsheet data).

3. PX4 Autopilots Support from Embedded Coder (px4demo_attitude_system) [36]

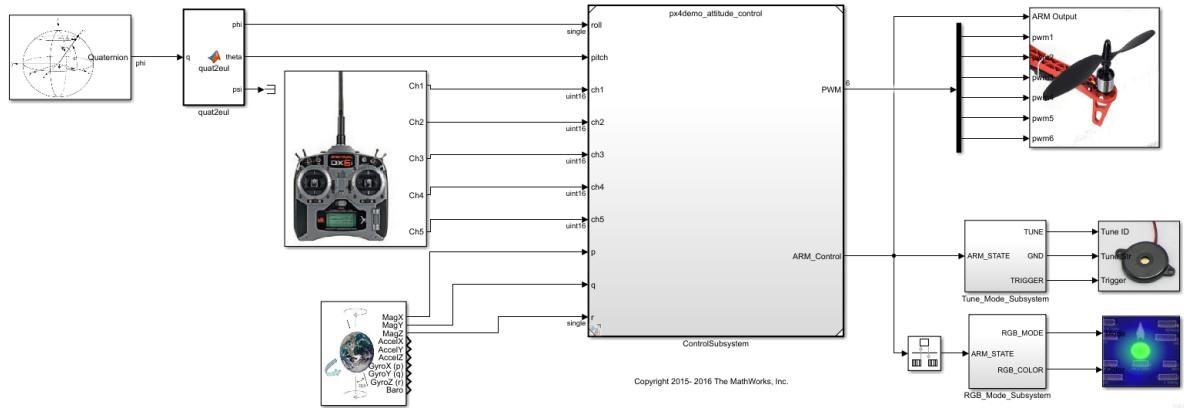


Figure 6E.3a: px4demo_attitude_system Simulink model

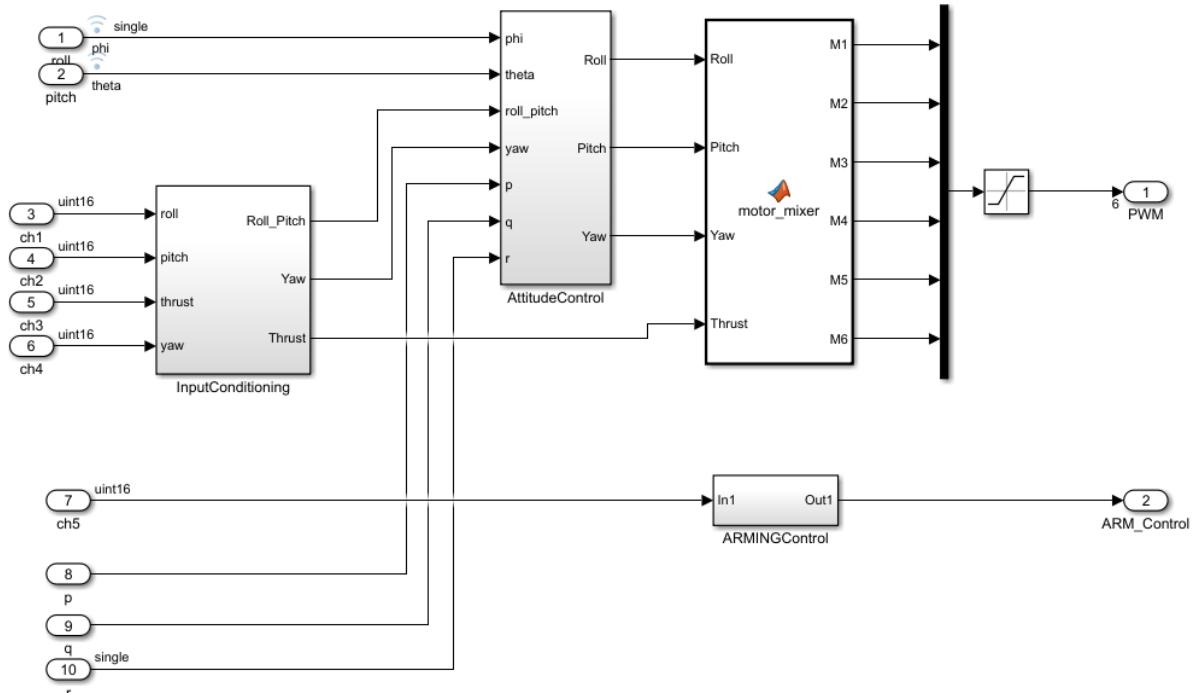


Figure 6E.3b: px4demo_attitude_system model Control subsystem

The PX4 Autopilots Support from Embedded Coder model implemented both Simulink and MATLAB's Embedded Coder to automatically build and deploy flight control algorithms to the Pixhawk flight controller hardware. This model made use of Simulink PID blocks for roll, pitch, yaw and attitude control.

4. PX4 Development Kit for Simulink (multi_model) [37]

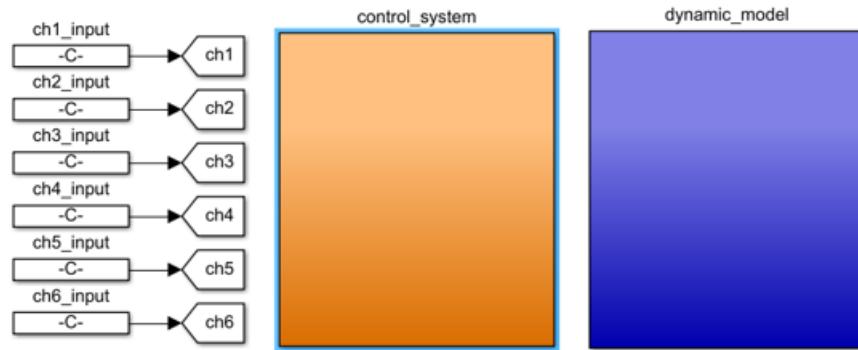


Figure 6E.4a: multi_model Simulink model

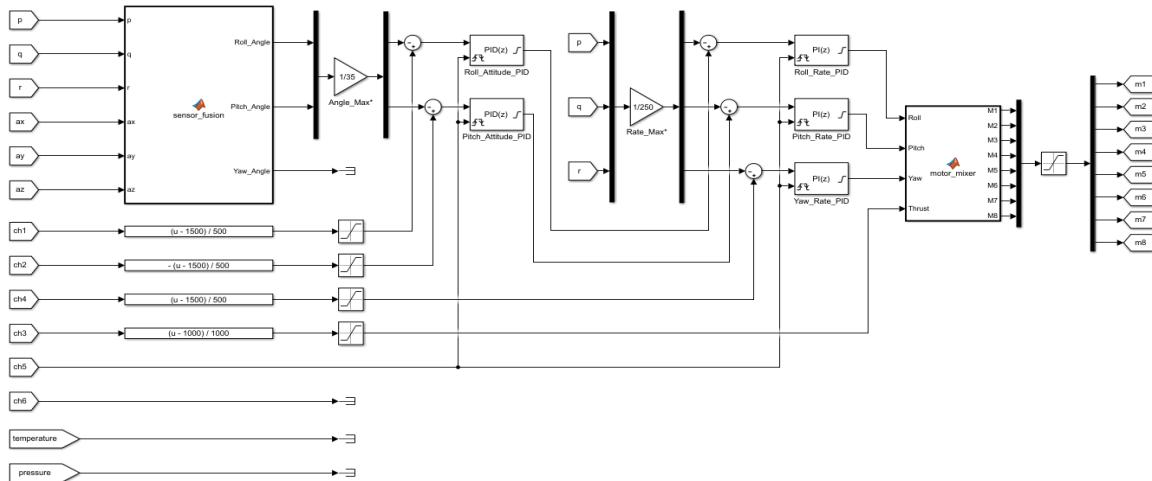


Figure 6E.4b: multi_model Simulink model Control subsystem

The multi_model Simulink model was a configurable dynamic model that contained a wide range of multi-copter configurations, as well as a more complex position control system that implemented a Kalman filter for navigation estimates with velocity updates that were provided by a downward facing on-board camera. This model also made use of Simulink PID blocks for roll, pitch, yaw and attitude control.

5. Robotics Toolbox for Matlab (sl_quadrotor) [38]

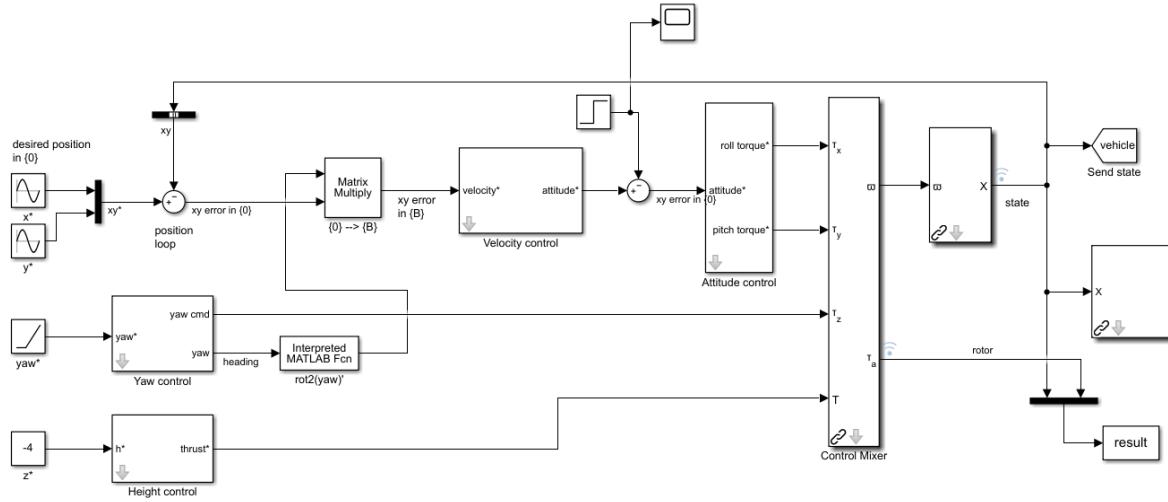


Figure 6E.5: sl_quadrotor Simulink model

The sl_quadrotor Simulink model was designed by Peter Corke and its main function was to simulate a simple quadrotor with thrust axes pointing upwards. This model had a script that created the workspace variables which described the dynamic characteristics of a quadrotor in motion and contained a dynamics block that represented the equations of motion that apply to a quadrotor.

On the other hand, the control system of the model involved multiple nested control loops that computed the required thrust and torques so that the vehicle moved to the a specific setpoint. The attitude controller was part of the innermost loop in the control system. The inputs for the attitude controller were the current and desired roll and pitch angles and their rates to provide damping. The xy-position controller was part of the outer loop in the control system and worked by processing the changes in the roll and pitch angles so as to provide a component of thrust in the direction of desired xy-plane motion. Finally, the altitude was controlled by a proportional-derivative controller that determined the average rotor speed necessary to generate a thrust equal to the weight of the vehicle. This provided for an example of feedforward control which was, in this case, used to counter the effect of gravity that is otherwise a constant disturbance to the altitude control loop. The outputs of the previously mentioned controllers were then combined through the mixer block which also enforced limits on rotor speeds. Once the rotor speeds were identified, the running simulation of the model showed the vehicle taking off and flying in a circle on the flight simulation plot.

6. OS4 (systema) [39]

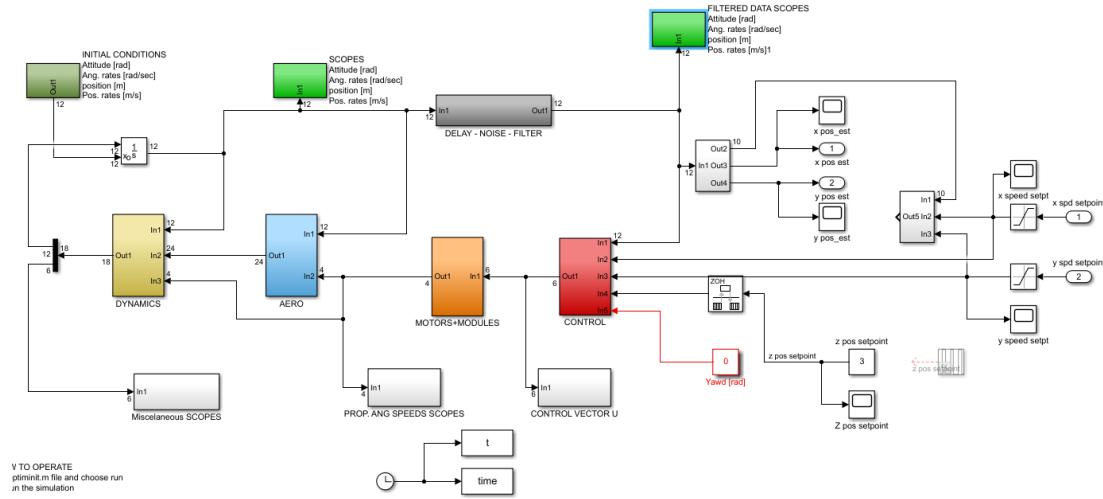


Figure 6E.6a: Systema Simulink model

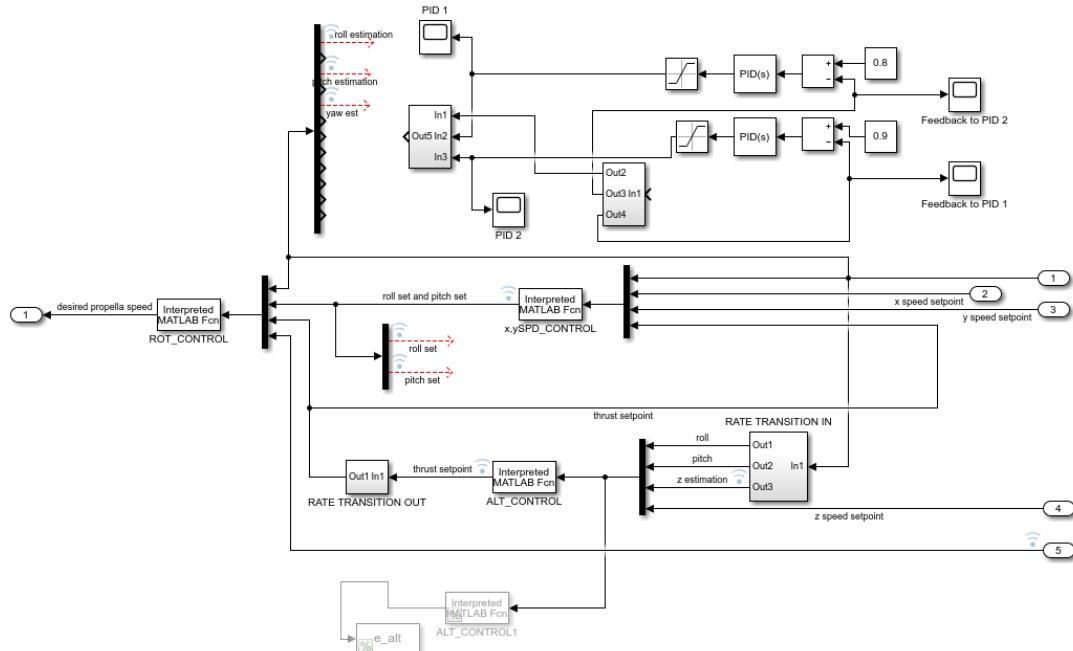


Figure 6E.6b: Systema Simulink model control subsystem

The OS4 Simulink model was a dynamic model that was designed using Newton-Euler formalism, DC motor equations, model identification, blade element and momentum theories. Tait-Bryan angles were used for the parametrization. This model implemented controllers for autonomous take-off and landing. Additionally, this model made the following assumptions: the quadcopter frame is rigid and symmetrical; the CoG and the body fixed frame origin coincide; the propellers are rigid; and the thrust and drag are proportional to the square of the propeller's speed. This model used Simulink PID blocks for roll, pitch, yaw and altitude control.

7. Quadcopter Dynamic Modeling and Simulation (Quad-Sim) v1.00 [40]

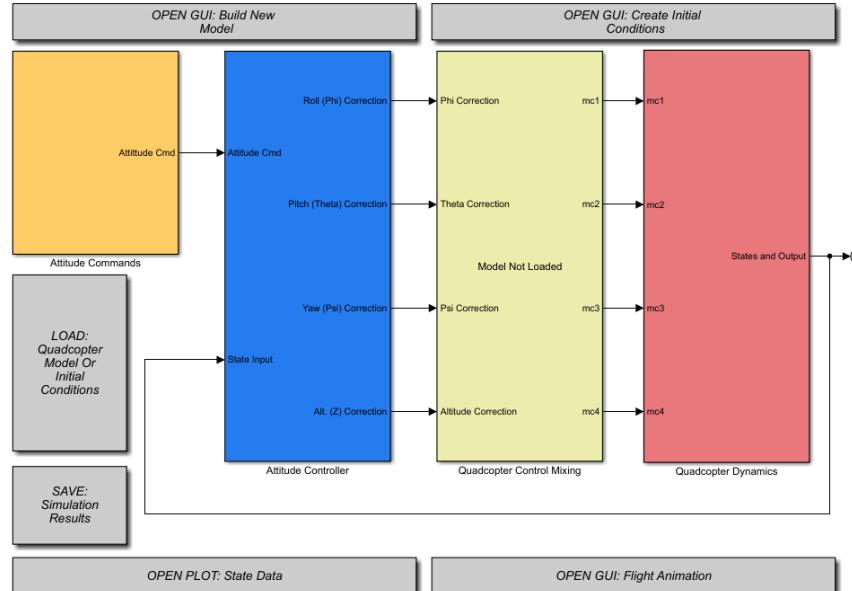


Figure 6E.7a: Quad-Sim Simulink model

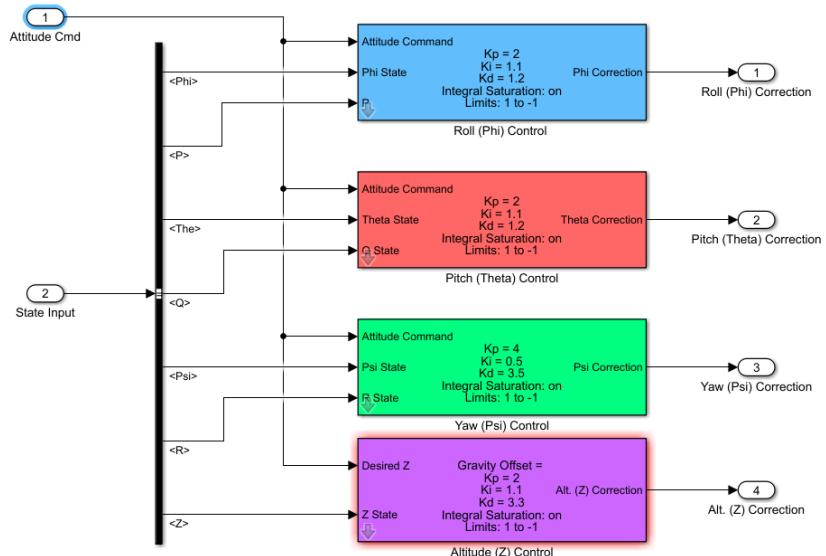


Figure 6E.7b: Quad-Sim Simulink model attitude control subsystem

The Quad-Sim Simulink model was designed to assist users in the process of modeling and simulating quadcopter vehicle behavior and designing the most appropriate control system for the modeled vehicle. This model used Simulink PID blocks for roll, pitch, yaw and altitude control. Additionally, code in the documentation for this model included a test rig that facilitated component performance measurement, several MATLAB data analysis tools and GUIs.

8. MatlabQuadSimAP (QuadrotorSimulink) [24]

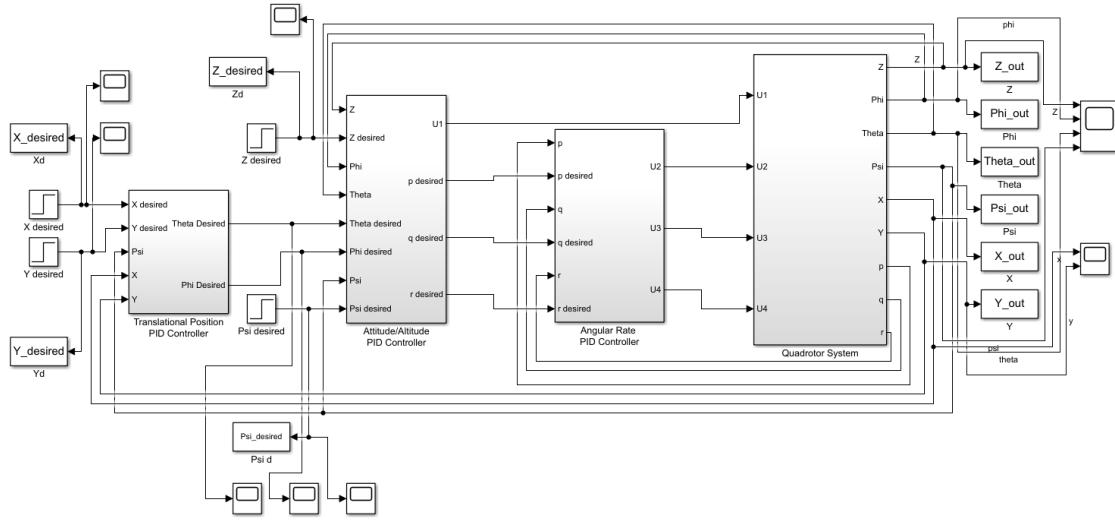


Figure 6E.8a: QuadrotorSimulink model

The QuadrotorSimulink model was designed by Will Selby using MATLAB/Simulink to simulate a 3DRobotics ArduPilot based quadrotor. This model received the quadrotor motor speeds as inputs to the quadrotor equations of motion block and the output was the corresponding simulated RPM of the propeller. The angular velocity and translational acceleration were also inputs to the model and could be received as sensor measurements or pre-defined values. These values were compared to the desired attitude and positions to calculate the error signal that was fed into the control systems. Then, the first subsystem of the model received the quadrotor position error as input and outputted the desired roll and pitch angles. The next subsystem, which was the attitude subsystem, received the roll, pitch, and yaw errors as input. The attitude subsystem was composed of explicit PIDs (not Simulink blocks). This attitude subsystem outputted the desired angular velocities for each axis which were fed into the final control system. The outputs of this system were the control inputs that could then be mapped into desired motor speeds.

Prototype Models

1. Quadrotor Dynamics model

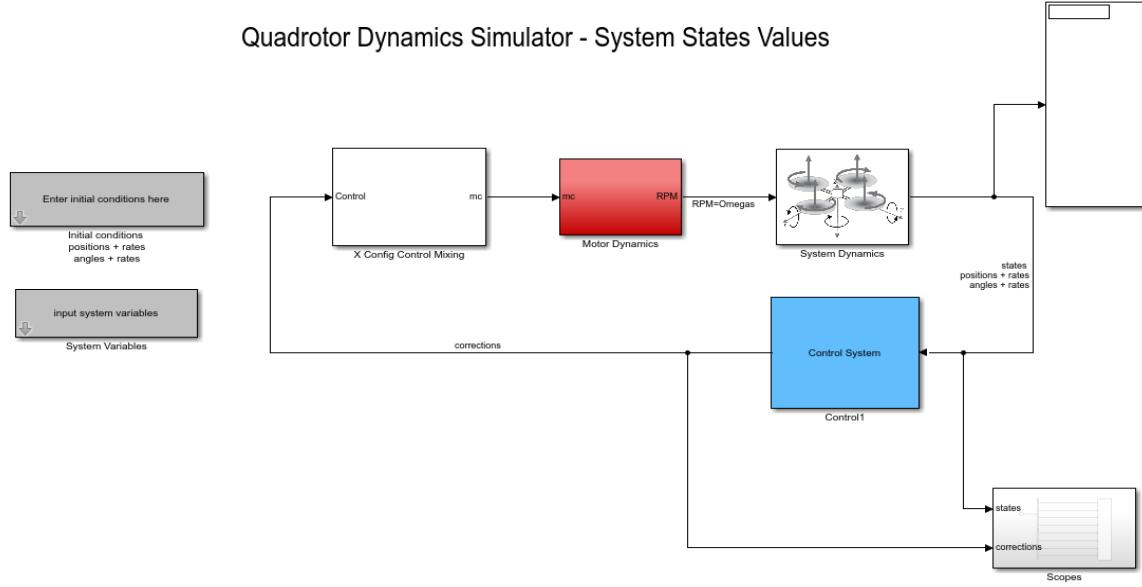


Figure 6E.9: Quadrotor Dynamic model

This Quadrotor Dynamic Simulink model was designed to evaluate the designed quadrotor system and the initial parameters shown in Table 6E.2. The model used Simulink PID blocks for roll, pitch, yaw and altitude control and the outputs from these PID blocks were mapped into desired motor speeds. This model also implemented a system dynamics block that represented the translational and rotational equations of motions for the designed quadrotor.

2. Quadrotor Pixhawk model

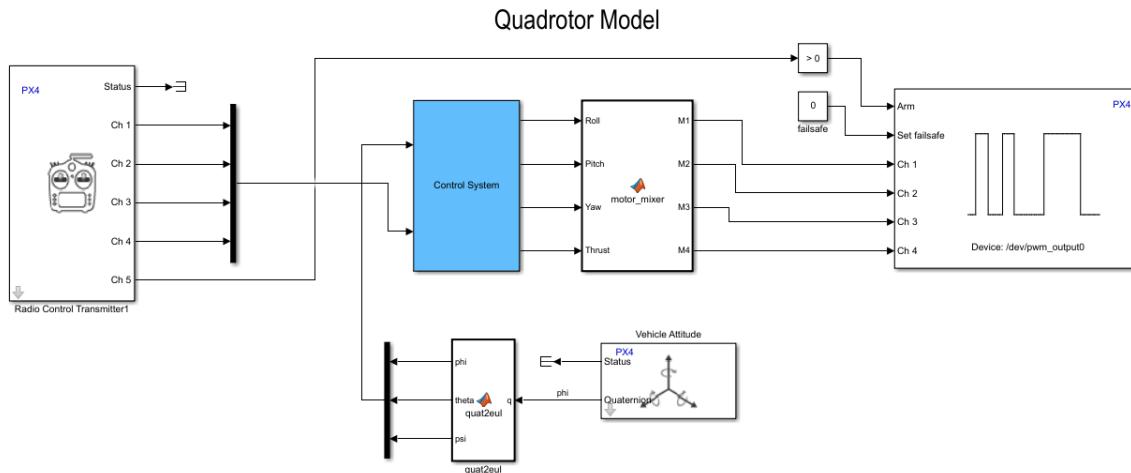


Figure 6E.10a: Quadrotor Model for Pixhawk

Using the Embedded Coder Support Package for PX4 Autopilots allowed for the generation of C++ code that is specifically tailored for the Pixhawk flight controller hardware. This code

was generated from the Simulink model shown in Figure 6E.10a. In this model, the control system block consisted of classic PID controllers for roll, pitch and yaw angles. The output data was then used to calculate the RPS of the motors. On the other hand, the same model was adapted for Ardupilot as shown in Figure 6E10b. Nevertheless, it was not possible to do further testing with this model due to complication while trying to upload the generated code to from Simulink to the flight controller hardware.

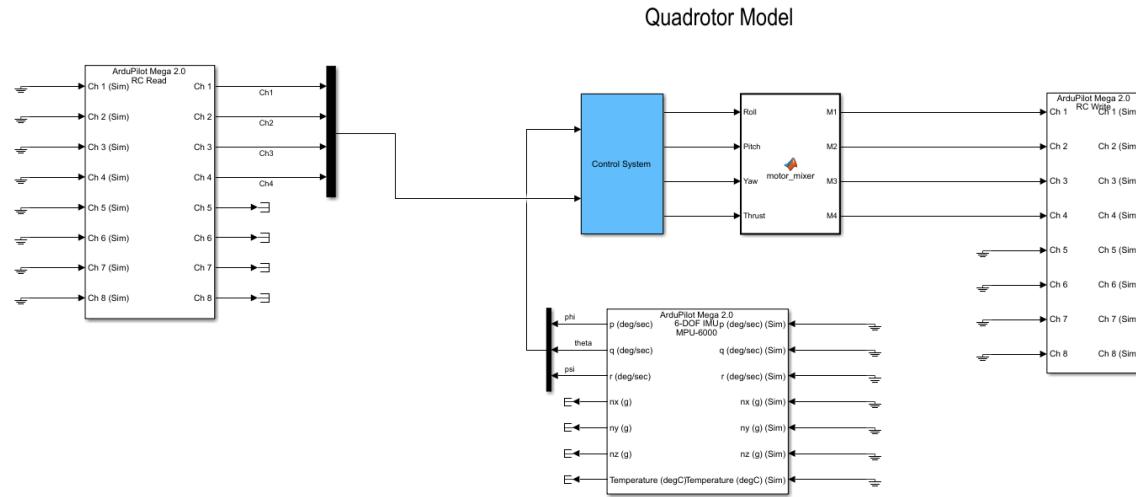


Figure 6E.10b: Quadrotor Model for Ardupilot

From another point of view, using the reference model shown in Figure 6E.11 a new version of the Simulink model was devised as shown in Figure 6E.11. This version was modified to include a trajectory plot and was used for testing purposes.

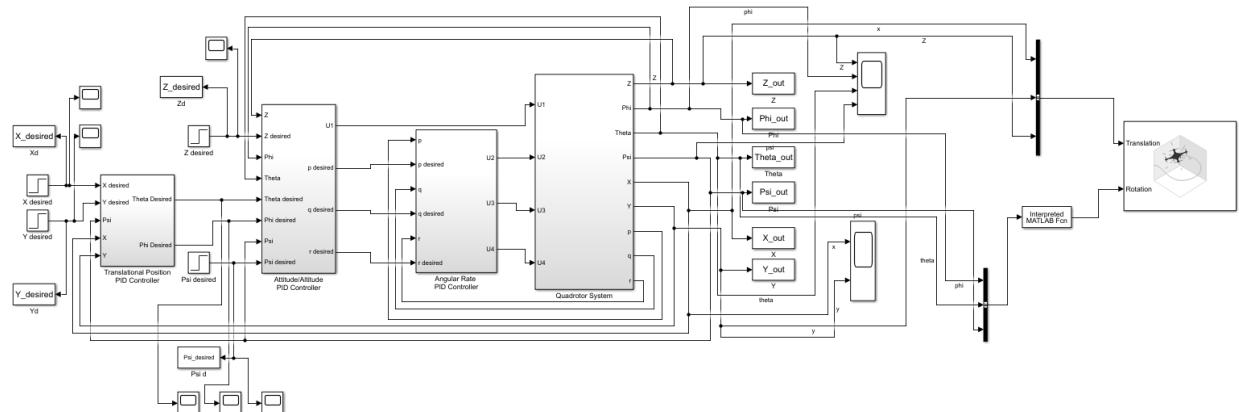


Figure 6E.11: Modified QuadrotorSimulink model

F. Economic Analysis and Budget

The comparison of the current cost performance against the planned amount of financed resources stated in the project budget resulted in:

Table 6F.1: Allocated budget for preliminary project design and development

Item	Quantity	Group	Honeywell	P.U.P.R.	Total
Initial funding	1		\$ 2,000.00		\$ 2,000.00
Professional Services	2	\$ 12,000.00			\$ 24,000.00
Transportation (provided)	1			\$ 0.00	\$ 0.00
Gyroscope, Accelerometer & GNSS	1		\$ 20,000.00		\$ 20,000.00
RF Transceiver	2			\$ 50.00	\$ 100.00
Motors	8			\$ 30.00	\$ 240.00
Electronic Speed Controller	8			\$ 18.00	\$ 144.00
Arduino Microcontroller	1			\$ 22.00	\$ 22.00
Power Supply (LiPo Battery)	1			\$ 120.00	\$ 120.00
Power Distribution Board	1			\$ 30.00	\$ 30.00
Project Plan software license	1	\$ 79.00			\$ 79.00
Shipping cost					\$ 150.00
Taxes					\$ 0.00
Total		\$ 12,079.00	\$ 22,000.00	\$ 270.00	\$ 46,885.00

Table 6F.2: Adjusted Budget for final project design and development

Item	Quantity	Group	Honeywell	P.U.P.R.	Total
Initial funding	1		\$ 2,000.00		\$ 2,000.00
Professional Services	2	\$ 12,000.00			\$ 24,000.00
Transportation (provided)	1			\$ 0.00	\$ 0.00
XT-XNITE GPS (donated)	1			\$ 0.00	\$ 0.00
Futaba 6J 6-Channel S-FHSS System	1			\$ 193.99	\$ 193.99
Cobra Motors CM-2213-26	4			\$ 30.99	\$ 187.96
Cobra 20A Multirotor ESC W/LBEC	4			\$ 21.99	\$ 87.96
APM 2.6 ArduPilot Mega 2.6 Flight Cont	1			\$ 0.00	\$ 0.00
Wild Scorpion LiPo Battery 4S 14.8V 42	3			\$ 103.66	\$ 310.98
Pixhawk 4 Power Module	1			\$ 33.00	\$ 33.00
Project Plan software license	1	\$ 79.00			\$ 79.00
Shipping cost					\$ 150.00
Taxes					\$ 0.00
Total		\$ 12,079.00	\$ 2,000.00	\$ 383.63	\$ 27,042.89

The ending budget was \$27,042.89 vs \$46,885.00 representing a substantial reduction over the initially proposed budget for project design and development.

Chapter 7: Project Results

The results from this project may be divided into three main categories: results from software implementation, results from hardware implementation, and the final version of the designed and manufactured UAV.

A. Software

In terms of software, two programs were used: MATLAB/Simulink and Mission Planner.

MATLAB/Simulink implementation

MATLAB/Simulink was used for the design of the UAV's control system. Various Simulink models were created or adjusted from previous references to evaluate our quadrotor system and its initial parameters as shown in the System Specifications section. The implemented models had Simulink PID blocks for roll, pitch, yaw and altitude control. Then, the control system outputs were mapped into desired motor speeds according to the appropriate translation and rotation equations of motions.

From the evaluation models, a Simulink model containing the desired PID parameters to be uploaded to the Cube flight controller was devised as shown in Figure 7A.1. However, the upload was unsuccessful due to the fact that there were some communication conflicts in the buses which impaired the code upload process shown in Figure 7A.2. Thus, the team opted to adjust the control system from the Mission Planner ground control station.

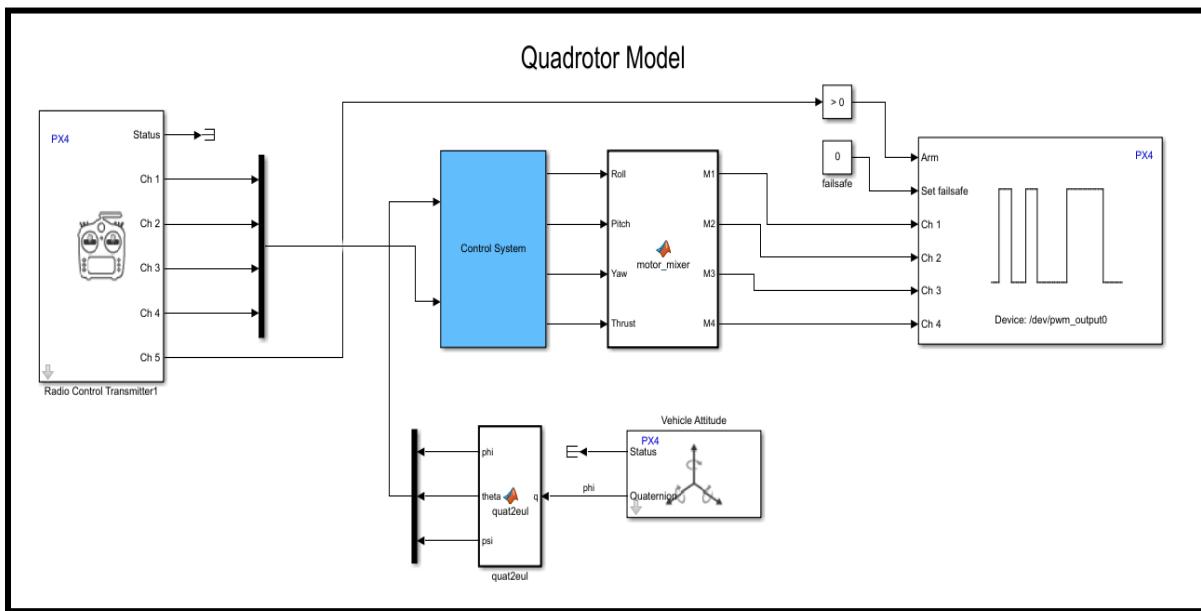


Figure 7A.1: Simulink model for Pixhawk 2

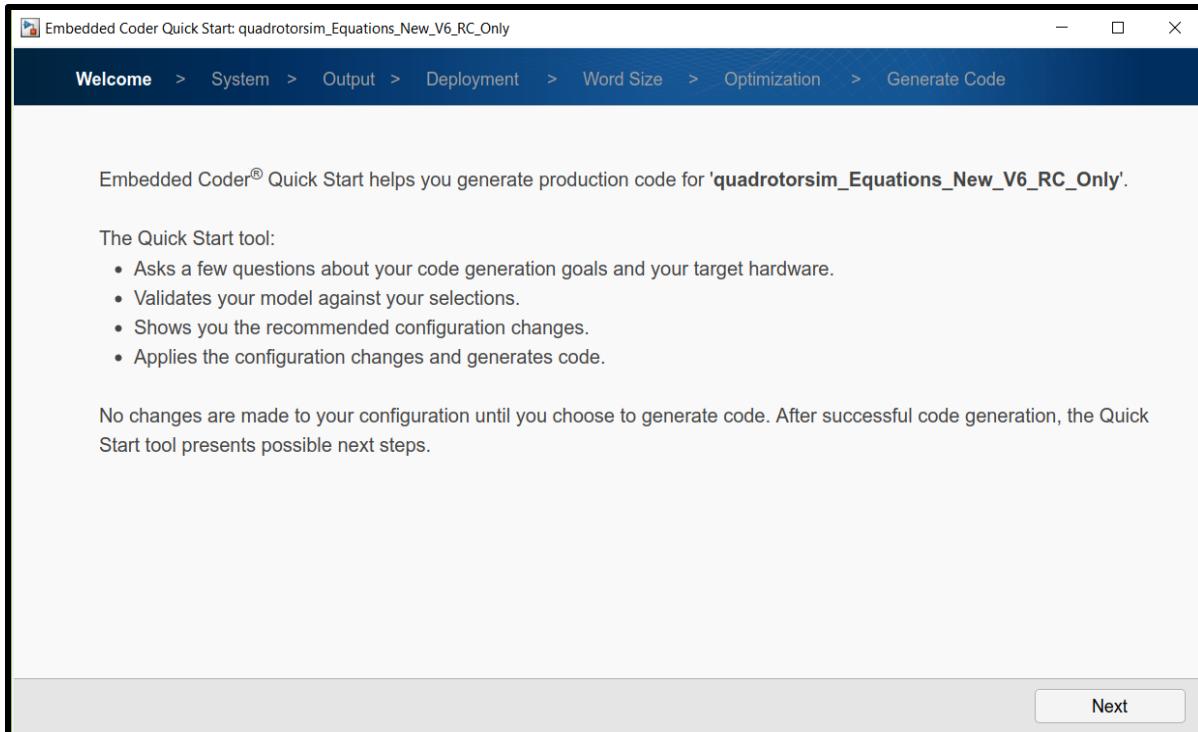


Figure 7A.2: Generating code for Pixhawk 2 from Simulink model

PID parameters in Mission Planner

As previously mentioned, due to the fact that the code from the Simulink model could not be uploaded to the Pixhawk flight controller, the UAV's PID gains were successfully adjusted through the implementation of Mission Planner and its Extended Tuning section as shown in Figure 7A.3. As a result, successful flight was achieved.

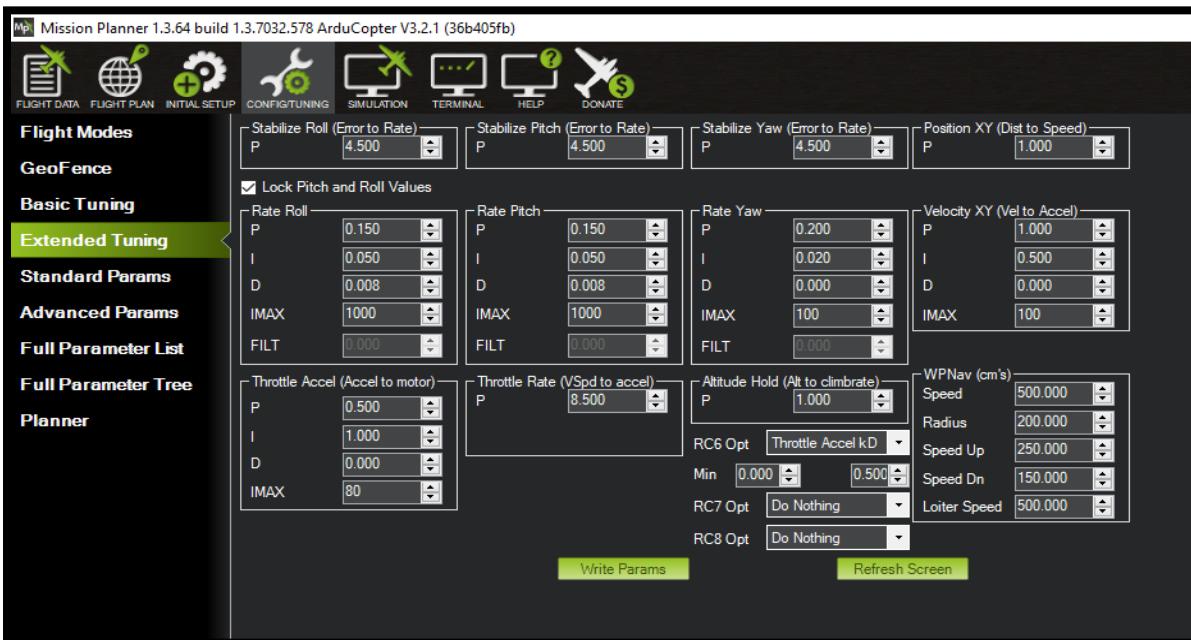


Figure 7A.3: Mission Planner Extended Tuning window

B. Hardware

In terms of hardware, the quality of the designed was revised. First the rigidness of the arms shown in Figure 7B.1 was observed. During initial fly tests the arms started to bend upward due to the force created by the rotation of the propellers. It was concluded that the Nylon G material used for the printing was not strong enough. Therefore, all the arms were re-printed with PLA, a stronger material thus providing a better rigid body.

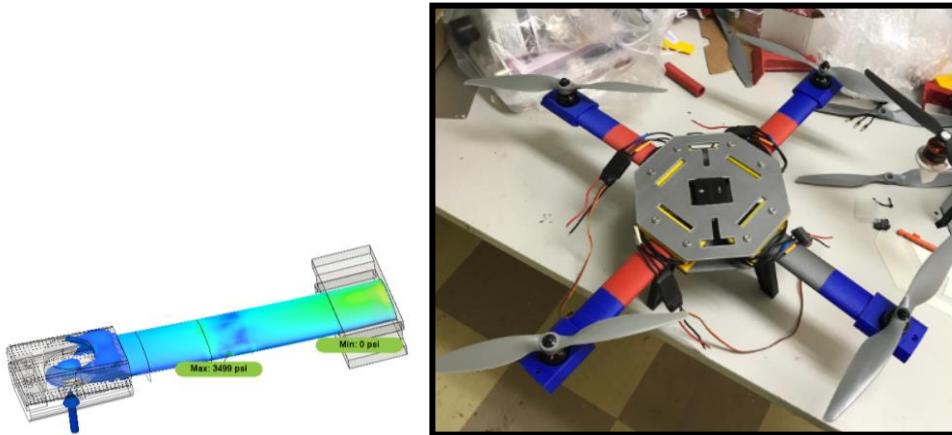


Figure 7B.1: Quadcopter arm

From another point of view, the UAV's propellers guards resulted in need of redesign as well due to their tendency to impact the propeller during flight which caused flight instability. Two versions of the propeller guards were printed as shown in Figure 7B.2. The printing for version#1 took more than 5 hours to finish. Then, to significantly reduce printing time, version #2 was designed to consist of two separated pieces ("trident" and arch) that were glued together. The material used was the same as the arms to obtain more rigidness. However, while in flight the arch of the propeller guard got loose which caused an impact to the propeller and made the quadcopter fall as shown in Figure 7B.3.

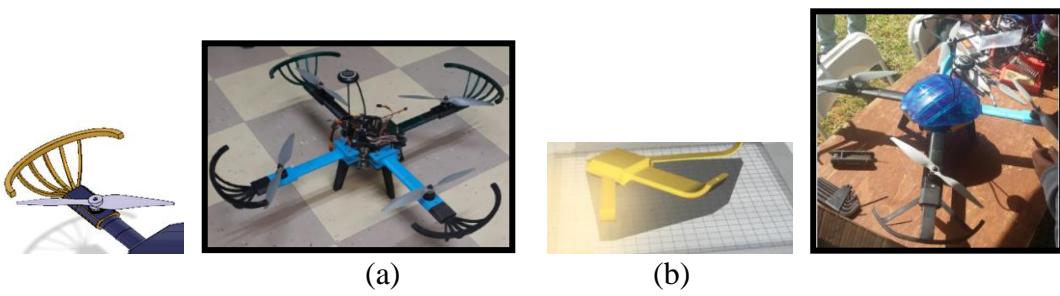


Figure 7B.2: Propeller guard (a) version#1 (b) version#2



Figure 7B.5: Quadcopter crash due to faulty propeller guard

Another part of the quadcopter that needed to be re-printed was the top and bottom bases shown in Figure 7B.6. The material used was Lexan and during tests, this material was observed to shatter easily.

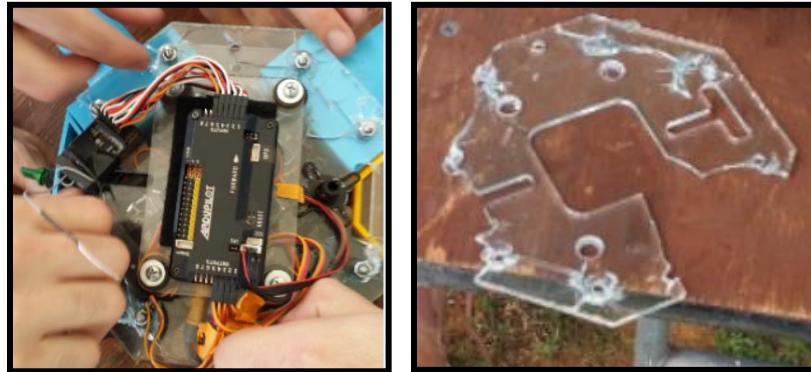


Figure 7B.6: Quadcopter top/bottom bases shattered

C. Final Design

After mentioning the results from software and hardware implementation, the following figure (Figure 7C.1) may be shown to represent the final version of the designed and manufactured UAV.



Figure 7C.1: Designed and manufactured UAV (final version)

Chapter 8: Conclusions

This project offered various opportunities for the professional development of students pursuing a career in engineering. One of these opportunities was to participate in a multidisciplinary team composed of electrical, mechanical, and computer engineers. As a result, the team acquired the ability to obtain the holistic view of a project and how the different disciplines are integrated to solve the different problems that emerge while executing the necessary tasks to comply with project requirements.

Another important aspect to this project was the fact that the sponsor, Honeywell required that NASA's system engineering guide be used to demonstrate project progress. For this reason, the following presentations were developed: Preliminary Design Review, Critical Design Review, and Flight Readiness Review. These presentations demonstrated how the systems engineering process was applied throughout the project and are included in the appendix. On the other hand, going through the process of research and development as was required by NASA's system engineering guide, the team also had the opportunity to broaden their knowledge concerning the application of engineering concepts in the design of a UAV and learn how to manage the unforeseen issues that may emerge during the design process. As a result, students learned the importance of developing a baseline to handle possible deviations in a project.

Through this project a quadcopter flight simulation was successfully executed through the implementation of MATLAB/Simulink and Mission Planner which facilitated the design of a UAV that has both remote and autonomous features. As a recommendation for future projects on UAV design, it is stated that the implementation of Simulink with Pixhawk requires further investigation. Precaution is advised when uploading firmware to telemetry modules since communication problems between the modules may occur if the wrong firmware is installed. Additionally, an alternative to UAV design concerning autonomy is presented by suggesting the use of the Ardupilot flight controller along with the Arduino IDE. From another point of view, participation in this project gave PUPR the opportunity to achieve a better understanding of the sponsor for future UAV design competitions that they may host. Finally, this project opened the doors to future projects related to the topic of UAV design by serving as a reference.

Chapter 9: Bibliography & Reference Citation

- [1] B. Clough, "Metrics, Schmetrics! How The Heck Do You Determine A UAV's Autonomy Anyway?", Dtic.mil, 2002. [Online]. Available: <http://www.dtic.mil/dtic/tr/fulltext/u2/a515926.pdf>.
- [2] D. Kotarski, M. Krznar, P. Piljek and N. Simunic, "Experimental Identification and Characterization of Multirotor UAV Propulsion", *Journal of Physics: Conference Series*, vol. 870, p. 012003, 2017. Available: 10.1088/1742-6596/870/1/012003.
- [3] O. Liang, "Types of Multirotor", *Oscar Liang*, 2016. [Online]. Available: <https://oscarliang.com/types-of-multicopter/>.
- [4] W. Fum, "IMPLEMENTATION OF SIMULINK CONTROLLER DESIGN ON IRIS+ QUADROTOR", *SemanticScholar*, 2015. [Online]. Available: <https://pdfs.semanticscholar.org/19b1/aafc474cbcd61b27d33475d05f7a9c6344e.pdf>.
- [5] K. Alexis, "LQR Control", *Dr. Kostas Alexis*. [Online]. Available: <http://www.kostasalexis.com/lqr-control.html>. [Accessed: 03- Feb- 2020].
- [6] F. Aula, "Robust Control: LQR, Kalman Filter, and LQG", University of Salahaddin, 2015. [Online]. Available: http://fadhil.yolasite.com/resources/Robust_Control/RC%2008.pdf
- [7] M. Shaik, "Quadrocopter Fuzzy Flight Controller", *Oru.diva-portal.org*, 2011. [Online]. Available: <http://oru.diva-portal.org/smash/get/diva2:567121/FULLTEXT01.pdf>.
- [8] R. Bappy, A. Rafi, S. Islam, A. Sajjad and K. Imran, "Design and Development of Unmanned Aerial Vehicle (Drone) for Civil Applications", Undergraduate, BRAC University, 2015.
- [9] Dronecode. "Telemetry Wifi · PX4 v1.9.0 User Guide", *Docs.px4.io*, 2019. [Online]. Available: https://docs.px4.io/v1.9.0/en/telemetry/telemetry_wifi.html.
- [10] A. Jacobs, "How to choose the right size motors & ESCs for your Drone, quadcopter, or Multirotor build - Quad Questions", *Quad Questions*. [Online]. Available: <https://quadquestions.com/blog/2017/02/22/choose-right-size-motors-drone/>.
- [11] Cobra motors. Image. [Online]. Available: <http://www.innov8tivedesigns.com/images/specs/Cobra-CM-2213-26-Specs.htm>
- [12] M. A. Sattar and A. Ismail, Dr., "PID Control of a Quadrotor UAV", *International Research Journal of Engineering and Technology (IRJET)*, Volume: 04 Issue: 08, Aug. 2017.
- [14] Quimby, P. (2019). Why is 'x' Configuration Preferred Over + Config of Quadcopter – Quora. [online] Quora.com. Available at: <https://www.quora.com/Why-is-x-configuration-preferred-over-+-config-of-quadcopter>

- [15] F. Sabatino, “Quadrotor Control: Modeling, Nonlinear Control Design, and Simulation”, M. S. thesis, Dept. Elect. Eng., KTH Royal Institute of Technology, Stockholm, Sweden, 2015
- [16] Tytler, C. (2017). “Modeling Vehicle Dynamics - Euler Angles”. [online] Autonomy in Motion. Available at: <http://charlestytler.com/category/quadcopter/>
- [17] Emissarydrones.com. (2017). “What is Pitch, Roll and Yaw ?”. [online] Emissary Drones. Available at: <https://emissarydrones.com/what-is-roll-pitch-and-yaw>
- [18] Rose, D. (2015). “Rotations in Three-Dimensions: Euler Angles and Rotation Matrices”, [Online] Engineering Notes. Available at: http://danceswithcode.net/engineeringnotes/rotations_in_3d/rotations_in_3d_part1.html
- [19] R. Stevens, “Quadcopter Axes and Motions”. [Online]. Available: <https://www.flickr.com/photos/ricstephens/16989545847>
- [20] E. Gopalakrishnan, “Quadcopter Flight Mechanics Model and Control Algorithms”, M. S. thesis, Dept. Control Eng., Czech Technical University, Prague, Czech Rep, 2016
- [21] T. Bresciani, “Modelling, identification and control of a quadrotor helicopter,” M.S. thesis, Dept. Automatic Control, Lund Univ., Lund, Sweden, 2008
- [22] H. M. Nabil ElKholy, “Dynamic Modeling and Control of a Quadrotor Using Linear and Nonlinear Approaches”, M.S. thesis, School of Sciences and Engineering, The American University in Cairo, El Cairo, Egypt, 2014
- [23] R. Beard, Class Lecture, Topic: “Quadrotor Dynamics and Control.” Dept. Electrical & Computer Engineering, Brigham Young University, Provo, UT, Oct 3, 2008
- [24] W. Selby, “Quadrotor System Modeling - Non-linear Equations of Motion”, Wil Selby [Online]. Available: <https://www.wilselby.com/research/arducopter/modeling/>
- [25] M. Ben-Ari, “A Tutorial on Euler Angles and Quaternions”, Weizmann Institute of Science, 2014. [Online]. Available: <http://www.weizmann.ac.il/sci-tea/benari/sites/sci-tea.benari/files/uploads/softwareAndLearningMaterials/quaternion-tutorial-2-0-1.pdf>
- [26] “How is orientation in space represented with Euler angles?”, Mecademics, n.d. [Online]. Available: <https://mecademic.com/resources/Euler-angles/Euler-angles.html>
- [27] J. Barbic. CSCI 520. Class Lecture, Topic: “Quaternions and Rotations”. Department of Computer Science, University of Southern California, Southern California, CA, Feb 15, 2012.
- [28] “Quaternion Math Tutorial”, Sacredsoftware, n.d. [Online].

Available: <http://sacredsoftware.net/tutorials/quaternion.html>

- [29] M. Timmons, “Quadrotor Dynamic Model: Propeller Gyroscopic Effect”, MTwallets July 27, 2018. [Online]. Available: <https://www.mtwallets.com/quadrotor-dynamic-model-propeller-gyroscopic-effect/>
- [30] “Quad Star Drones, Part 5: Drag”, Quad Star Drones, n.d. [Online]. Available: <https://quadstardrones.com/part-5-drag/>
- [31] G. Regula, “Formation Control of Autonomous Aerial Vehicles”, Ph.D. thesis, Dept. of Control Eng. and Inf. Tech., Budapest University of Technology and Economics, Budapest, Hungary, 2013
- [32] A. R. Partovi, “Development of a cross style quadrotor,” Ph.D. thesis, Dept. of Electrical and Computer Engineering, National University of Singapore, Singapore, 2012
- [33] S. Praharaj, T. Nanda, and K. Rana, “Power And Other Requirements For Unmanned Aerial Vehicles (UAV)”, presented at 2016 Southwest Decision Sciences Institute Conference, Oklahoma City, OK, 2016
- [34] “Quadrotor dynamics modelling using Simulink”. [Online]. Available: <https://www.mathworks.com/matlabcentral/fileexchange/42666-quadrotor-dynamics-modelling-using-simulink>
- [35] “Quadcopter Project”. [Online]. Available: <https://www.mathworks.com/help/aeroblks/quadcopter-project.html>
- [36] “PX4 Autopilots Support from Embedded Coder”. [Online]. Available: <https://www.mathworks.com/hardware-support/px4-autopilots.html>
- [37] A. Polak, “PX4 Development Kit for Simulink”. [Online]. Available: <http://polakiumengineering.com/px4-development-kit-for-simulink/>
- [38] P. Corke, “Robotics Toolbox for Matlab”. [Online]. Available: <http://petercorke.com/wordpress/toolboxes/robotics-toolbox>
- [39] “OS4-vertical-square-2017a”. [Online]. Available: <https://github.com/cnpscshangbo/OS4-vertical-square-2017a/blob/master/MUDAGRAF.m>
- [40] “Quad-Sim” . [Online]. Available: <https://github.com/dch33/Quad-Sim>

Chapter 10: Magazine Article

Honeywell Puerto Rico Navigation Challenge

2018-2019

Mara De Leon (Author)
Polytechnic University of Puerto Rico
de_79250@students.pupr.edu

Amaris Vélez (Author)
Polytechnic University of Puerto Rico
velez_90013@students.pupr.edu

ABSTRACT – This paper is the product of a capstone project that was originally proposed to complement the multidisciplinary team that was created as a result of Honeywell's First Annual Navigation Challenge where different universities from Puerto Rico were given the task of building an unmanned aerial vehicle (UAV) from scratch. The UAV team required people from the following disciplines: Mechanical Engineering, Computer Engineering, and Electrical Engineering. This paper has been written by the teammates from the electrical engineering team. The main task for the electrical engineering team was the design of a UAV control system to ensure a successful flight. The software that was chosen to design the control system was MATLAB/Simulink. Through this software a control system for the UAV was successfully designed using the PID control technique. Thus, various Simulink models were implemented to simulate UAV flight. From another point of view, although the control logic could not be uploaded to the UAV's flight control hardware, the PID parameters for the UAV were successfully specified to the flight control hardware through the ground control station software (GCS) called Mission Planner which facilitated both manual and autonomous flight for the UAV. This paper shows the complete design process for the UAV.

Keywords: Unmanned aerial vehicle, control system, and flight control hardware

INTRODUCTION

A UAV or drone can be defined as an aircraft that does not carry a human pilot or passenger and is fully or partially autonomous. The term unmanned aerial system (UAS) is used when the selection of the ground control station and communications unit is included in the design. Thus, the term UAS includes the whole system involved when operating a UAV. Thus, for this project, a UAS will be designed since the vehicle must be developed from scratch along with an interface to interact with it and a communication scheme to facilitate this interaction.

To design a UAV there are some basic factors to take into consideration such as the weight, lift, drag, and thrust. According to these parameters, elements such as frame, motors, propellers, and batteries are identified, as well as some electrical components such as the flight controller and the electronic speed controllers (ESCs). Flight controllers are designed to assist UAV flight. However, they must be fed information from the user specifying the desired state or position to execute their function. The drone may interpret the desired position in two different ways depending on the mode in which it is operating. If it is in manual mode, then it will know the desired position by receiving values for throttle, roll, pitch, and yaw. If it is in autonomous mode, then it will know the desired position by receiving values for coordinates x, y, and z. The drone must be able to interpret the desired position for every instance of time. According to this

information, the flight controller will receive information from its navigator and send separate digital signals to each of the UAV's ESCs. Then, the function of the ESCs is to receive these digital signals known as pulse width modulation (PWM) and control the revolutions per minute of the UAV's motors and power them at the same time. Since the UAV is being designed from scratch, all these elements must be selected individually, which requires comparisons between products that would perform the same function in our UAS and verify compatibility between each of the selected products. This selection process will be described in the following section.

PROJECT SOLUTION

This project required a multidisciplinary team. The team was divided into sub-teams per discipline according to the different UAV elements to be considered. The Electrical Engineering (EE) team focused on identifying the UAV's flight technique, power supply, control technique and flight controller. The Computer Engineering (CE) team focused on identifying the interface to be developed to facilitate communication between the user and the UAV. Finally, the Mechanical Engineering (ME) team focused on identifying the UAV's motor, propellers, and frame according to the multi-rotor's weight and thus, required thrust. This paper will show the complete UAV design process through the Project Solution section.

A. Multirotor Technology

One of the factors to take into consideration when designing a UAV is the type of multi-rotor technology that will be implemented. The term multirotor is used for any rotorcraft containing more than two rotors. Multi-rotors facilitate an aircraft's flight control. There are many types of multirotor to choose from and these are categorized by the number of motors used. The most common are the tri-copter, quad-copter, hexa-copter, and octo-copter [1].

To select the type of multirotor to be implemented, each of the mentioned multirotor types must be evaluated. It is important to recognize that there is no such thing as the best motor configuration. The motor configuration is selected according to the aircraft's needs. While the increase in number of used motors increases the aircraft's lift capacity and redundancy, it also decreases the aircraft's power efficiency and increases manufacturing and maintenance costs [2].

The final selection for the type of multi-rotor to be implemented was the quad-copter. This selection was made due to the quad-copter's mechanical simplicity and popularity among UAV designers and enthusiasts which allows to use a greater amount of references for the design of the UAV.

B. Quadcopter Control Technique

The control technique to be implemented on the quadcopter system must also be considered. Every system has a set of inputs and outputs. Then, the objective of control system engineers is to identify the necessary inputs to produce the desired outputs in a system known as the setpoints.

For the case presented in this paper, the inputs for the control system come from the user interface and the remote controller depending on whether the quadcopter is set to run on autonomous or manual mode. The controller is the processing unit to be implemented. The process is the quadcopter itself. The output is the pulse width modulation (PWM) to be inputted to each motor and produce the required thrust in each of them to achieve the desired quadcopter state. Finally, through the quadcopter's sensors the current state of the quadcopter is measured and feedback to the control system producing its setpoint. The previously mentioned elements resemble that of a typical closed-loop control system shown in Figure 1.

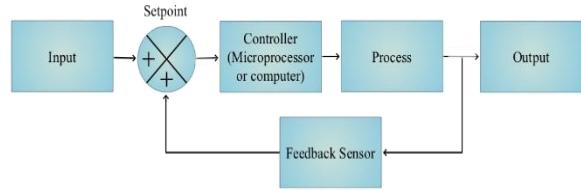


Figure 1: Typical Closed-Loop Control System

The final selection for the control technique to be implemented was the PID technique. This technique is one of the most commonly used control strategies for quadcopters [3]. It continuously measures the error signal (e) identified through its setpoint which is equal to the difference between the desired output and the actual output. The function of the PID controller or control signal (u) may be identified through the following equation:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_p \frac{de}{dt} \quad (\text{Eq.1})$$

The tuning for this control system is executed by iteratively changing the parameters of each PID controller and observing its respective step response. The following principles must be taken into consideration while changing PID parameters:

1. Adjust P to get fast response, but you will get oscillation at steady state.
2. Adjust I to get rid of oscillation, but you will get overshoot.
3. Adjust D to lower overshoot.

However, to properly implement the control technique, one must mathematically characterize the process that is present in the system. Through a mathematical model of the process one can determine the inputs to place into the system and the controller to design to obtain the desired outputs.

In this case, a nonlinear modelling approach was chosen to describe a more realistic representation of the process to be controlled.

When modeling a quadcopter, it should be understood that the quadcopter is an underactuated system since it has 4 motors and 6 degrees of freedom (6DOF). The directions in which the quadcopter may move are divided into 2: translational directions (up/down, left/right, forward/backward) and rotational directions (roll, pitch, yaw). Rotation and thrust is then coupled to accomplish the goal of control over a quadcopter's movement. The final mathematical nonlinear model of the quadcopter is shown in the Mathematical Justification section of the paper and the complete derivation process is shown in the project report

included in the following website: <https://github.com/PUPR-Capstone/Quadcopter-design>.

C. Quadcopter Components

Another factor to take into consideration for the design of a UAV is the set of components to be implemented which includes the following: control board; sensors and peripherals; remote control and receiver; motors, ESCs, and propellers; frame materials; power supply and power management board.

UAV Control Board

Initially, it was assumed that Honeywell would not permit the implementation of the Pixhawk or Ardupilot flight controllers which are ideal for the open source development of autopilots. As a result, the following three processing units were considered for the physical implementation of the control system: the Arduino Nano development board, the NVIDIA Jetson TK1, and the ARM Cortex-M4F LaunchPad. To execute a final selection for the UAV control board a decision matrix, shown in Table 1, was devised using a criterium with the Arduino Nano as the datum. In other words, each control board was evaluated against the Arduino Nano and was given a score between 1 and -1 to show whether the board was better or worse than the Arduino Nano for each criterion.

Table 1: Control Board Decision Matrix

Criteria	Problem Statement	Weight Percentage	Arduino Nano	NVIDIA Jetson TK1	ARM Cortex-M4F LaunchPad
Simulink Integrability	40.00%		-1	0	
Hardware compatibility	40.00%		1	0	
Ease of use	10.00%		-1	-1	
Energy Consumption	3.33%		-1	0	
Volume Constraints	3.33%		0	0	
Weight	3.33%		0	0	
Total %	100.00		-0.53	-0.10	

Legend	
1	Improvement
0	Same
-1	Not Acceptable

Through the control board decision matrix, the Arduino Nano was initially selected as the most viable board. Nonetheless, the Computer Engineering team selected the Arrieta G25 board to develop the interface between the user and the Arduino.

However, after being informed that Honeywell would not be able to provide competitors with the HGuideN580 navigation units and that the implementation of the Pixhawk or Ardupilot flight controller was permitted for the competition, the UAV team decided to discard the implementation of the Arrieta G25 processor along with the Arduino Nano board in favor of using either a Pixhawk or Ardupilot

flight controller. Both flight controllers have integrated inertial measurement units which would have been present in Honeywell's navigator. The UAV team had both a Pixhawk and an Ardupilot available for the physical implementation in PUPR's manufacturing room, specifically the Pixhawk 2.1 (Cube) flight controller and the Ardupilot v2.6 flight controller. Thus, each flight controller was individually evaluated and among the two flight controllers available, the Pixhawk was initially selected due to its higher processing capability.

UAV Sensors and Peripherals

Since the HGuideN580 navigator was no longer required, a research was executed to select a set of sensors that would enable the quadcopter to receive a state estimation similar to that which it would have received from Honeywell's navigator that had an integrated IMU and GPS. In terms of the IMU, a selection was not necessary since the selected flight controller (Cube) had an integrated IMU. Thus, the only sensor to be selected was the GPS. Additionally, a telemetry module had to be selected to calibrate the flight controller's integrated sensors through the ground control station.

The evaluated GPS modules were mainly those recommended in the PX4 user guide such as the: Zubax GNSS 2, Here GNSS GPS (M8N), Here+ RTK GNSS/GPS. The majority of the considered GPS modules are categorized as Global Navigation Satellite System (GNSS) as well. However, an additional GPS module that is compatible with Ardupilot was evaluated to leave open the possible of switching from the Pixhawk to the Ardupilot flight controller. To select the GPS module the features for each of the considered modules were identified. The initially selected GPS module was the Here+ RTK GNSS/GPS.

From another point of view, for the telemetry module selection there were two options for its implementation. These options were to either use a telemetry Wi-Fi module or a telemetry radio module. Both offer the same feature of wireless connection, through MAVLink communication protocol, between a ground control station and a vehicle running PX4 or APM firmware. The differences between the two telemetry options are observed in terms of range and data rates. Wi-Fi telemetry modules provide for higher data rates than telemetry radio modules, but telemetry radio modules have a longer connection range than Wi-Fi telemetry module.

Since the UAV was not being designed to fly long distances the Wi-Fi telemetry module was selected.

However, it is important to recognize that this implementation required flashing the ESP 8266 Wi-Fi module and then soldering its output to a cable that fits in the flight controller's inputs. If the procedure failed then, the telemetry radio module was to be implemented.

UAV Remote Control and Receiver

The remote controller along with its compatible receiver is required to manually control quadcopter movement (speed, direction, throttle, yaw, pitch, roll). Through the remote controller the different quadcopter's flight modes such as takeoff, land, return to land, stabilize, mission, etc. can be enabled. An important feature of the remote controller is the number of channels it supports. This number defines how many physical controls can be used on the remote controller to send commands to the quadcopter. Since the quadcopter is an aerial vehicle, the remote controller to be selected must support a minimum of 4 channels to have control over roll, pitch, yaw, and thrust. The remaining channels in the remote controller can then be used to control other mechanisms or activate different flight modes provided by the autopilot.

Additionally, since the Cube flight controller is being used, the selected remote-control system (transmitter + receiver) must be compatible with the selected flight controller. According to the Pixhawk User Guide, the Cube flight controller is compatible with a specific set of RC systems.

The selected remote-control system was the Futaba T6J RC transmitter along with its compatible receiver (R2006GS) because it presented the highest response speed with its frame rate of 6.8ms. Also, its unique helicopter features are believed to facilitate the manual operation of the UAV. It is important to note that the selected remote controller had the least amount of comm. channels. The minimum number of channels that a remote controller should have is 4 to be able to successfully control a UAV's throttle, roll, pitch, and yaw. Additional channels may be used to send additional information to the UAV such as the flight mode in which to operate or whether the motors may be armed or not.

UAV Motors, ESCs, and Propellers

After selecting all the electrical components for the UAV design, the ME team went through the process of selecting the motors, ESC, and propellers which are selected in combination. The type of ESC and propeller to be implemented depends on the selected motor. Then, the specific motor to be implemented depends on the total weight of the quadrotor. For this

reason, the ME team executed a mass analysis procedure to obtain the following outputs: estimated drone weight, frame weight, motor weight, power consumption, and battery weight. The program was able to provide these outputs by taking into consideration factors such as: frame type, weight of electronic components, disc loading, flight time, air density, and figure of merit for the drone.

The final estimation for the quadcopter's weight was 4.85 lb. which is equal to 2199 g. To determine the motor to be implemented, a hovering at 50% had to be ensured. Thus, the estimated quadcopter's weight was doubled resulting in a weight of 4398 g and by adding a safety factor of 20% the total weight was 5277 g. In addition, the thrust system had to be capable of producing at least 5600 g of thrust, or a thrust of 1400 g by each motor. As a result, the Cobra CM-2213/26 950 KV motor was selected. Note that the 22 refers to the motor diameter and the 13 refers to the height in mm. The higher the motor shaft, the more torque to motor will produce and the more weight your system can carry. Additionally, the KV value is the number of rpms that the motor can produce per volt. While a smaller KV value results in more power and a slower propeller speed, a larger KV value results in less power and a faster propeller speed [2].

Once the motor has been selected, the ESCs and the propellers are selected. The ESC is easily determined by looking for the motor's maximum continuous current in the datasheet offered by the manufacturer. The maximum continuous current for the selected motor was 14A. Thus, an ESC capable of withstanding maximum continuous current of 14A or more had to be selected. As a result, the Cobra 20A Opto Multirotor ESC was selected. Then, the propeller was selected by looking at the propeller data chart for the selected motor offered by the manufacturer. In this data chart the manufacturer recommends different types of propeller dimensions from different manufacturers. The considered propeller dimensions for this project were from the manufacturer named Advanced Precision Composites (APC) as this is the brand that is most commonly used for small UAV development. For this reason, only the recommended propeller dimensions from APC were evaluated and the final selection for the UAV's propeller was the APC 11x4.5-MR propeller was selected. The 11 refers to the propeller diameter in inches and the 4.5 refers to the propeller pitch in inches. A 4.5 propeller pitch will cause the quadrotor to move forward 4.5 inches after one full rotation. While a higher propeller pitch produces faster top speeds and draws more current, a lower propeller pitch provides for more torque and maneuverability while drawing less power [2].

UAV Frame Materials

This selection phase was also executed by the ME team. In terms of UAV materials, these were chosen according to the material availability, cost efficiency, manufacturability, thermal properties, resistance to shear stress, bending moment, signal interference, and water resistance. The initially evaluated materials were carbon fiber, weighting, fiberglass, ABS (filament for 3D printing), and PLA (filament for 3D printing). Among these 5 types of materials, PLA was selected due to its low cost and practicality given that the UAV team had manufacturing room equipped with 3D printers. Other materials that were later included to the design were Nylon X and Nylon G filaments and Lexan.

UAV Power Supply and Power Management Board

Most UAVs use batteries as their power supply since fuel engine-based UAVs are expensive, heavily regulated and application specific. In the beginnings of RC aircraft and drone design, Nickel Cadmium (NiCad) and Nickle Metal Hydride (NiMH) were used. However, now the most commonly used batteries on UAVs are lithium based due to the different advantages that they have over NiCad and NiMH batteries such as their higher capacity, higher discharge rate, and lower weight. There are different types of lithium-based batteries. The types that are most frequently considered for UAV design are the Lithium Ion (Li-ion) and Lithium Polymer (LiPo).

Due to its variety in configuration, convenient packaging, and high discharge rate, the lithium polymer battery was found to be the best type of battery to be implemented on the UAV to be designed. However, this selection does not limit the consideration of alternatives for the UAV's power supply. LiPo batteries are offered in a wide variety of configurations according to its C-rating, milli amperes per hour, and number of cells.

The C-rating of a battery is directly related to a battery's discharge rate. The higher the C-rating the lower the internal resistance in a battery, and the higher the discharge rate. This also makes the battery more expensive. It is important to note that some batteries will be sold showing a range of C-rating. An example of a C-rating range offered in batteries is 20C-30C. This range means that the battery will provide a constant C-rating of 20C and can provide burst of 30C. Thus, they should not be used on drones that need a constant C-rating greater than 20C. The LiPo battery used in this project was selected to have a C-rating of 35C.

Another important consideration is the battery's milli ampere-hour (mAh) rating. The mAh rating is an electric charge unit specifying the charge transferred by a steady current of one milli ampere flowing for one hour. Thus, this rating is directly related to the flight time that the battery will be capable of providing. The longer the flight time, the larger this rating must be. The LiPo battery selected for this project has a 4200 mAh rating.

On the other hand, there is the consideration of the number of cells which is directly related to the amount of voltage provided by the battery. The greater the number of cells, the greater the amount of voltage. For each cell in a battery there is a voltage of 3.7 V.

The minimum number of cells (3S) the battery should have to power the selected motors is specified in the selected motor's propeller data chart offered by APC. As a result, the number of cells determined for the battery selected was 4S to ensure power compliance.

After selecting the specifications for the LiPo battery, the flight time for the case in which each motor is drawing maximum current (14 A) may be measured by doing the following division [4]:

$$\frac{\text{Ampere hour}}{\text{total current draw}} \quad (\text{Eq.2})$$

Finally, the power management board component was considered to facilitate the distribution of power among the different electronic components implemented for the UAV design. For this project, the selected power distribution board was the HolyBro PM07 specifically designed for the Pixhawk Flight Controller. This module was selected since a flight controller from the Pixhawk brand had already been selected.

D. Quadcopter Software

Since the use of the Arduino Nano was discarded in favor of the Pixhawk and Ardupilot flight controller, an additional evaluation had to be executed. These flight controllers must run on a firmware to be functional. Also, ground control stations must be used to calibrate the flight controller's sensors. Thus, as part of the design the firmware and the ground control station to be implemented must be selected.

Flight Controller Firmware

While the Ardupilot flight controller can run only on the APM firmware, also known as flight stack, Pixhawk's Cube flight controller can run on either the APM or the PX4 firmware. Thus, there are two

implementation options concerning the firmware for the selected flight controller.

The PX4 firmware is composed of three layers the flight stack (contains individual programs for applications such as flight control and state estimation), the middleware (facilitates communication between programs installed in the flight controller), and the drivers. This architecture allows for a modular design since each of the three layers can run independently. This also allows for the individual development of programs that may then be integrated to the flight controller by installing them on the PX4 flight stack. Additionally, each application connects to other processes and drivers using a Publisher-Subscriber framework. This allows for efficient communication between processes and simplifies the process of adding new applications to the flight controller.

On the other hand, there is the APM firmware to consider which was originally designed for the Ardupilot flight controller but has been ported as a single application to the PX4 flight control architecture so that it may be run on any Pixhawk flight controller through the PX4 middleware layer. By selecting to implement the APM firmware, the APM application will substitute the PX4 flight stack for the APM flight stack to control the flight controller's drivers. From the user's perspective, this allows the PX4 flight controller to behave like the legacy APM hardware.

After evaluating each firmware, the PX4 firmware was ultimately selected since it was specifically designed for the selected flight controller and the greater flexibility that is offered by its flight stack which was not composed of a single application contrary to the Ardupilot firmware. Most importantly, the PX4 firmware had greater compatibility with MATLAB/Simulink since MathWorks had already initiated an open source community effort to promote the implementation of the Pixhawk autopilot through its PX4 Pilot Support Package (PX4 PSP). The PX4 PSP provides the means to implement controllers and models designed in Simulink onto the PX4 flight control hardware. It contains a library of PX4 Simulink blocks along with PX4 example Simulink models that can be used for developing the plant or controller of a vehicle. However, it is important to notice that although the PX4 PSP functions as a good guide to create Simulink models that may be uploaded to the PX4 flight control hardware to optimize its control logic, the PX4 PSP is outdated and cannot be used to upload control logic to the Cube flight controller specifically. Then, to upload the control

logic to the Cube flight controller, MathWorks's Embedded Coder Support Package for PX4 Autopilots is used which contains a library of PX4 Simulink blocks similar to that of the PX4 PSP.

UAV Ground Control Station

A GCS is an application that runs on a ground-based computer and allows users to communicate with their UAV via telemetry. It can display real-time data concerning the UAV performance and position. Commands such as executing a mission may be pre-programmed or sent to the flight controller via the GCS. Additionally, GCSs may be used to calibrate the selected remote controller and the UAV's integrated sensors. There are different ground control stations to choose from. However, the two most commonly used are the Mission Planner, and QGroundControl. Both have GCSs have very similar features, however, while Mission Planner only supports flight controllers running the APM firmware, QGroundControl supports flight controllers running either the APM or the PX4 firmware. For this reason, QGroundControl was the selected GCS.

E. Mathematical Justification

The mathematical model for the UAV must be derived in order to design its control system. In order to describe the behavior of the UAV the mathematical model is defined based on the rigid body theory, the Earth fixed frame coordinate system and the Body fixed frame coordinate system. By using mathematical model, it is possible to predict position and attitude of the UAV and allow us to simulate the quadcopter behavior using simulation tools such as Simulink.

The first step is to define the mathematical equations based on the UAV kinematic features which addresses the relation between the vehicle's position and velocity. The second step is to identify all the forces and momentum acting on the UAV which results in the equations for translational and rotational accelerations.

Due to its complexity, the derivation for the mathematical model is not shown in this paper. However, the complete mathematical nonlinear model for the UAV is shown through the following equations in matrix form shown in Figure 1.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \cos \theta \cos \psi & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & \cos \psi \sin \theta \sin \phi + \sin \psi \cos \phi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \psi & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} + \begin{bmatrix} -g \sin \theta \\ g \cos \theta \sin \phi \\ g \cos \theta \cos \phi \end{bmatrix} + \frac{1}{m} \begin{bmatrix} 0 \\ 0 \\ -F^b \end{bmatrix}$$

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

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} J_{yy} - J_{zz} & qr \\ J_{xx} & \\ \frac{J_{zz} - J_{xx}}{J_{yy}} pr & \\ J_{xx} - J_{yy} & pq \\ \frac{J_{zz}}{J_{yy}} & \end{bmatrix} \begin{bmatrix} \frac{1}{J_{xx}} \tau \phi \\ \frac{1}{J_{yy}} \tau \theta \\ \frac{1}{J_{zz}} \tau \psi \end{bmatrix}$$

Figure 1: Mathematical nonlinear model for the UAV in matrix form

From another point of view, the parameters that are used on the previously shown equations are described through Table 2.

Table 2: Parameters for UAV nonlinear mathematical model

Variable	Description	Unit
x	Position along X-axis in inertial frame	m
y	Position along Y-axis in inertial frame	m
z	Position along Z-axis in inertial frame	m
ϕ	Roll angle with respect to body frame	deg
θ	Pitch angle with respect to body frame	deg
ψ	Yaw angle with respect to body frame	deg
u	Velocity along the X-axis in body frame	m/sec
v	Velocity along the Y-axis in body frame	m/sec
w	Velocity along the Z-axis in body frame	m/sec
p	Roll rate with respect to body frame	deg/sec
q	Pitch rate with respect to body frame	deg/sec
r	Yaw rate with respect to body frame	deg/sec
J_{xx}	Moment of inertia along the X-axis in body frame	kg.m^2
J_{yy}	Moment of inertia along the Y-axis in body frame	kg.m^2
J_{zz}	Moment of inertia along the Z-axis in body frame	kg.m^2
F^b	Nongravitational forces acting on the quadrotor in the body frame	N
m	Mass of quadrotor vehicle	kg
τ	Torque	Nm

F. Testing Phase

This section of the paper is divided into various subsections such as: Power Demand Test, Dynamometer Test, Simulink Tests, and Ground Control Station Tests. The following content describes how each of the necessary tasks for the testing phase were executed.

Power Demand Test

The objective of the power demand test was to determine the average kWh consumed by one motor while generating the necessary thrust to lift $\frac{1}{4}$ of the quadcopter's weight. The following graph was produced through this test.

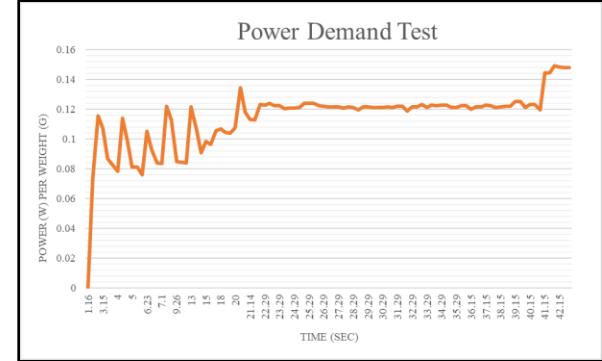


Figure 2: Power Demand Per Weight

From the obtained data while the motor's throttle was maintained at an approximately constant rate (from the time period of 22.29 s to 40.15 s), it may be stated that the motor consumes approximately 0.12 W/g.

Dynamometer Test

The objective of the dynamometer test was to determine the thrust and drag factor for the selected motor-propeller configuration as shown in Figure 3.

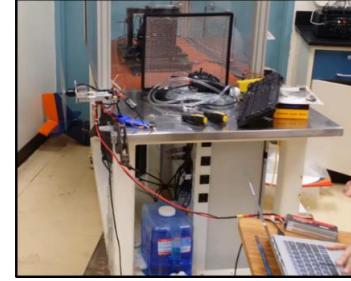


Figure 3: Dynamometer Test

The following table was produced from the dynamometer test which facilitated the modeling phase for the designed quadcopter by providing the values for the propeller's drag and thrust factors.

Table 3: Characterizing motor-propeller configuration

Thrust coefficient (C_t)	Power coefficient (C_p)
0.102	0.0449
Thrust factor (b)	Drag factor (d)
1.9366×10^{-5}	3.7909×10^{-7}

Simulink Tests

The objective of the Simulink tests was to determine a quadcopter model from which quadcopter flight may be successfully simulated.

Simulink Reference #1: Quadrotor Dynamic Model

The performance of the Quadrotor Dynamic Model, shown in Figure 4, did not result as expected. The Simulink PID Tuner Application was used to find the values of proportional, integral, and derivative gains of each PID controller to achieve desired performance and meet design requirement, but the performance was far from ideal.

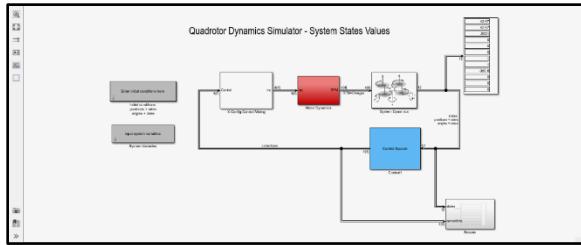


Figure 4: Quadrotor Dynamic Model

Simulink Reference #2: sl_quadrotor model

The sl_quadrotor model, shown in Figure 5, was set up with our quadrotor parameters and the model was simulated for proper tuning. The quadrotor trajectory plot was produced through this model. Additionally, a disturbance was added to observe its effect on system performance. It was observed that the responses were significantly similar, so it is understood that the model was adjusted correctly.

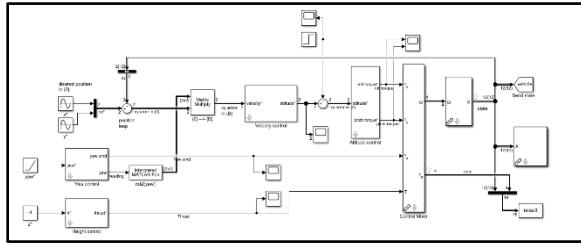


Figure 5: sl_quadrotor model

Simulink Reference #3: QuadrotorSimulink model

Finally, the QuadrotorSimulink model, shown in Figure 6, is similar to the sl_quadrotor model. This model also includes a plot of the quadrotor trajectory. However, this model makes use of the Simulink UAV Animation block instead of the plot3 function. The advantage is that the block animates the UAV flight path based on the translations and rotations estimated by the model. However, the results were limited since the model did not provide for the input of waypoints.

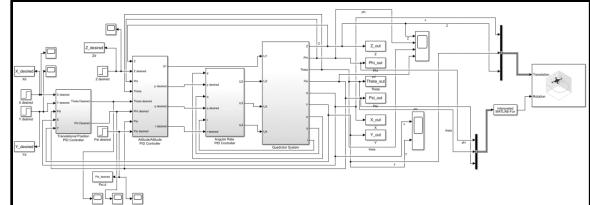


Figure 6: QuadrotorSimulink model

The references were mainly used to execute the PID tuning. Once the proper PID parameters were identified, a Simulink model was created containing the control system to be uploaded into the flight controller hardware by using Simulink's supported Pixhawk flight controller blocks.

G. Quadcopter Wiring

Once all the factors concerning the quadcopter's hardware and software analyzed, the team proceeded to wire all the quadcopter's electronic components to be used in the final design. The wiring for the final design is shown in Figure 7.

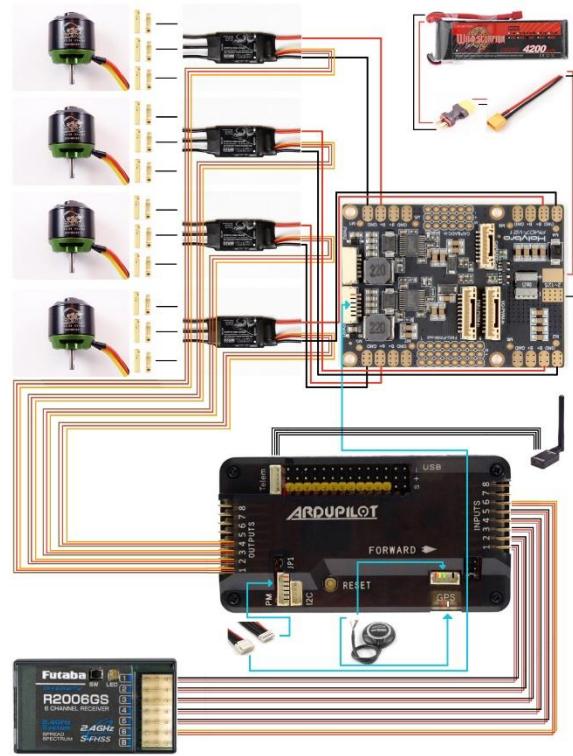


Figure 7: Wiring for final design

H. System Requirements

After finalizing the UAV design, the following table was developed to summarize the UAV system requirements.

Table 4: PUPR Quadcopter System Requirements

Component	Power Requirement	Weight
ArduPilot Mega 2.6 Flight Control Board	Operating Voltage: +5 V	28 g
X94 3DR Radio Telemetry Kit	Supply voltage: 3.7-6V DC	19g
XT-XINTE GPS	Working voltage is 5V	29g
Cobra CM-2213/26 Multirotor Motor	950 RPM per Volt 20 A (Max), 300 W (Max)	65g
Cobra DL20A Electronic Speed Controller	Operating Voltage: 6-16 V; Maximum Continuous Current: 20 A	12.2g
Power Supply	LIPO 4S 14.8V 4200MAH 35C	419g
Power Distribution Board	12V and 5V outputs	13g
Futaba 6J 6-Channel S-FHSS Remote Controller System	6-Channel 2.4GHz S-FHSS and FHSS Modes	N/A

PROJECT RESULTS

The results from this project can be divided into two categories; those related to software and those related to hardware. In terms of software, a Simulink model to be uploaded to the Cube flight controller was devised as shown in Figure 8.

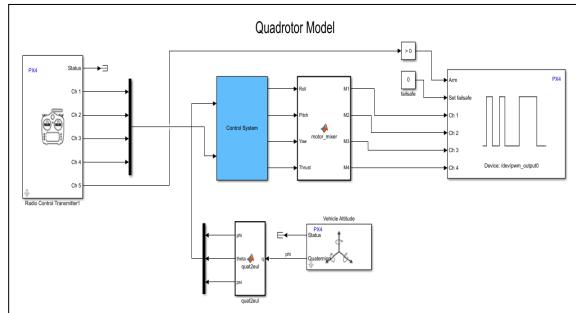


Figure 8: Simulink model for Pixhawk 2 (Cube flight controller)

However, the production code could not be generated due to complications during the build process. There were some communication buses conflicts that impaired the process to complete successfully and the code could not be uploaded. Further research is required in order to resolve the issue. As an alternative, the UAV's PID gains were successfully adjusted through the implementation of Mission Planner and its Extended Tuning section.

In terms of hardware, the UAV's propellers guards required redesign due to their tendency to impact the propeller during flight which caused flight instability. Additionally, the material selected for the UAV's top and bottom base (Lexan) appears to shatter easily.

CONCLUSIONS

Through this project a quadcopter flight simulation was successfully executed through the implementation of MATLAB/Simulink and Mission Planner which facilitated the design of a UAV that has both remote and autonomous features. Consequentially, this project opens the doors to future projects related to the topic of UAV design by serving as a reference. As a recommendation for future projects on UAV design, it is stated that the implementation of Simulink with Pixhawk requires further investigation. Precaution is advised when uploading firmware to telemetry modules since communication problems between the modules may occur if the wrong firmware is installed. Additionally, an alternative to UAV design concerning autonomy could be to program the Ardupilot flight controller using Arduino IDE.

REFERENCES

- [1] D. Kotarski, M. Krznar, P. Piljek and N. Simunic, "Experimental Identification and Characterization of Multirotor UAV Propulsion", *Journal of Physics: Conference Series*, vol. 870, p. 012003, 2017. Available: 10.1088/1742-6596/870/1/012003.
- [2] O. Liang, "Types of Multirotor", *Oscar Liang*, 2016. [Online]. Available: <https://oscarliang.com/types-of-multicopter/>.
- [3] W. Fum, "IMPLEMENTATION OF SIMULINK CONTROLLER DESIGN ON IRIS+ QUADROTOR", *SemanticScholar*, 2015. [Online]. Available: <https://pdfs.semanticscholar.org/19b1/aafc474c/bcd61b27d33475d05f7a9c6344e.pdf>.
- [4] M. Gerber, "Motor, Propeller, ESC, and Battery Sizing Introduction", *Maclab.seas.ucla.edu*, 2016. [Online]. Available: http://www.maclab.seas.ucla.edu/downloads/16_2D_Sizing_Introduction.pdf.

Chapter 11: Administrative Section & Appendix

Proposal Report

Progress Reports

Work Schedule

Appendix: PDR Presentation

Appendix: CDR Presentation

Appendix: FRR Presentation