Autonomous UAV for the International Aerial Robotic Competition

ECE 487 Final Report

Prepared by:

Bradley McKee (CpE)

Faculty Advisor:

Dr. Chung-Hao Chen

Old Dominion University

Department of Electrical and Computer Engineering

April 20th, 2018

ODU Honor Pledge

"i pledge to support the Honor System of Old Dominion University. I will refrain from any form of academic dishonesty or deception, such as cheating or plagiarism. I am aware that as a member of the academic community it is my responsibility to turn in all suspected violation of the Honor Code. I will report to a hearing if summoned."

Abstract

This proposal introduces the design for the unmanned aerial vehicle (UAV) that will be used to compete in the 2018 International Aerial Robotics Competition, which will be held on July 31st – August 2nd, 2018. The IARC is the longest running collegiate aerial robotics challenge in the world and is funded by AUVSI. The competition was designed to incorporate 3 behaviors. First, interaction between the UAV and moving objects. The second behavior, navigate the UAV through a sterile environment devoid of navigation aids, such as GPS or large stationary points. The third behavior is avoidance between other competing UAVs. All vehicles that enter this competition must be autonomous. The drone in its current state can be flown from the flight joystick transmitter/receiver or from a Linux terminal. When launching the UAV with a terminal, it uses a series of code and scripts to create autonomous flight. My main goal is to further research and development to help create sense and avoid technology for UAV's.

The current state of the drone is that it is manually operational through a joystick and receiver that has a 2.4 GHz frequency band. It can take commands from a terminal and can deploy the sensors through python scripts. …but autonomous flight has not been achieved nor has data communication between sensors and flight l. Our goal is to further the research and development to create the sense avoid technology for a UAV.

Table of Contents

1. Introduction 5
1.1 The Competition 5
2. Design Approach 6
2.1 Summary of Design Method 6
2.2 Detailed Description 7
2.2.1 Task 1: Image Processing 7
2.2.2 Task 2: Control Structures 8
2.2.3 Task 3: Integration of Systems 8
2.2.4 Task 4: Stabilization 8
2.3 Parts List and Analysis 9
2.3.1 Chassis 9
2.3.2 Raspberry Pi 3/Navio2 9
2.3.3 RPLIDAR 10
2.3.4 PX4FLOW v1.3 10
2.3.5 HC-SR04 10
2.3.6 915 MHz and 2.4 GHz (Wireless communication) 11
2.3.7 Electronic Speed Controller 11
2.3.8 Brushless DC Motors 12
2.3.9 Battery 12
2.3.10 Arducopter 13
2.3.11 Mission Planner 13
2.3.12 Ubuntu 13
2.4 Alternative Design 14
2.5 Realistic Constraints 14
3. Project Considerations 15
3.1 Engineering Ethics 15
3.2 Engineering Standards 16
4. Broader Impact 17
4.1 Economic and Environmental 17
4.2 Health and Safety 18
4.3 Social 18

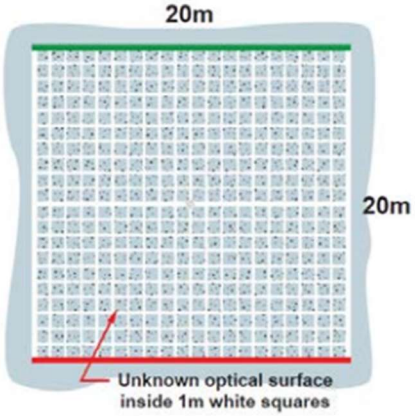
4.4 Global 18
5. Lifelong Learning 18
6. Project Contributions 19
6.1 Team 1 – Physical construction/Start-up scripts 19
6.2 Team 2 – Sensors on UAV 19
7. Testing and Analysis 20
7.1 Camera and Visualization 20
7.2 Battery Life 21
7.3 Stabilization 21
8. Experimental Results 22
References 23
Appendices 23

List of Figures

Figure 1: Arena Layout 5
Figure 2: Quadcopter axis' 6
Figure 3: Chassis of Quadcopter 8
Figure 4: Raspberry pi with Navio2 attachment 8
Figure 5: 360 sensor 9
Figure 6: Screen Capture of live feed from 360 Sensor 9
Figure 7: Optical Flow Smart Camera 10
Figure 8: 3DR Telemetry with Navio2 10
Figure 9: ESCs connected to the Navio2 10
Figure 10: Brushless Motors 11
Figure 11: Quadcopter Battery 12
Figure 12: Battery Graph 12

Introduction

This project is required by Old Dominion University's Electrical and Computer Engineering department for the undergraduate Bachelor of Science degree. Students are to use knowledge developed from previous semester to work on a group project. The project for this group is to design an unmanned aerial vehicle (UAV) to enter into the International Aerial Robotics Competition (IARC). The competition was designed to incorporate 3 behaviors. The first behavior, interactions between the UAV and moving objects. The second behavior, navigate the UAV through a sterile environment devoid of navigational aids, such as GPS or large stationary points. The third behavior is avoidance between competing UAVs. All UAVs must be autonomous. The scope of this project is too large for the amount of people in this group. The focus of this group will be to obtain stable flight and the install a kill switch that will immediately disable the UAV [1].



1.1 The Competition

For this competition each team's UAV will have 10 minutes to guide, a maximum of ten, ground robots into a designated area. These ground robots are programmed to travel in a straight line and reverse direction by 180° after 20 sec. If an UAV comes into contact with the ground robot from above, the ground robot will turn 45° clockwise. If the forward motion of a ground robot is blocked, the ground will change its direction by 180°. There will also be ground robots (obstacle robots) with tall, vertical cylinders attached to them that will move in random directions. In addition to the ground robots there will also be another competing UAV trying to guide the ground robots. Our UAV must be able to guide the ground robots while avoiding the obstacle robots and the other UAV all without the aid of GPS. The UAV cannot fly more than 2 meters outside of the area for more than 5 seconds nor can the UAV fly more than 3 meters above the ground. All UAVs must be equipped with a kill switch in order to immediately disable the UAV at the discretion of the judges [1]

2. Design Methodology

The two objectives of this project are to maneuver an Unmanned Aerial Vehicle, or UAV, by using two sensors attached to the device and to establish stabilized flight autonomously. These

two sensors are designed to detect other objects around it so that it can avoid a collision. The sensor does this by communicating with a computer that is attached to the device. The computer then relays information to a host computer that collects the data during the flight. The Raspberry pi with the Navio2 shield will serve as platform to control the UAV. From the host computer, the pilot of the device can give the UAV instructions based on information relayed through the sensors.

As an individual developer I went through various designs. A lot of the design approach is constantly changing, the biggest changes come from the sensors that I would attach to the drone. The biggest change in our design would be the choice to not use the RPILIDAR over some of the hardware that I have readily available. Instead of the omni-directional sensor, I will be using four individual sensors. The sensors that I will be attaching to each side of the drone would be the HC-SR04, and one sensor would be used for each side of the drone (North, East, West, South). These sensors are to be mounted on the bottom side of the quadcopter.

2.1 Summary of Design Method

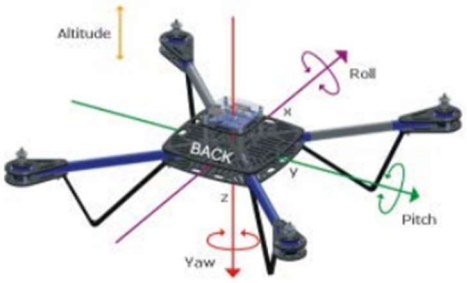


Figure . Quadcopter axis'

Our drone UAV consists of a Raspberry Pi Model 2 B paired with a Navio2 that utilizes a RPLIDAR sensor and a PX4FLOW v1.3 along with the motors and ESCs to produce lift. This design has two parts a hardware and a software. The main focus of the physical design for the drone is to keep the center of gravity in the middle. The physical design consists of mounting various objects and sensors to the bare UAV. The software design of this project is to fully develop the operating system and user interface for the drone. Our whole design will center around an autonomous UAV that has sense and avoid technology.

The onboard computer is receiving its instructions through a Raspberry transmitter. The transmitter works in junction with a PX4Flow v1.3 smart camera. These are the devices that analyze the area surrounding the UAV and sends information back to the host computer. These devices are primarily used so that the pilot can see and instruct the UAV on its flight path. The sensors are responsible for detecting hazards, such as walls, ceilings, floors, and any other obstructions. The PX4Flow v1.3 has an ultrasonic sensor built into it, so then I will only need to use three other ones to cover the sides that the camera mount is not facing. All the sensors will be mounted with a 3d printed bracket that fits the sensor precisely because if there was any vibrations or any obstruction of the sensors, it just simply wouldn't function correctly. Once I verified that none of the

ultrasonic sensors were broken, I needed to create a housing for it. The housing will protect the sensor and hold it in a secure location.

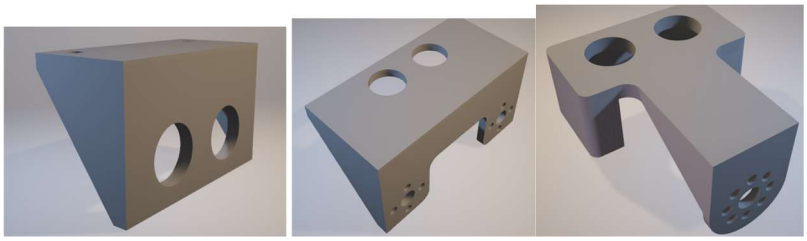


Figure . 3D Drafted mounts

The figures above show some of the alternate designs I came up with when discussing how to mount the HC-SR04 sensors. This plays a crucial role in the design because the quadcopter needs to be aerodynamic. The current mounts should have minimal material because it needs to be light as possible. These are all factors considered when designing the quadcopter.

2.2 Detailed Description

Our implementation will include three tasks. The first task will be setting up the Image Processing. The second task is the control structure of the quadcopter and lastly the Integration of all systems.

2.2.1 Task 1: Image Processing

In order to detect and track the ground robots the UAV is equipped with a small camera. This camera will be the megapixel Raspberry Pi camera module. The feed from the camera will be sent to the Raspberry Pi 3 microprocessor onboard the UAV. The Raspberry Pi 3 will have the open-source imaging software, known as OpenCV, and relay the appropriate navigation commands to the Navio2. The Navio2 will take the feed from the camera and pipeline it to the Raspberry pi to isolate only color of the ground robots and record the path of the ground robot to determine its trajectory. According to the IARC rules, 5 ground robots will have red plates on top of them and the other 5 will have green ones. Using the OpenCv software, the UAV will be able to locate and track the red and green plates. There is a function in OpenCv to find the center of a specified object and then perform the operations accordingly.

2.2.2 Task 2: Control Structures

This UAV will have to perform several aerial maneuvers in this competition. Those maneuvers include ascension, descension, hovering, pitch, roll, yaw (turning left or right) and bounce (rapid descension, make contact to the ground and ascend). The first step for flight is to have the appropriate propeller direction. For ascension, descension and hover the onboard processor will send the appropriate power level to the electric motors equally. For pitch forward, I will increase the speed for motors 3 and 4 and decrease the speed of motor 1 and 2. Reverse the motor speeds for pitch backwards. For roll left, increase the speed of motors 2 and 3 and decrease the speed for motors 1 and 4. Reverse the motor speeds for roll right. For yaw left, increase the speed of

motors 2 and 4 and decrease for 1 and 3. Reverse the motor speeds for yaw right. For hovering, this UAV will use a combination of the optic flow sensor and the onboard barometer from the Navio2 to keep the distance from the ground constant (no more than 3 meters) and prevent horizontal drift.

2.2.3 Task 3: Integration of Systems

In the current setup of the UAV, the camera send the image directly to the ground station, via a rf link, where the image is processed and the appropriate command was send back to the UAV via RF link. Since the UAV will be in an environment where it has to keep track of several moving objects, there will need to be fast communication between the camera, the processor and the flight controller. To meet this demand I have decided to use a Raspberry Pi microprocessor and the NAVIO2 autopilot shield programmed with Ardupilot. Ground control will be using a variation of Mission Planner, Arducopter and QGroundControl to directly communicate with the quadcopter.

2.2.4 Task 4: Stabilization

The code is based on the dronekit kit library created by developers all over the world in order to create a simple way to code to any UAV. Using these libraries I was able to manipulate the drone in order for it to stabilize itself go up 10 meters from the initialized launch zone then drop back down to that drop zone. The code first starts by declaring the drone. Then it connects to the drone using the argparse class. This allows us to code to the drone. I added an error code if three is an issue with connecting to the drone.

I then added the arm_and_takeoff function to arm the drone and have it fly to the desired altitude that is specified in the ATargetAltitude function.. Since I'm using the computer to fly this drone I cannot arm it using the transmitter, this line of code allows us to bypass that. On top of that I have to make sure that the drone checks to see if it is armed or disarmed. I did this by using the vehicle.is.armable function and had it spit out a text to tell us that it is currently arming. Once it is armed it then goes to the vehicle.simple_takeoff(aTargetAltitude) function which powers the motors and tells the vehicle to fly to the target altitude using the built in compass on the Navio 2. When then have it tell us the location of our drone as it flies. I then have it wait at 10 meters for 10 seconds and then it returns to the original place that it took off from. I then stop the loop manually using the terminal.

2.3 Parts List and Analysis

2.3.1 Chassis

The frame of the quadcopter is the S500, which is 20cm high and 39cm X 39cm and weighs 403g. There will be addition parts that will be 3D printed to help minimize the vibrations given from the motors.

2.3.2 Raspberry Pi 3/Navio2

Figure 4. Raspberry pi with Navio2 attachment

The Navio2 is autopilot hat, developed by company name Emlid, for the Raspberry Pi. It is equipped with a gyroscope, accelerometer, and a magnetometer for orientation in 3-D space. In addition, it also has an analog to digital converter (ADC), 14 PWM output pins, and an I2C, and UART interface [2].

The Navio2 was originally designed to be used for the ArduPilot system. ArduPilot is an open source autopilot software for RC vehicles. I will be repurposing the Navio2 by writing custom code for the IARC mission. The Raspberry Pi has Software Development Kits (SDK) for the sensors, so that it can be processed and communicated to the flight controller.

2.3.3 RPLIDAR

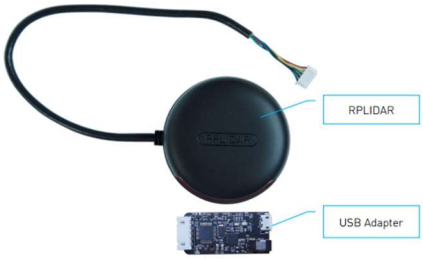


Figure 5. 360° sensor

The RPLIDAR A2 development kit contains standard RPLIDAR A2 unit (A2M4-R1). The RPLIDAR is embedded with logic IO drivable motor controller which can be used to configure the scan frequency by tuning motor speed [3]

This is omnidirectional sensor, it has the ability to do real time mapping, but takes up a lot of Central Processing Unit (CPU) of the Raspberry Pi. The motor can also be turned off to reduce power consumption, which is a huge obstacle that I needed to overcome when supplying power to the quadcopter.

Due to the input voltage varying from 0V-5V, the Raspberry Pi's USB port can only supply 3.3V out, The alternative design would be to remove the Serial to USB converter and directly get power from the rail of the Navio2.

2.3.4 PX4FLOW v1.3

Figure 7. Optical Flow Smart Camera

PX4Flow is an optical flow smart camera (it provides the image for setup purposes, but it not designed to capture images like a webcam). It has a native resolution of 752×480 pixels and calculates optical flow on a 4x binned and cropped area at 400 Hz, giving it a very high light sensitivity [4]. In addition to being a flow smart camera, it has an onboard sonar sensor as displayed in the figure. I used the OpenCV open source library to perform image processing.

2.3.5 HC-SR04

Figure . HC-SR04 Module

The HC-SR04 module has the ability to tell distance (in metric[mm]) between an object and the sensor. This sensor is perfect especially when in line with the Raspberry Pi and Navio 2 hat. In

the documentation of the Navio 2, it states that it converts almost all pins to have a neutral and 5v power rail for each pin once assigned and programmed. Since it can easily be integrated with our existing hardware and can send commands to the Arducopter from what the sensor is reading.

2.3.6 915 MHz and 2.4 GHz (Wireless communication)

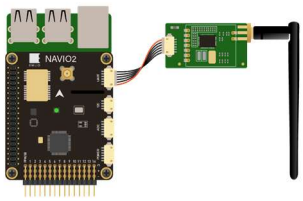


Figure 8. 3DR Telemetry with Navio2

The transmitter and receiver are used to establish a linked connection between the quadcopter and a computer for manual controls and to deploy the scripts for autonomous flight. For this project I'm using 3DR telemetry has a lot of open-source support through SIK firmware. The 2.4 GHz is a wireless band that is used for File Transfer Protocol (FTP) between the quadcopter and computer. I have been successful on loading various C++ and python code to the quadcopter and run simulations.

2.3.7 Electronic Speed Controller

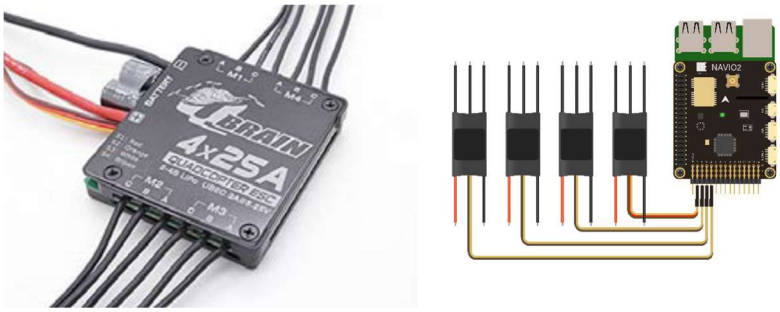
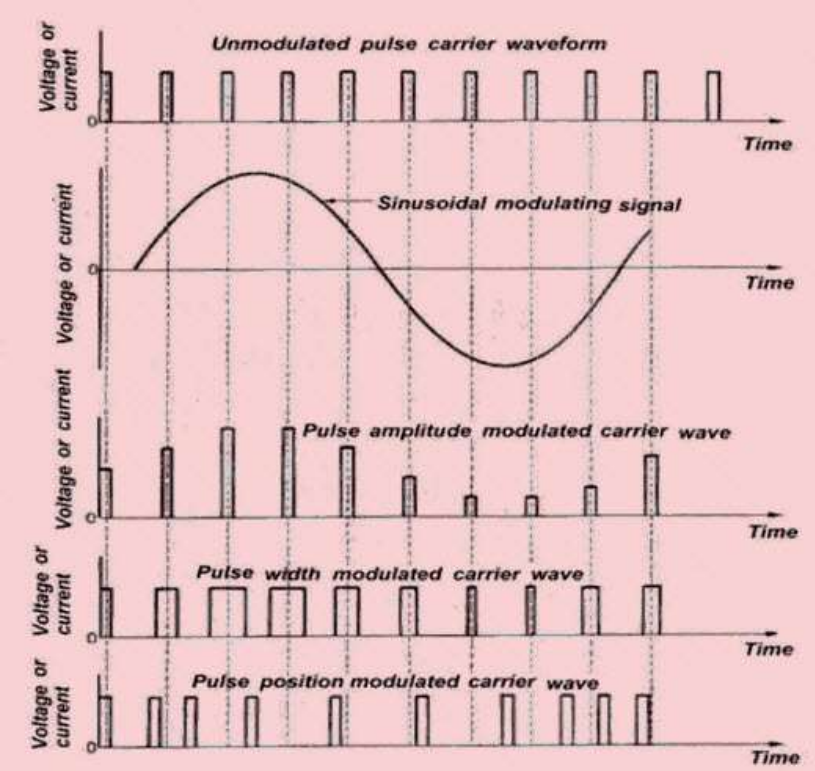


Figure 9. ESCs connected to the Navio2./Q-brain

The Electronic speed controller (ESC) for the quadcopter is a 4 in 1 from the brand Q-brain. It controls the flow of power to the motors very effectively. The Q-brain can provide bursts of up to 25A to each motor at any given time. With the control and calibration of Q-brain, I was able to effectively control each motor without sporadic bursts outputted by the battery. The power delivered to the motors is based on a pulse width modulated signal (PWM) and the signal is received from the flight controller or controlled autonomously through various scripts of code.

2.3.8 Transmitter and Receiver

The transmitter and receiver are used for manual controls for the quadcopter. The switches on the transmitter are used to initialize autonomous flight through the attached sensors. For this project, I used the FlySky FS-i6 and the FS-IA6B 6 Channel receiver in Pulse Position Modulation (PPM). This information is important because I know that ESC and Navio2 sends signals using PWM instead of PPM. This created a lot of complications because the position of the controls on the transmitter or code will determine the duration of "on" time. The ratio of the signal "on" time compared to signal "off" time is called the duty cycle.



https://www.elprocus.com/difference-between-pam-pwm-ppm/

2.3.9 Brushless DC Motors

Figure 10. Brushless motors

This quadcopter uses 4 1000KV brushless DC motors. The KV rating "loosely" means revolutions per minute(RPM) per volt at no load. A more accurate description would be the ratio of the RPM to the back electro-motive force(EMF) of the motor [5].

$$RPM_{no\ load} \approx \frac{KV_{rating}}{\text{EMF}}$$

These motors should be rated to take an input max of 12 amps. Four of these motors at its maximum amperage can pull 48 amps, from prior calculations the battery is sufficient enough with a comfortable overhead.

2.3.10 Battery

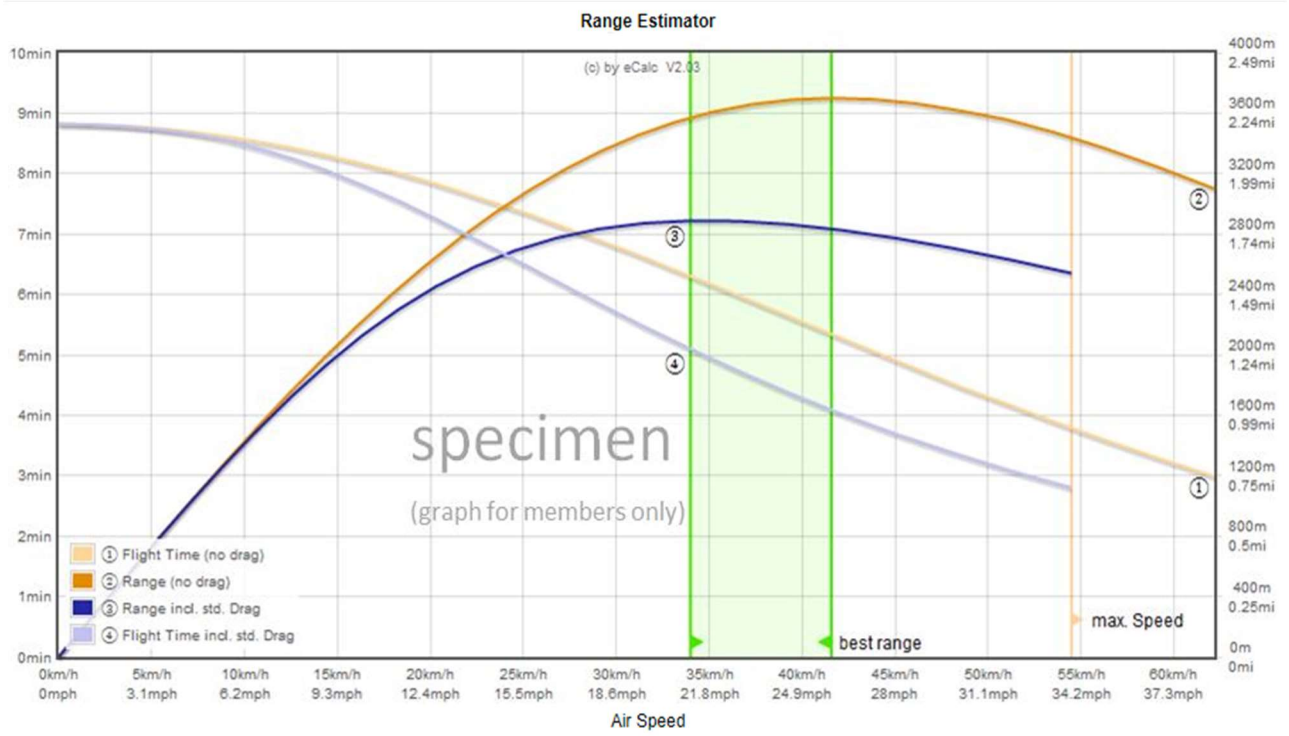
Figure 11. Quadcopter Battery

The battery for the quadcopter is a 4S (14.8 volt), 50C, 8000 mAh lithium polymer as our main battery. 25C is the Crating. A C-rate is a measure of the rate at which a battery is discharged relative to its maximum capacity [6]

$$C_{rating} = \frac{Current(A)}{Capacity(Ah)}$$

From the C-rating formula I calculated that the current battery that I'm using can theoretically provide 77 amps safely to the motors, I'm looking at a battery that has around the same voltage, but a higher mAh.

I'm still experimenting with the battery, propellers and motors to find the optimal flight time. One observation that I noticed, is as I increased the mAh it became much more difficult to stabilize manual flight, with no success of automated flight.



The figure above shows an estimation of how long the current battery that I'm using will last, calculations are done through eCalc. From the competition specifications, the mission test is allowed 10 minutes to complete the given tasks. The graph shows optimal range and speeds to ensure that the battery will last during the whole mission test.

2.3.11 Arducopter

Arducopter is used as the controller that interacts with the UAV. This allows us to code and configure the Raspberry Pi with operations.

2.3.12 Mission Planner

Mission Planner is used with Arducopter. It details what happens during the flight and monitors the voltage, amperage, sensor readings, etc. Using Mission Planner will help with flight simulation, both prior to and during actual testing. It will help pinpoint error in order to help stabilize flight and actually fly around.

2.3.13 Ubuntu

Ubuntu is a Linux Operating Software I use to program the Raspberry Pi's settings, such as running scripts and adding libraries. This will also help when I am creating an IP address for the drone as well as connect it via Wi-Fi.

methodical analysis and results:

2.4 Alternative Design

Although many alternative designs have been considered, I'm always looking at what is the best hardware and software is for how the interaction between two devices are taken. I'm constantly finding better algorithms that will help us develop UAV stability and how the data from the sensors are formulated to ensure object avoidance.

I have went through many different open source flight controller software that can be used with the Navio2. LibrePilot is an open source software that I loaded onto the Navio2 flight controller. I found it very hard for it to take predetermined paths using GPS because it was set up for manual control. Another software that I attempted to use was a program called QGroundControl.

From a hardware standpoint, with increasing the capacity to a higher mAh such as 5000 mAh versus the current 2200 should result in better stability and less fluctuation in amperage. Another alternative design could come solely from the propellers or motors used, depending on the degree of the blade and the diameter of the propeller. Even increasing the motor's KV will change the whole aerodynamics of the quadcopter.

The HR-04 ultrasonic sensor was considered in the initial design for extra collision avoidance, but since the Raspberry Pi has a limited number of ports for information that it can handle from

outside components. Also, the LIDAR is capable of 360° object avoidance, so it was deemed better to have more of the processing for gathering data than object avoidance at this time.

2.5 Realistic Constraints

There are a number of constraints that I have to face throughout this entire project. A restraint that I must incorporate is where and when I can fly the drone. Since I am near an airport and a military base I must contact the navy and ODU whenever I want to fly at ODU. By notifying the public of our testing, I need avoid any confusion with ODU sponsored events, the air traffic, and navy operations. This avoids any confusion and cause for alarm that could complicate further flight testing.

I have to schedule in advance when I want to test to see how our drone is working and to make sure it is kept below 400 feet. Also, if I fly too high, I'll cross into FAA territory and can be picked up on their radar as an unidentified flying object and may be treated as a threat. This could lead to a global incident and I must do my best to avoid reaching these altitudes. I also need to take into consideration the social effects flying our drone may produce. Since the drone will be autonomous this means that I'll be using cameras and sensors to track objects. I need to take in consideration an individual's right to privacy and must ensure I fly the drone in public areas only to avoid any privacy issues. Since I am flying outside, I'll need to be careful to avoid any animals that may be in the sky. If a bird flies into the propeller of the UAV it is very likely to kill or severely harm the animal. In order to avoid this, I've implemented a kill switch so if my visual observer notices that an animal is too close to the UAV I can instantly cut power to the motors.

One other constraint is the synchronization of all the parts. Theoretically, I can be able to simulate and test all sensors and parts on our drone. However, there is a possibility that when it comes down to it I will not be able to synchronize all parts properly. I need to prepare to critique and run multiple tests to get the drone to work properly. I also need to take into consideration the cost of the parts. I was provided some of the necessary parts listed, however, I decided to purchase a Raspberry Pi 3 rather than the version 2 I was provided to ease communication between the computer and the drone via built-in Wifi.

3. Project Considerations

3.1 Engineering Ethics

There are very strict laws that need to be followed when dealing with a drone. These laws are very important and must be considered whenever flying a drone. These laws are put in place by the Federal Aviation Administration (FAA) [8]. These laws are given by the FAA Modernization and Reform Act of 2012 (FMRA) [8]. Some the law I must follow include but are not limited to [8]:

- Faculty teaching aviation-related courses at accredited educational institutions may assist students who are operating a model aircraft under section 336 and in connection with a course that requires such operations, provided the student maintains operational control of the model aircraft such that the faculty member's manipulation of the model aircraft's controls is incidental and secondary to the student's (e.g., the faculty member steps-in to regain control in the event the student begins to lose control, to terminate the flight, etc.).

- A person may operate an unmanned aircraft for hobby or recreation in accordance with section 336 of the FAA Modernization and Reform Act of 2012 (FMRA)[1] at educational institutions and community-sponsored events[2] provided that person is (1) not compensated, or (2) any compensation received is neither directly nor incidentally related to that person's operation of the aircraft at such events;

- A student may conduct model aircraft operations in accordance with section 336 of the FMRA in furtherance of his or her aviation-related education at an accredited educational institution.

Section 336(a) of the FMRA provides special rules for model aircraft that require the aircraft to be:

1) Flown strictly for hobby or recreational use;

2) Operated in accordance with a community-based set of safety guidelines and within the programming of a nationwide community-based organization;

3) Limited to not more than 55 pounds unless otherwise certified through a design, construction, inspection, flight test, and operational safety program administered by a community-based organization;

4) Operated in a manner that does not interfere with and gives way to any manned aircraft; and

5) When flown within 5 miles of an airport, the operator of the aircraft provides the airport operator and the airport air traffic control tower (when an air traffic control facility is located at the airport) with prior notice of the operations (model aircraft operators flying

It is also important to consider the privacy of other individuals as well. There are laws set in place to protect and individuals rights when dealing with drones. According the the Virginia Acts of Assembly - Chapter 18.2-130.1 “It is unlawful for any person to knowingly and intentionally cause an electronic device to enter the property of another to secretly or furtively peep or spy or attempt to peep or spy into or through a window, door, or other aperture of any building, structure, or other enclosure occupied or intended for occupancy as a dwelling, whether or not such building, structure or enclosure is permanently situated or transportable and whether or not such occupancy is permanent or temporary; or to do the same, without just cause, upon property owned by him and leaser or rented to another under circumstances that would violate the occupant’s reasonable expectation or privacy. A violation of this section is a Class 1 misdemeanor. The provisions of this section shall not apply to a lawful criminal investigation [9].” It is important to follow these laws whenever a drone is flown order to avoid any violations of the law.

3.2 Engineering Standards

IEEE 802.11b/ IEEE 802.11g- The 802.11b standard has a maximum raw data rate of 11 Mbit/s, and uses the same media access method defined in the original standard. 802.11g works in the 2.4 GHz band (like 802.11b), but uses the same OFDM based transmission scheme as 802.11a. I must take these standards into consideration since I’m using wi-fi as communication between the drone and our computer to control it. This means I’m not misusing the wi-fi with the software as well as accepting the risks of interference from other devices such as microwaves and other bluetooth devices. I am called to operate within the bandwidth approved by the IEEE committee with the UAV communication through wi-fi to other devices.

IEEE STD 730-2002- “The purpose of this standard is to provide uniform, minimum acceptable requirements for preparation and content of software quality assurance plans. In considering adoption of this standard, regulatory bodies should be aware that specific application of this standard may already be covered by one or more IEEE or ANSI standards documents relating to quality assurance, definitions, or other matters.” This is used to allow the user to know what and how the software should operate under ideal conditions and to inform should an issue arise. I must follow this by documenting any code scripts and programs used by the UAV.

FAA Modernization & Reform- “The legislation also seeks to improve aviation safety and capacity of the national airspace system, provide a framework for integrating new technology safely into our airspace, provide a stable funding system, and advance the implementation of the Next Generation Air Transportation System (NextGen).” This standard deals with aviation safety in which I must consider since I’m flying a drone. Another consideration is for the laws for flying around within a controlled airspace and to be able to execute safe flight in our airspace.

4. Broader Impact

4.1 Economic and Environmental

First used primarily for military usage, UAV's were expensive since many experimental parts were expensive and uncommon. As testing progressed, however, the components used to make UAV's became more common and cheaper. As technology rapidly evolved, it took only ten years for the public to have access to UAV's and expand their uses. Companies, like Amazon, have invested research in using UAV's to deliver packages from warehouses in as little at 30 minutes. Other companies, like Google, use UAV's with cameras to monitor environments [10]. This idea has been implemented by observing the environment and mapping changes that occur throughout the year. UAV's can be sent to monitor weather changes and alert citizens of incoming natural disasters. A notable instance was when the government used drones to monitor Californian forest fires in August of 2016 [11]. Another instance of using UAV's in the environment is to monitor the wildlife. Instead of having game wardens traveling across various terrains, UAV's can cover more land or sea and relay any disruptions. For example, conservationists stationed near Antarctica use UAV's to detect if the whale population is being attacked by Japanese whale hunters [12].

4.2 Health and Safety

UAV's are also applied in maintenance work. Windmills that convert wind energy into electrical energy are inspected by using UAV's. The old process consisted of strapping someone to a safety line to the turbine and having that individual lowered to inspect the unit. With UAV's, a visual feed can be sent to a host computer, and someone can inspect the components without being suspended from the turbine [11].

4.3 Social

UAV's can be found in services provided by society, such as the police, emergency, military, and commercial services. In Britain, the police have turned to using UAV's to investage theft cases as well as high risk operations, such as hostage situations. Currently, law enforcement leaders want to increase the amount of UAV's in their inventory to search for suspected criminals as well as monitor airlines [13]. As for emergency responses, UAV's were used to monitor forest fires in California. This helped reduce the amount of casualties and gave firefighters more information concerning the severity of the fires [11]. Some militaries also expects to increase the amount of UAV's used for reconnaissance missions. Both Britain and the United States are investing and researching more practical ways for UAV's to gather and relay information on targets such as terrorist organizations [13]. Amazon seeks to use UAV's to deliver packages that weigh 5 lbs. or less to residential and business addresses from the warehouses. UAV's would be able to transport packages faster and more reliably than mail personnel due to GPS flight plans.

4.4 Global

Although UAV's have been introduced to the public, most of the applications are still used by the military. These UAV's have surveillance abilities, and some have been modified for raid attacks. This has caused for some countries to move towards creating international agreements between the usage of drones in military matters. Their political arguments are that drones are not sophisticated enough to differentiate between friend or foe. Some argue that drones armed with weapons cause too many civilian casualties to be effective. Only three countries currently have armed drones with weapons: the United States, Israel, and Britain. Even fears of political espionage between rival countries creates a level of uneasiness. Although some claims have little evidence to support them, UAV usage creates a certain level of discomfort between countries. UAV usage laws may have to be enacted to ease the fears of smaller countries [14].

5. Lifelong Learning

Bradley McKee- In this project, I have gained a better understanding on the development of sense and avoid technology. From this project I wish to further my understanding of robotic operating software and various linux distributions. This project has also helped me understand development of software and hardware. From understanding UAV's, I hope to create sense and avoid technology that can be used in everyday application.

6. Project Contributions

Bradley McKee

For this I am tasked with constructing the physical drone and monitoring real time feed. In addition to this, I'm responsible for achieving auto stabilization with scripts that are launched via terminal. All research and development required will be used to build and construct the UAV's functionality to ensure that the hardware and software is communicating. I will be focused on getting data from the sensors that will work hand in hand with the real time feed. Also, I will be responsible for making sure that the sensors are sending data to the rest of the system. I will also be responsible for any housing that is required for the Raspberry Pi and Navio 2. The housing will be developed in AutoCAD and most likely be 3D printed. All research and development required will be used to construct data communication links between the drone and the sensors to perform the described operations required for Autonomous flight with sense and avoid technology.

7. Testing and Analysis

7.1 Camera and Visualization

I tested the killswitch that was provided and successfully got it to work. I then calibrated and tested the RPLIDAR, PX4FLOW, and the HC-SR04, all of which work on an individual basis. I simulated flight stabilization using Ardupilot after launched with the command "sudo systemctl enabled.". According to our results, I should have been able to achieve stable flight with ease. The figure below shows that the RPILIDAR device is working correctly. I was able to successfully deploy the sensor on the Raspberry Pi through open sources SDK's provided by Slamtech Inc. The system is currently running on the Raspberry Pi, but there is no data communication between the sensor and flight controller. Figure 6 shows the functionality of the Raspberry Pi. The picture shows what happens in a perfect room (indoors), it shows the boundaries by creating a list of vector points that the sensor is reading. The vector list that is in this is changing after every single revolution it has.

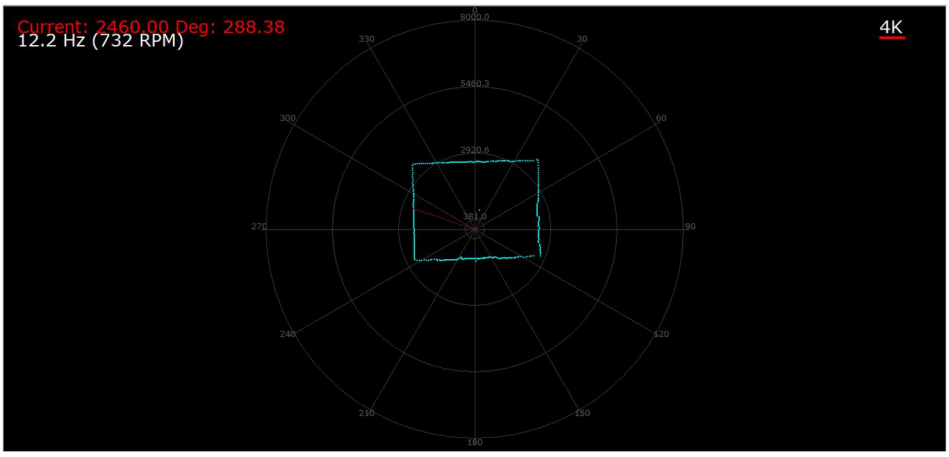
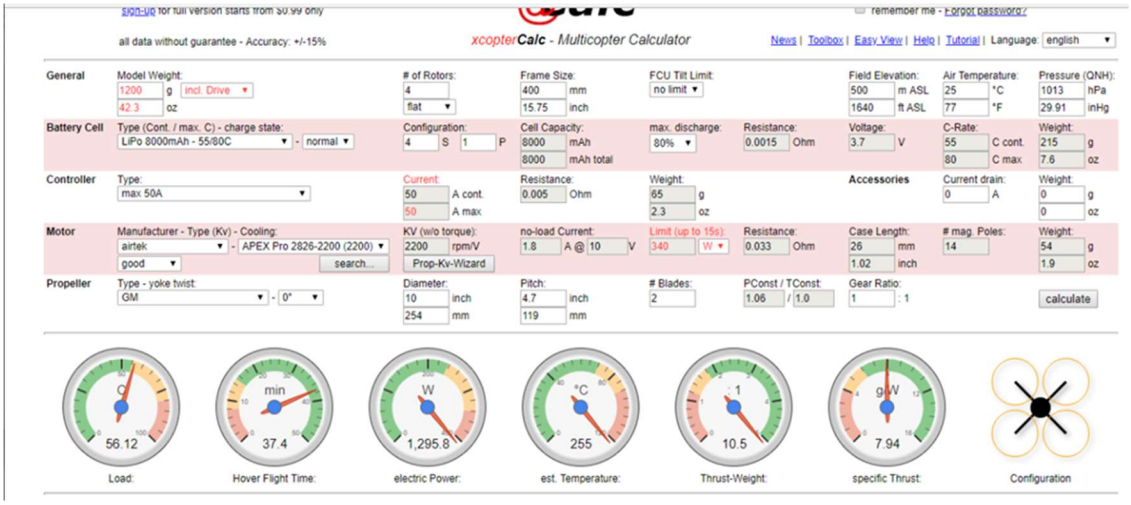


Figure 6. Screen Capture of live feed from 360 Sensor

The RPLIDAR and PX4FLOW tested to be functional and both receive data from the Raspberry Pi. However, due to time constraint I was not able to actually fully implement a feed from the camera or generate a screen capture from the RPLIDAR while in the air. I was able to get the sensors working but unable to implement the object detection portion.

7.2 Battery Life

To understand how long the battery could operate, I calculated the amount of time a standard flight should work. I used online free software known as eCalc™ to estimate how long the battery life should be and



performed a hover test. The image below shows the eCalc™ results:

Figure 7: eCalc™ results for flight time

The eCalc™ results shows that the UAV's battery should support about 37 minutes of hovering without any harm to the battery. After receiving these theoretical results, a real hover test was conducted. The UAV battery was able to hover for approximately 30 minutes before the ground control signaled that it was at a low voltage.

As I began testing the drone, the 2200 mAh battery I was using drained and died, making it unable to recharge. I was provided with a 8000 mAh battery which was usable. Due to time constraint, I was unable to order a new 2200 mAh battery, so I used the 8000 mAh for the rest of our testing. It held up just as well as the other battery.

7.3 Stabilization

I was able to successfully compile a code using python that would stabilize the drone. The code uses drone kit and dronekit SITL prebuilt libraries. The issue is that due to the fact that the compasses were not able to get calibrated the stabilization code did not work for it. I was not able to test to see if the code worked but I was able to see that it did indeed compile.
I attempted to create a proxy to simulate the stabilization code using MAVproxy which creates a proxy drone that connects to Mission Planner and runs your code as though it was connected to an actual drone. Using this it should show the proxy drone flying to the specified locations from the code. However, I was not able to connect the code to MAVProxy and it I could not upload the code to the simulation. I was able to connect the proxy drone to Mission Planner and I was able to open the ports for the MAVProxy drone to connect to but I was experiencing issues with MAVProxy code.

8. Experimental Results

The results of the analysis shows that the UAV's sensors are able to communicate with the Raspberry Pi, and the flight stabilization scripts that were uploaded to the Raspberry Pi. The UAV is able to leave from a starting point and would not fall towards the ground if the motors were rotating at a different speed. The PX4FLOW camera was also attached to the UAV and receives data that can be seen by the Raspberry Pi.

Due to time constraints, the functions that were not completed were the object avoidance from multiple directions with either the RPLIDAR or the HC-SR04. Neither method was implemented onto the UAV. Some scripts were tested and added to the Raspberry Pi, but a final version was not completed for the design. Also, an autonomous flight script was added to the Raspberry Pi, and the testing showed that the PX4FLOW camera was able to send signals to the Raspberry Pi but not able to send a video image to a ground station. The UAV can still be controlled by a remote transmitter.

These results show that more time was needed to accomplish all requirements of the competition's requirements.

References

[1] "International Aerial Robotics Competition," International Aerial Robotics Competition, 2016. [Online]. Available at: http://www.aerialroboticscompetition.org/index.php. [Accessed 14 Oct. 2017].

[2] "Emlid," 2016. [Online]. Available at: https://docs.emlid.com/navio2/. [Accessed 27 Oct. 2017].

[3] "Slamtec"2016. [Online]. Available at: http://bucket.download.slamtec.com/a7a9b856b9f8e57aad717da50a2878d5d021e85f/LM204_SLAMTEC_rplidarkit_usermanual_A2M4_v1.1_en.pdf [Accessed 27 Oct. 2017].

[4] "PixHawk". [Online]. Available at: https://pixhawk.org/modules/px4flow [Accessed 27 Oct. 2017].

[5] "Battery Basics" MIT. [Online]. Available at: http://web.mit.edu/evt/summary_battery_specifications.pdf. [Accessed 27 Oct 2017].

[6] E. Eekhoff, "BRUSHLESS MOTOR KV CONSTANT EXPLAINED," LearningRC, [Online]. Available at: http://learningrc.com/motor-kv/. [Accessed 27 Oct. 2017].

[7] Ieee.org. (2017). *IEEE IEEE Code of Ethics*. [online] Available at: https://www.ieee.org/about/corporate/governance/p7-8.html [Accessed 27 Oct. 2017].

[8] Anon, (2017). [online] Available at: https://www.faa.gov/uas/resources/uas_regulations_policy/media/interpretation-educational-use-of-uas.pdf [Accessed 27 Oct. 2017].

[9] Dronethusiast. (2017). *Drone Laws in Virginia*. [online] Available at: https://www.dronethusiast.com/drone-laws-virginia/ [Accessed 27 Oct. 2017].

[10] Nath, Trevor. *How Drones Are Changing The Business World*. [online] Available at: https://www.investopedia.com/articles/investing/010615/how-drones-are-changing-business-world.asp [Accessed 30 Nov. 2017].

[11] Edie.net. (16 August 2017). *Five ways drones are being used to help the environment*. [online] Available at: https://www.edie.net/news/8/Drone-technology-environmental-sustainability-impact-for-the-UK/ [Accessed 30 Nov. 2017].

[12] Environtech-online.com. (2015) *How is Drone Technology Affecting Environmental Monitoring?* Available at: https://www.envirotech-online.com/news/environmental-

laboratory/7/breaking-news/how-is-drone-technology-affecting-environmental-monitoring/35247 [Accessed 30 Nov. 2017].

[13] Eysenck, Juliet. (25 May 2016). *How drones are changing our lives: the good, the bad and the lazy*. Available at: http://www.telegraph.co.uk/technology/2016/05/23/how-drone-technology-is-changing-our-lives-the-good-the-bad-and/ [Accessed 30 Nov. 2017].

[14] Eyal, Jonathan. (27 Oct. 2014). *Analysis: Drones Posing Global Security Issues*. Available at: http://www.govtech.com/products/Analysis-Drones-Posing-Global-Security-Issues.html [Accessed 30 Nov. 2017].

Appendix