Drone Stabilization System Based on Kalman and Proportional-Integral-Derivative algorithms

---

A Thesis Presented to
The Faculty of College of Computer Studies
Foundation University, Dumaguete City
Philippines

In partial Fulfillment
of the requirements for the Degree
BACHELOR OF SCIENCE IN COMPUTER SCIENCE

By
Jamie Joe M. Van Stone
Jassam Alinsub
Jeffrey Tubog

March 2017

APPROVAL SHEET

This thesis project entitled Drone Stabilization System based on Kalman and Proportional-Integral-Derivative Algorithm, prepared and submitted by Jamie Joe M. Vanstone, Jassam V. Alinsub, and Jeffrey Tubog in partial fulfillment of the requirements for the degree of Bachelor of Science of Computer Science, is hereby accepted and recommended for Oral Examination.

JOHN AUDI P. BATO
Member

ALBERTO C. DIEGO JR.
Member

JEFFREY M. RIVERA
Member

ACCEPTED as fulfillment of the requirements for the degree of Bachelor of Science of Computer Science.

MARK JAMES KHO
OIC, CCS

APPROVED by the panel at the Oral Examination with the grade of **PASSED.**

ALBERTO C. DIEGO JR.
Member

MARK JAMES KHO
OIC, CCS

JEFFREY M. RIVERA
Member

ACKNOWLEDGEMENT

The researchers would like to give thanks to all those to helped in this research. Most of all to the Heavenly father for the guidance, knowledge and motivation to make this research possible.

The researchers would also like to thank the faculty and staff of the College of Computer Studies for providing valuable inputs and insights that helped make this research. Especially to Mr. Mark James Kho, Mr. Jeffrey River and Mr. Alberto Diego Jr., who guided us as our panel throughout the development of this research.

To our families who supported us through the trials and struggles. Who kept us going through the most difficult times and encouraged us even when we lose hope.

To our friends and classmates that also collaborated with their ideas and knowledge which help in times when the researchers were stuck. Also their encouragement which motivated the researchers to pursue and aim to complete this research.

ABSTRACT

A multi rotor copter is a rotorcraft with more than two rotors. A typical multi rotor copter has around 4 rotors. These devices are mainly controlled by a radio controller or a wifi

controller. They are supported by a flight controller which handled most of the balancing and output for the commands from controllers. Flight controllers are a powerful piece of technology since they are able to compute data quickly and provide a precise response which in return greatly improve the experience of flying the device. Most flight controller include a gyroscope and accelerometer to gather data about the devices orientation during flight which then would be computed to an output for the rotors. The same can be achieved using other components such as a micro controller and an Inertial Measurement Unit. This study aims to provide a learning basis to develop flight controllers using available components such as a GizDuino microcontroller, GizDuino Bluetooth Shield, an Sparkfun IMU, an android device and a quad copter. The results show that it may be possible to develop a flight controller with such components however the accuracy results have come out below satisfactory due to the time constraints to conduct tests.

Keywords: copter, drone, gizduino, android, bluetooth, flight controller, IMU

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I
Rationale

Technology brings new solutions and efficient ways to complete tasks. Technology improves everything from farming to space travel. The evolution of technology is even greater from land and now to space. One of the recent invention that changes the world we know is drone. Even though drones were invented for military purposes, many people are beginning to understand the true potential of these incredible devices.

Drones are flexible. With different sensors and camera attached, it has do a broad range of applications that industries will interest on it. Some applications includes logistics, surveillance, tracking, planning, military operations and recreational use. There are even some companies that are planning to use drones as means of logistics, which means that less vehicles would be needed to deliver packages thus reducing traffic levels as stated here "UAVs could provide major relief for inner cities by taking traffic off the roads and into the skies. So far, payloads are limited but a network of UAVs could nevertheless support first and last-mile logistics networks (Drones in logistics – internet of things, unmanned cargo, uav delivery. (n.d.))."

Drone has an advantage in surveying a infrastructure, site and layout planning in constructions and surveying a forestry. It can be operated in extreme weather conditions and in geographically challenging locations without putting personnel at risk. It can follow a preprogrammed flight path, and fly closer to both the infrastructure and the ground. This allows for highly detailed flight plans, higher measurement accuracy, and increased

repeatability (Drones in logistics – internet of things, unmanned cargo, uav delivery. (n.d.)). With drones surveying or planning is easy and not more hustle. It can gathered a reliable data that may used.

Drone can also be used in emergencies, tracking and photography. Equipped with sensors and camera, drone can take pictures, track animals or even used in search and rescue operations. In Fukushima, a radioactivity leakage from damage of reactors which result of high risk of radioactive contamination. Drones are used in surveillance allowing them to keep human in danger zone or limit their risk exposure.

In the Philippines, drones are already use in means of search and rescue operations. A project named “Kwago” which “marks an area once a thermal scan has been detected. A low altitude sweep will follow to get the exact location of the coordinates of the detected heat signature and also to get more detailed information of the subject (News, A. (2016)).” Project “Kwago” uses an automated flight system which uses the capabilities of a global positioning system to track and guide the drone to certain location assigned by the user which could greatly reduce the time and effort required to find injured, trapped or missing personnel and animals in disaster events.

With the use of drones different things will do. It is a start up for new innovative invention that will help the society to be more productive. It makes life more easier and yet

more fun. It is an invention that change the way we live and open opportunities to a better world. In a year from now, people will leave in sky.

Theoretical Background

The theoretical components in this system are a combination of the Kalman filtering and Proportional-Integral-Derivative algorithms applied to the researchers theories and algorithms. These have been used in most robotic applications. Combining these techniques along with some custom programming and tweaking may achieve a reliable balancing algorithm for hardware that is more cheaper and available. The Kalman filter is essentially a set of mathematical equations that implement a predictor-corrector type estimator that is optimal in the sense that it minimizes the estimated error covariance — when some presumed conditions are met (Welch, G., & Bishop, G. (n.d.)). Kalman filter is often used in tracking and navigation applications, where it is used to reduce the noise from sensors like in the use of a radar system to track to position of aircraft in flight. With the Kalman filter, we are able to reduce or filter most errors and noise in the sensory data received from the Inertial Measurement Unit and make calculations more reliable. The data can then be used as crucial information about the current orientation and situation of the copter that can be applied to further calculations to produce the expected results.

The Proportional-Integral-Derivative (PID) loop is used in this system to generate the optimal speed output for the motor compensation when the copter goes off balance. This process consists of three components, the proportional response, Integral response

and the derivative response. The PID processing provides a good output value for the motors to reduce over compensation and provide more stability for the copter in flight, however tuning and finding the right values for each component of the PID algorithm takes a lot of time and tedious experimentation to find the correct values for the best results.

The Kalman filtering technique and PID processing algorithm can be apply in the stability of drone. The errors or the noise that can be a hindrance of being unstable are being eliminated. Thus, providing a more efficient and accurate results. With the use of Kalman filtering technique and PID processing algorithm, it is possible create a drone that is stable. Combining these techniques, theories and algorithms, it can developed a fully stable drone that may used in many purposes.

Problem Statement

Drone is widely used today. The NPD group's research suggests that users tend to be more seasoned drone flyers, with 56% of people buying drones costing $500 (Php 25147.50) or more. $500 is the cheapest drone for the professional-grade drones which is more expensive. The professional-grade drones is a type of drone that can carry a heavy load. A commercial drones come in the largest sizes. They can range from four to eight blades, depending on the rig’s carrying capacity.

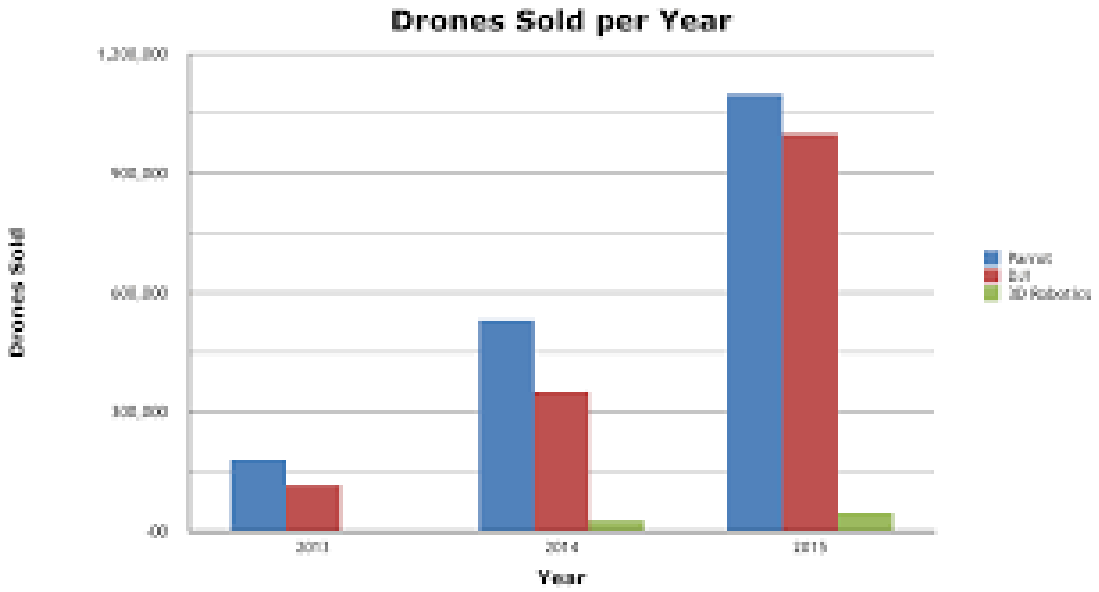


Figure 1.1 Drone Sold per Year

As the table in figure 1.1 shows from the year 2013 to 2015, the number of drones sold are rapidly increasing. Year after a year, the number of people buying drones are rapidly increasing. It implies that there will be well over 1 million privately owned consumer drones taking off by the end of the year (Amato, A., Says, A., Says, C., Says, M., Says, K., & Says, R. (2015, April 16)).

Even DIY drones can be expensive, one of the most expensive components, and most complicated, is a flight controller. This can be improvised by developing a flight controller using a micro controller and an IMU but there are few practices of this and little documents in developing flight controllers. In order attain the objectives the following questions must be answered:

1. How to develop a drone that is low cost?

2. What are the variables required for multi-rotor stabilization system?

3. What is the stability/accuracy rate of the system?

Significance of the study

Developing a research about this topic would be most useful to explore a more cost efficient way to solve copter stabilization. This research would also open more possibilities for further research to enhance and improve current stabilization algorithms. This research would be best used by students or enthusiasts who wish to explore the development of a quadcopter. This research aims to add information about copter flight controllers and stabilization systems as well as the experimentation of an android - bluetooth controller for copters.

The research and development of this topic could provide useful information, data and knowledge in developing quadcopter. Researchers used a low cost materials in developing a quadcopter. The total cost is lessen compare to the commercial drones (See Chapter II). This research helps students or enthusiasts develop their own homemade quadcopter that is low cost.

Scope

The focus of this study is to develop an algorithm that allows a quadcopter to fly stable with the cheaper and more available equipment and if it is possible to use an android-bluetooth controller application for sending commands to a copter. Developing an algorithm which will be installed on an arduino or gizduino microcontroller. The researchers will focus on the data collected from the Inertial Measurement Unit to find the

current position of the device within a given space. In this study the he researchers will only explore on the following: Open source technology, controlled environments, acceleration, direction and altitude. The algorithm will solve this problem by interpreting data collected.

1. By collecting data (such as direction of the device, determine speed the device, orientation of the object, altitude of the device) from different sensors.
2. By adding IMU (Inertial Measuring Unit) the Drone will be able to determine its orientation, altitude, direction and speed.

Limitations

The researchers are anticipating possible problems that might hinder the objectives of the study. The possible problems that may encounter is inconsistency of variable data. Inconsistency of data from sensors could develop problems in the algorithms for finding the position of the device in the given space. Due to the instability of the quadcopter, the data from sensor is affected and may lead to inconsistency of data. To hinder the possible problem, the researchers uses a kalman filtering technique to filter the data.

Another possible problem is inaccuracy of kalman filtering technique. If the data cannot be properly filtered with the kalman filtering technique then the inaccuracy of the system would increase. To hinder the possible problem, the researcher find another technique that would provide more reliable data.

Uncontrolled environments are another possible problem. Random factors from the environment could affect the accuracy of the system like wind and moving object. To hinder the possible issue, the researcher set the testing areas to be a controlled environment.

CHAPTER II
Review of Related Literature

A project of Filipino design and concept arose where drones could be used effectively in disaster, search and rescue operations, and the project named “Kwago” which “marks an area once a thermal scan has been detected. A low altitude sweep will follow to get the exact location of the coordinates of the detected heat signature and also to get more detailed information of the subject (News, A. (2016)).” Project “Kwago” uses an automated flight system which uses the capabilities of a global positioning system to track and guide the drone to certain location assigned by the user which could greatly reduce the time and effort required to find injured, trapped or missing personnel and animals in disaster events.

More and more drone applications are emerging which uses some sort of automated flight or autopilot system however there are still some limitations with drone automated flight, According to an article that can be found at dronetrest.com “No single sensor is good enough to control your drone, that is why several sensors are used. By combining the measurements from all the sensors on your drone, and applying some fancy math (Kalman filtering) the autopilot can keep your aircraft stable very effectively (Beginner's guide to drone autopilots and how they work. (n.d.)).” Also drones are incapable of operating autonomously indoors or in areas where a strong GPS signal cannot be acquired. Drones that are capable of operating in these areas could greatly increase the usages of these

devices in many more applications such as being able to autonomously map the structure of a building or provide roaming surveillance within buildings.

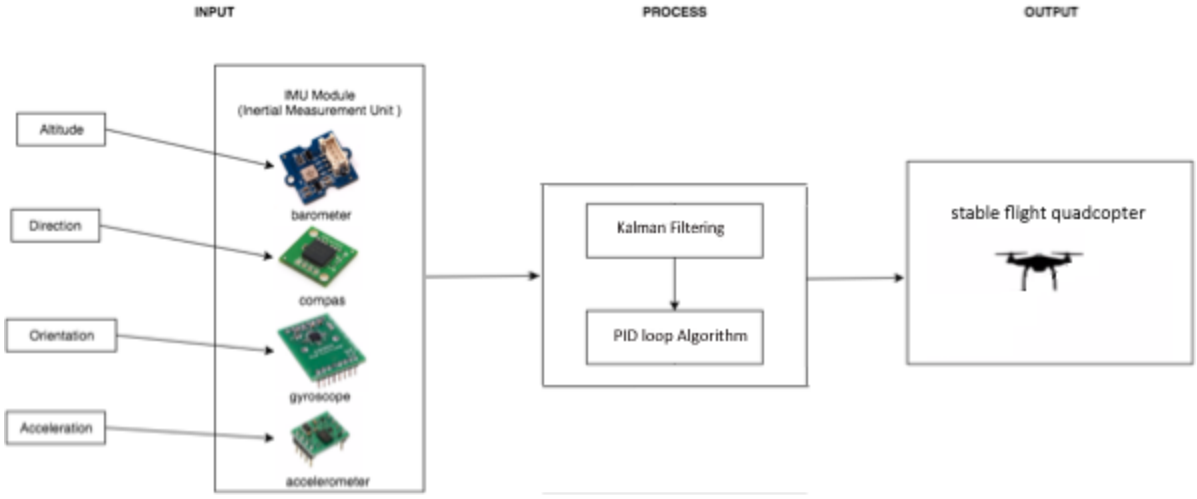
Global Positioning Systems require a stable connection to at least 4 satellites orbiting earth to provide accurate data; these connections may be obstructed by many physical barriers. Barriers like walls, trees and even bad weather can obstruct signals. Being in areas such as valleys or being surrounded by cliffs can even disrupt the signals. Global Positioning Systems cannot operate in subterranean environments, such as caves, tunnels, or even under the surface of the water. This means that we cannot always rely on these external references for navigation and should have an assist or backup system. An alternative which Aircraft, ships, land vehicles, and spacecraft have been using is an internal system used for navigation assist which is called Inertial Navigation which utilizes data gathered from internal sensors, such as gyroscopes and accelerometers, and a method called “Dead reckoning” to calculate their position relative to a given point.

This can be achieved by calculating the velocity, changes in directions and time travelled to provide data about the position of the device. Dead-Reckoning can be used as part of a method to develop a system that would enable autonomous flight of drones indoor since this method could use the information gathered from sensors within the device to calculate the position, speed and orientation from a known starting point (Welch, G., & Bishop, G. (n.d.)). However there are small variations in the data gathered from this system which over time can accumulate to greater inaccuracy, due to the inaccuracies both GPS and

INS cannot be completely reliable unless they are merged together which is what most vehicles, aircraft, ships and spacecraft use today.

Since drones have relatively short flight times and use indoors or in areas with minimal external interference such as wind, an autonomous inertial navigation system for a drone may be possible in these areas. These systems use readily available flight controllers that can still be costly. Providing a cheaper way for one to develop a DIY quadcopter and providing an algorithm for the flight system may be useful to those who wish to develop a quadcopter that would cater to specific needs.

Methodology



This system will rely heavily on the use of a gyroscope and an accelerometer sensor to determine the current orientation and movement of the copter. The system will be able to use the gathered data to produce a speed output for the motors to achieve balanced and stable flight. The basic steps for the system are 1. Acquire data, 2. Filter data, 3. Compute

data, and finally 4. Produce an output. To test the accuracy of the results we will test how stable the copter is able to fly over a short duration.

Data gathering from the sensors is simple enough by connecting the the IMU module to the micro controller and applying it to the copter platform. However when the IMU is mounted to the copter its is best to keep it as parallel to the ground as possible to provide better results. Once the data has been gathered the system put the data through the kalman filtering algorithm to produce reliable data for computation later on. The kalman filter will generate two results, one for the actual sensor data and another for the predicted data by the algorithm and chooses the value closest to the two data sets.

The filtered data goes through a proportional–integral–derivative or PID loop algorithm to continuously calculates an error value as the difference between a desired setpoint and a measured process variable and applies a correction based on proportional, integral, and derivative terms (sometimes denoted P, I, and D respectively) which give their name to the controller type (PID controller. (2017, February 24)). This means that the return value from this loop should be applied as the output value to the motors to achieve the desired output, in this case a stable flight where the orientation of the drone should be level.

The return values of the PID loop is then adjusted and calculated to be applied as an output value for the ESC to control the speed of the copters motors. Once everything has been applied, testing and data gathering can begin.

Budget Plan

Components	Option	
	1	2
Microcontroller	SparkFun Redboard **Php 823.45**	Arduino Mega **Php 780.00**
Bluetooth Module	Gizduino Bluetooth Shield Master **Php 935.00**	Smart Bluetooth Shield **Php 827.08**
Inertial Measurement Unit	Sparkfun mpu 9250 IMU Sensor Board Module **Php 1212.00**	
Battery	2200mAh 2S 20C Lipo **Php 410.78**	
Quadcopter Frame	GAUI 500x Frame **Php 3,995.99**	
Electronic Speed Controller	GAUI GE-183 18A ESC x4 **Php 1,934.57**	
Motors	GAUI GM-412S BRUSHLESS MOTOR 960KV **Php 2,794.57**	
Propellers	GAUI 500X 8 IN. PROPS 8A AND 8B x 4 **Php 401.19**	
TOTAL	**Php 12,507.55**	**Php 12,356.18**

Table 2.1 Budget Plan

The proposed budget for this research is based on availability and specifications of the required components. Some of the components were ruled out due to price or specifications and replaced with more applicable components. As seen in table 2.1 a selection of components are available and the cost of the project can be reviewed.

Conceptual Design

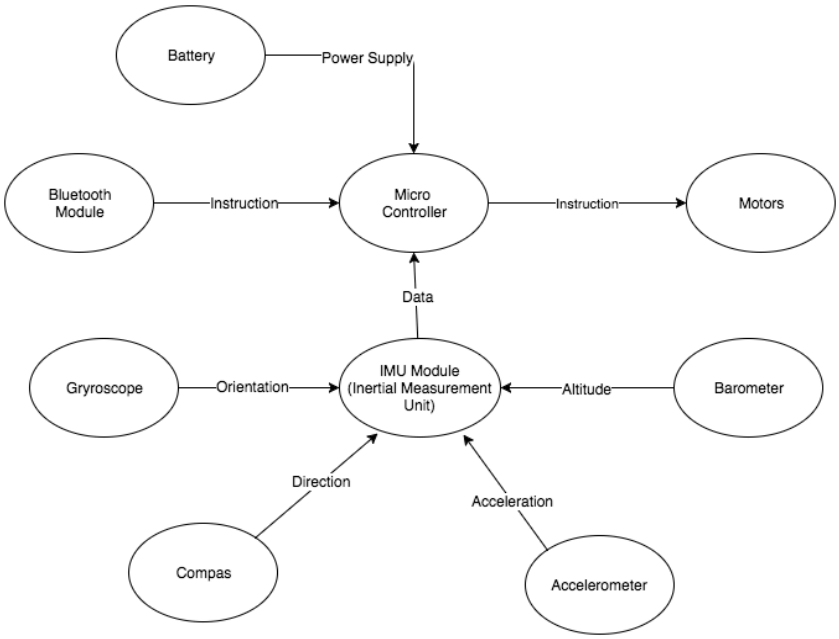


Figure 2.1 Conceptual Design

The conceptual design of the project required minimal hardware components since the weight and power supply capacity is to be considered to develop an efficient system. As seen in figure 2.1 there is a simple but efficient layout of the components and the way the data flow throughout the system. The system is compact and light enough for the application.

Use Case Diagram

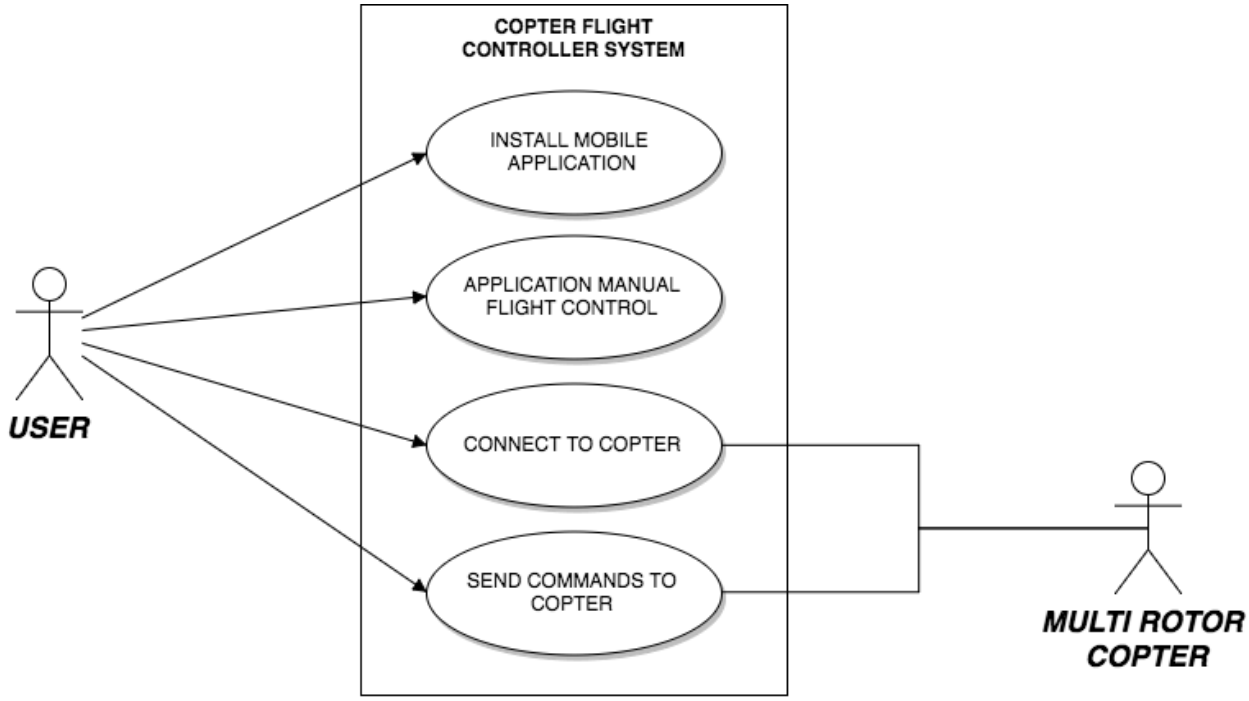


Figure 2.2 Use Case Diagram

Figure 2.2 show the user may interact with the system via the mobile app. The user is able to install the application, select the manual flight control option in the application, connect to the device via bluetooth and finally send the commands to the device. The device only connects to the bluetooth controller application and receives the commands from the application.

Use Case Narrative

Use Case	Install Mobile Application
Actors	User
Purpose	This enables the user to install the mobile application on an android smartphone.
Overview	The application will be installed on the smartphone which is used

	by the user to connect and send command signals to the device via bluetooth.
Type	Essential
Precondition	None
Post-condition	Application has been installed.
Flow of events	
Actor Action	System Response
1. Run Installer	1. Install Application 2. Application Installed Successfully

Table 2.2 Install Application

Table 2.2 is the use case narrative for the installation of the mobile application on an android smartphone. This use case is meant to allow the user to use the mobile application. The installation of the application involves uploading the application using the Android Studio IDE which will compile and install the application on the user's smartphone.

Use Case	Application Manual Flight Control
Actors	User
Purpose	This use case allows the user to select the manual control flight feature of the application.
Overview	The user may select the option to direct the application to display the main controller for the device.
Type	Essential
Precondition	Application has been installed
Post-condition	Application will display main flight controller.
Flow of Events	
Actor Action	System Response

1. Open application 2. Select manual flight feature	1. Application opens 2. Displays main flight controller

Table 2.3 Application Manual Flight Control

Table 2.3 describes the Manual flight control use case narrative which is used when the user selects the manual flight option from the initial interface of the mobile application. This tells the application that the user will be manually sending flight commands to the device and initializes the application for that operation.

Use Case	Connect to Copter
Actors	User and Robot
Purpose	This use case is used to establish bluetooth communication between the android smartphone and the bluetooth shield on the copter.
Overview	When the user and the copter is ready to establish a bluetooth connection the user then presses the connect button which then the system begins searching for the copter and once found, establishes a bluetooth connection.
Type	Essential

Precondition	Application is installed and bluetooth is turned on.	
Post-condition	Bluetooth connectivity is established.	
Flow of events		
Actor Action		System Response
1. Connect to Copter		1. Begin bluetooth scan 2. Once found initialize pairing 3. Once paired begin comms thread.

Table 2.4 Connect to Copter

Table 2.4 describes the connect to copter use case which is the initialization of the bluetooth communication from the android smartphone controller application and the bluetooth shield which is connected to the microcontroller on the copter. If the bluetooth is not turned on the application will prompt the user to activate bluetooth on their device. Once bluetooth is activated scanning and pairing may commence on the two devices.

Use Case	Send Commands to Copter
Actors	User and Robot
Purpose	This use case handles the main communications between the android smartphone controller and the bluetooth shield on the copter.
Overview	The application continuously sends commands over the bluetooth communication connection using a thread. The values that are send are updated when the user makes changes in the joysticks or trims.

Type	Essential	
Precondition	Application is installed and bluetooth is turned on.	
Post-condition	Commands are sent and copter responds.	
Flow of Events		
Actor Action		System Response
1. Input Commands		1. Mobile Application collects command data and converts it to JSON String 2. Send JSON String via bluetooth to copter 3. Microcontroller processes and generates output

Table 2.5 Send Commands

Table 2.5 describes the send commands use case where the user inputs commands that are sent and interpreted by the copter. The application uses a lightweight thread that continuously sends command data to the copter which provides an efficient and quick communication that generates optimal responses by the devices. The thread also ensures that the bluetooth connection is never idle and never interrupted unless out of range or one device fails.

CHAPTER III
Gizduino atmega644+

The microcontroller in this research is an GizDuino 644 ATmega. This microcontroller is compact, and powerful enough to support our proposed algorithms and calculations to reliably handle the processing of the system. The arduino megas 54 Digital and Input/Output and 16 analog Input pins provide enough slots to handle the necessary components for a quadcopter and the system itself. 128 KB, which 4 KB is allocated for the

bootloader, this provides some room for our system codes, however an external storage may be required if the code becomes too large. The arduino mega microcontroller operates at 5v which is optimal for us to use with a quadcopter platform, to an estimate of at least 10 mins flight time with a 2S200mAh Lithium Polymer battery as the power supply.

It is a Filipino version of the Arduino Microcontroller. It has a speed grade of ATmega644 0 - 10 MHz @ 2.7V - 5.5V, 0 - 20MHz @ 4.5V - 5.5V. It has a 32 Programmable I/O Lines and 40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF. It also execute up to 20 MIPS Throughput at 20MHz. It has a write or erase cycles up to 10,000 Flash/100,000 EEPROM and has a 64 Kbytes of In-System Self-programmable Flash program memory. This microcontroller is used to collect instructions from the android controller using the bluetooth module. The data is manipulated and relay the commands to the esc to control the rotor of the quadcopter. Using the Arduino IDE, it enables us to upload and communicate with the microcontroller.

Gizduino bluetooth shield v1.1

This shield receives the data from the android controller and gives the data to the microcontroller. It bridges the android application and microcontroller to communicate. The maximum range signals can travel is 9 meters. This module is capable of operating using an external power supply that ranges from 8v to 12v and can also use an internal supply at 5v. A master or slave switch is also available but for this project master option will be used to send and receive data from the quad copter for debugging and analysis.

Wiring Schematics

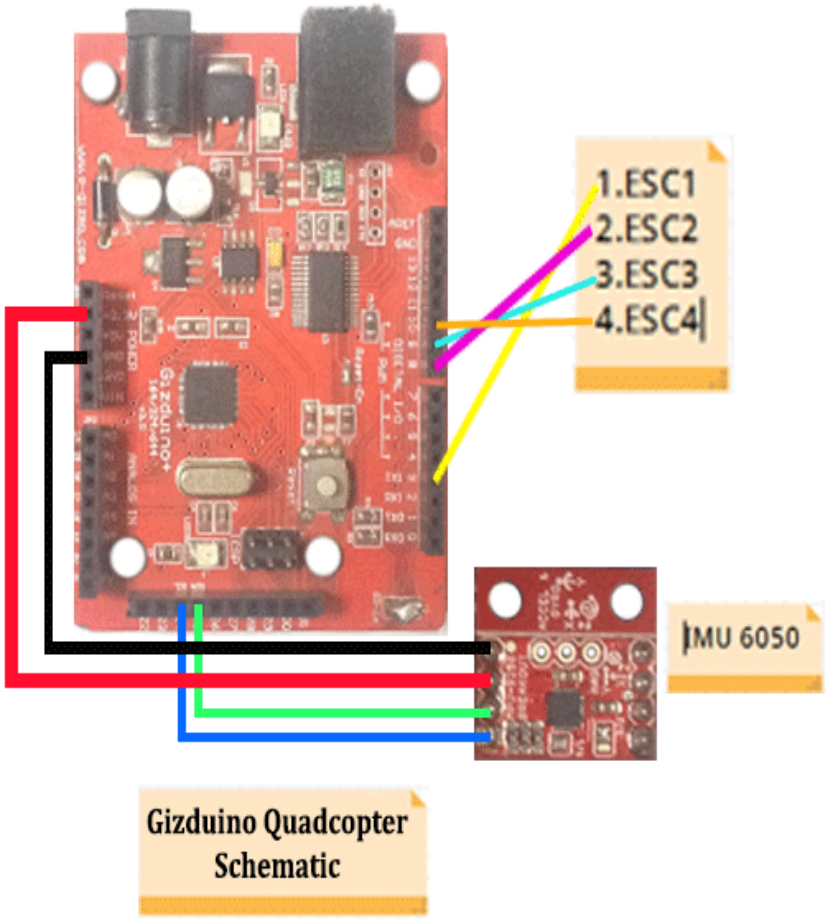


Figure 3.1 Wiring

Figure 3.1 illustrates the wiring schematics for the Bluetooth shield or Microcontroller Unit (MCU) , Inertial Measurement Unit (IMU) and the Electronic Speed Controllers (ESCs). The Ground (Black) wire for the IMU should connect to one of the ground pins on the MCU. The power (Red) wire for the IMU should connect to the 3.3v pin

on the MCU. While the SDA (Data) and SCL (Clock) for the IMU should connect to the respective pins on the MCU with the same labels. The ESC wires are connected in accordance to the PWM requirements for the ESC components. The ESCs should be connected to the PWM pins on the MCU.

Inertial Measurement Unit

The IMU is a module in electronics which collects angular velocity and linear acceleration data. The IMU contains two separate sensors. The first sensor is the accelerometer triad. It produces three analog signals that describes the accelerations in its axes generated by the vehicle. Because of the physical limitations, the most significant acceleration data is caused by gravity. The second sensor is the angular rate sensor triad. Like the first sensor, it also generates three analog signals. It describes the angular rate of each axes. Although the IMU in not at the center of the vehicle, the linear or angular accelerations does not affect the angular rate measurement. The accelerometer triad and the angular rate sensor are attached in such a way that their coordinate axes are not aligned.

Another major component is an Inertial Measurement Unit or IMU. The researchers preferred to use the Arduino MPU-9250. This MPU is composed of, accelerometer, gyroscope, and compass. These sensors provide necessary data for us to calculate the approximate position of the device from a known point. This sensor is compact and light. Applying Kalman filtering with these sensors would provide reliable data for our algorithms. An alternative to the Arduino MPU-9250 would be the AltIMU-10 which comes

with the same sensor but has an altimeter built in to provide altitude data of the device. This sensor has the dimensions of 1.0″ × 0.5″ which is very compact and light.

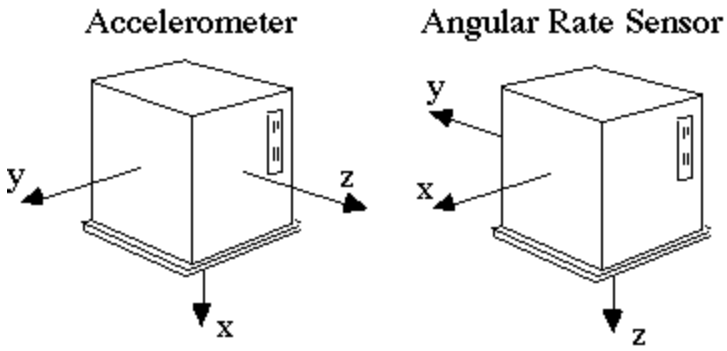


Figure 3.2 Initial IMU sensor coordinate axes

The accelerometer triad was manufactured using a left handed coordinate system. The transformation algorithm first uses (Figure 3.3) to align the coordinate axes of the two sensors.

$$x_{comp} = -x$$
$$y_{comp} = y$$
$$z_{comp} = z$$
$$\omega_{x_{comp}} = -\omega_x$$
$$\omega_{y_{comp}} = \omega_y$$
$$\omega_{z_{comp}} = -\omega_z$$

Figure 3.3 Left handed coordinate system

These simple transformations align the axes of the two sensors so that they appear axes then appear as shown in Figure 3.2.

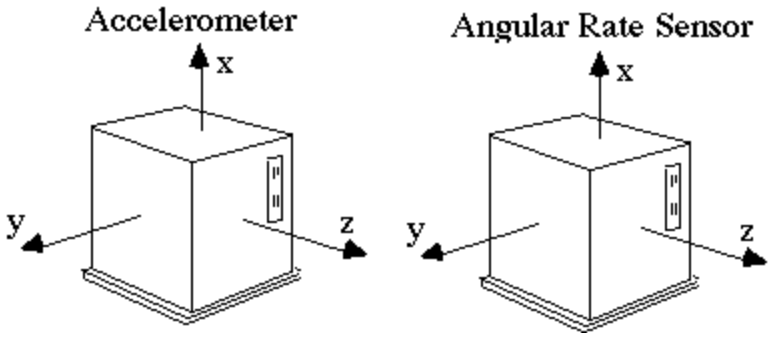


Figure 3.4 Transformed IMU sensor coordinate axes

Quadcopter Frame & Propellers

GAUI 500X is the newest edition of UAV models by GAUI, it follows the success of its smaller version GAUI 330X. This Frame is sturdy enough to endure the research and testing. This frame is also lightweight and can be modified to accommodate the hardware for this research.

Figure 3.5 Foldable frame design

The GAUI foldable frame as seen in figure 3.5 is very useful for transportation. It allows the frame to be encased in a small container or box and secured while being transferred from test sites or development sites.

Figure 3.6 U shaped position locking

The U shaped position locking for easy installation of boom and preventing twist by motor. This design secures the motors to the frame and helps keep them in a certain angle. These position locking is needed for the rotate features of the copter since the system relies on the motors to be set a specified angle of about 5 degrees off center.

Figure 3.7 Stainless standard landing gear spring.

The landing gear helps soften harder landing when the drone fails, goes out of control or power supply runs low. These stainless steel landing gear springs are durable and just flexible enough while being lightweight.

Figure 3.8 Stylish design low wind resistance canopy

The canopy secures the system hardware and components from external events. This ensures the components are safe from physical contact and some weather effects on the copter while in flight.

Figure 3.9 Fast installation/removal system

Fast installation/removal system platform with anti-vibration design for easy maintenance. This frame component improved research efficiency due to the easy removal of the micro controller for development since it is easy to detach and reattach to the frame. There are also rubber dampeners on the legs of the component which reduces vibrations to the Inertial Measurement Unit and its gyroscope.

Figure 3.11 GE-183 ESC 18A

The Electronic Speed Controller is the bridge between the micro controller and the brushless motors. Here the speed signal is received from the micro controller and converted to an electric current which is sent to the brushless motors. The ESC has a continuous current of 18A. There are also a burst current about every 10s of 22A. A BEC is basically a step down voltage regulator. It will take your main battery voltage (e.g. 11.1 Volts) and reduce it down to ~5 Volts to safely power your receiver and servos. The BEC mode is Linear, while BEC output is 5V/2A with lipo cell count of 2-3. The ESC is also user programmable. While the weight is 21g.

Figure 3.12 GM-415 Brushless Motor (800KV)

The brushless motor is outputs the data sends by ESC. It has a motor wire up to 200mm. A 800 number of revolutions per minute that the motor will turn. When their is no load, it needs a 11.1 volts and a current lesser or equal to 0.8A. The revolution per minute

is greater or equal to 9000 RPM. When their is a load, needs a 14.8 volts and a current of lesser or equal to 10A. The revolution per minute is greater or equal to 9300 RPM.

Figure 3.13 10 in. Propellers

These 10 inch propellers are used in this research. The pitch angle is 4.5 degrees and diameter is 25.5cm. The weight is approximately is 10.4g (each). These propellers are advantageous in several ways. These propellers are able to produce more lift thus increasing efficiency and increasing responsiveness and flight time. They also have excellent wind resistance which also improves efficiency and stability. Using these propellers also allows us to add more components if necessary due to the higher load capacity generated from the extra lift.

Power supply

We used two 4000mah 3.7v li-ion rechargeable batteries in series configuration to power the system. However once the voltage goes below 8.0v the behaviour of the system becomes unstable and cannot sustain flight. Aside from the ESC modules becoming unstable, the bluetooth module tends to lose power or disconnect from the paired android controller. We tried adding two more batteries in series and paralleled with the initial

batteries to increase the lifetime of the batteries but still only allows ~5 minute flight time. In order to improve the system a 2S2000 Li-poly battery (depending on the total weight) would be ideal for use in this project since it would provide more stable and better current to the system and flight time would increase.

* The payload calculations and the battery options :

Ex 1: If the 500X carries a HD Camera (weight 350g) and uses a 3S/2000mah LiPo(weight 150g), the flying weight is 1170g (670+350+150).The power consumption in this condition is 141w (as table above). The Power of this battery is 11.1(V) * 2(A) * 60(Min.) = 1332.The flying time is about 7.1 Minute Calculation : (1332 / 141) * 75%(Bat. Factor) = 7.1 (Min.) Ex 2: If the 500X uses 2pcs of 3S/2000mah LiPo(in parallel), the flying weight is 1050g (670+350+150X2). The power consumption in this condition is 167w (as table above).The total Power of the batteries is 11.1(V) * 2(A) * 60(Min.) * 2 (Bat. in parallel)= 2662. The flying time is about 12(Min.) Calculation : (2662 / 167) * 75%(Bat. Factor) = 12 (Min.)"

The researchers choose to use Li-poly battery pack to attain best flight performance. The researchers were able to acquire two Lithium Polymer power supplies one with a capacity of 4000mAh and another of 5000mAh. These power supplies greatly improved the performance and responsiveness of the copter during flight. They also greatly improved the flight time of the copter from around 5 minutes to approximately 30 minutes of flight time.

GAUI 500x ESC and brushless motors

The researchers used the stock GAUI 500x motors and ESC as they came with the drone kit and provide adequate power for flight of the drone system. That's the easy part, calibrating each ESC for the motors is a different story. For all rotors spins at different rate. It took us almost three weeks doing research and understanding the ESC and doing trial and error. Then the researchers found this forum where one person is having the same problem the researchers are facing and post several instruction as to how to calibrate ESC.

However as the researchers applied this, there were still errors in the motor responses. Further research was needed which lead to another solution found another post from the same person which was able to provide a more compatible set of instructions to calibrate the ESC components to our specific requirements.

CHAPTER IV
Android Application

The controller application is an android app that enable control of the quadcopter system by sending instructions or commands via the bluetooth feature of an android device. The lowest android version for this application is kitkat 4.4 since this version provides the necessary programming features for development of a stable and efficient bluetooth connection to the quadcopter system. There are two modes available for the user to choose, the first mode is manual flight control and the other mode is the INS or Automatic flight control. For the manual flight control the application uses images to display two joysticks for the throttle, rotation, roll and pitch of the quadcopter. There is also a display of the percentage of the current joystick value that is sent to the quadcopter system for the user to monitor. In manual control the user may also adjust the trim settings

of the quadcopter for fine tuning of the system. For calibration of the quadcopter using manual flight, the user would follow the normal calibration steps for the GAUI 500x quadcopter where for pre flight the throttle would be set to 100% the brought down to 0% after initial power up until the ESC modules are ready for main flight operation.

This is driven by a Bluetooth class where all bluetooth connection and functions can be found for the system. The bluetooth class handles the connectivity where initial set up occurs and in the event the connection would be interrupted an automatic reconnect would occur as to keep control of the quadcopter in flight. This class also handles receiving and writing of data to and from the android device to the bluetooth module on the quadcopter. Issues we encountered during the development of the manual control application where where the bluetooth could not detect the bluetooth device on the quadcopter, there would be difficulties paring to the device if the distance would increase to about 15m and above. Also, once the connection of the paired devices becomes interrupted the system would freeze for a few seconds as it tries to re pair to the disconnected devices. Usually the connection get interrupted when too many commands are sent or the Voltage of the power supply connected to the bluetooth module and the quadcopter goes below 8.0v. The values that are sent from the Android controller to the bluetooth module are string values with a tag and a value ex. Throttle: 100%. The range of values for the throttle, pitch, roll and rotation are 0-100 float values with 50% being the middle point. There are also trim values for the quad copter for the roll, pitch, and rotation of the quadcopter which range from 0-100 integer values with the middle and default value being 50.

The Autoflight option of the application would allow the user to upload a sketch of the area or building schematics where the device would operate and overlay instructions or waypoint for the quadcopter system to follow depending on the details set by the users. However the researchers will not apply this feature for this research due to time constraints.

One of the notable points to research in the area of the application is the reliability and effectiveness of using a bluetooth system to send instructions to a quadcopter in flight and receiving data from the quadcopter for whatever purpose. Initial testing of this is promising where the connection would be stable without disconnection when there is sufficient power supplied to the system however when there is insufficient power supplied to the bluetooth module the connection becomes very unstable and will be interrupted easily when required to process larger amounts of data. Sample screen shots of the application can be seen in the figures below.

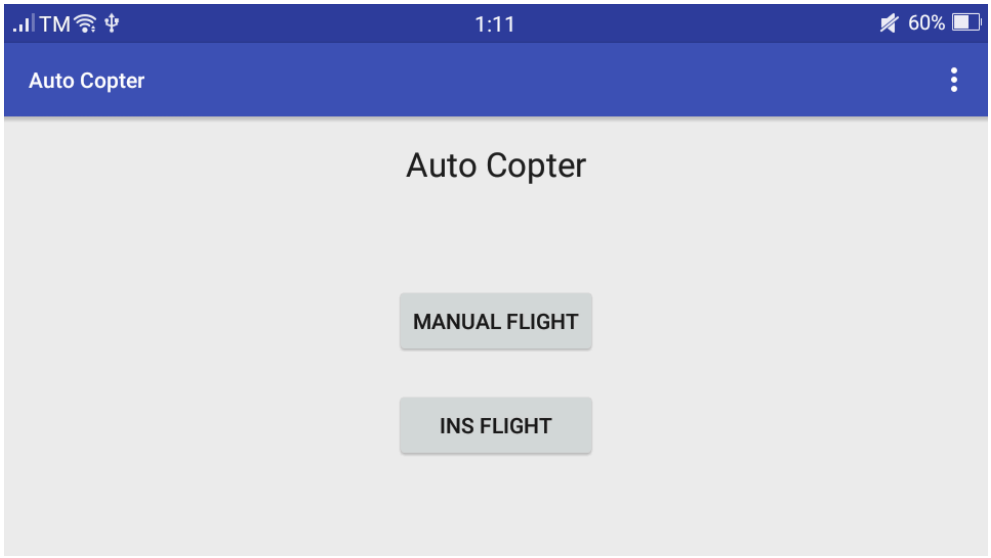


Figure 4.1 Application Initial Screen

Figure 4.1 is a screenshot that shows the initial screen of the android bluetooth controller application. It provides two options, one for manual flight and another for automated or Inertial Navigation System flight. However the INS feature of the application will not be explored during this research due to scope and time constraints. This research will focus on the manual control flight feature of the android controller application.

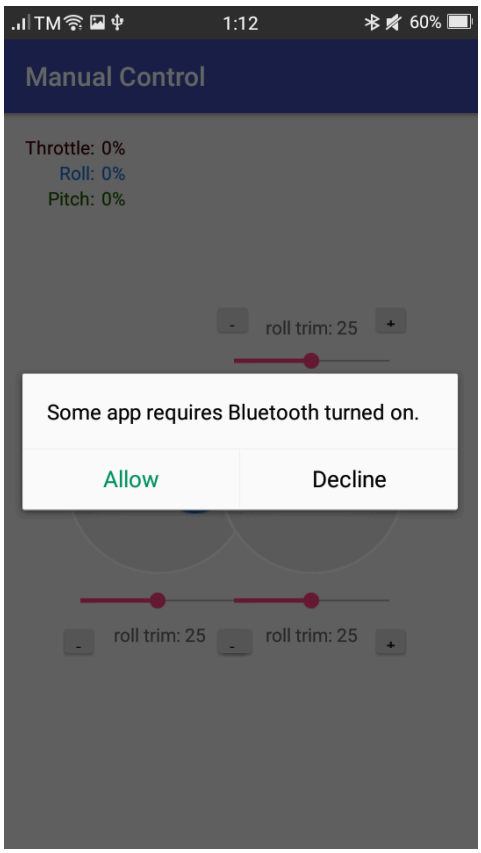


Figure 4.2 Bluetooth Required Screen

Figure 4.2 shows the screen that prompts the user to allow bluetooth to be turned on by the system, since android requires permission from the user to access certain features of the device. There are two options, allow and deny. Allowing the bluetooth

feature to be turned on will initiate bluetooth communication from the controller to the receiver on the copter. Declining will not turn on the bluetooth feature.

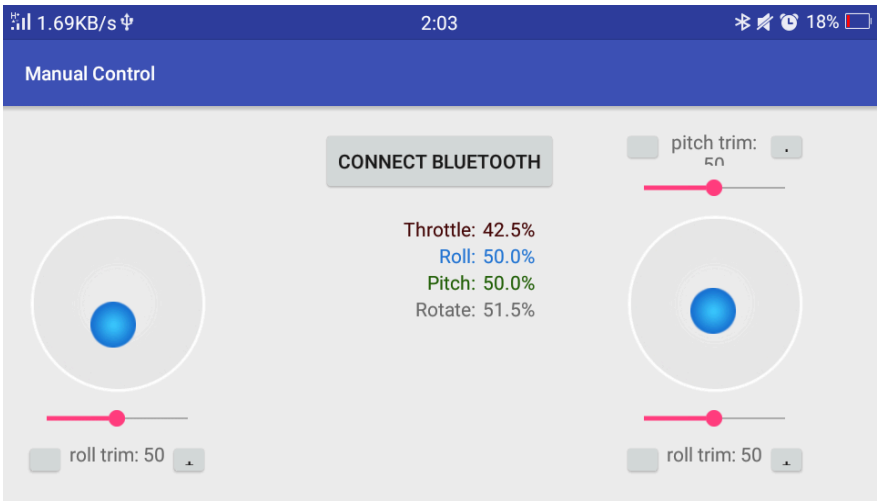


Figure 4.3 Main Controller Screen

Figure 4.3 is the main manual controller screen used to send commands to the copter via bluetooth. In this screen the user will find the two joysticks, trim sliders, value prints for debugging, and a connect to bluetooth button. The joysticks are setup and calibrated upon opening of the screen. The joysticks (blue circles) are then center in the center of the joystick areas (grey circles). The joysticks will stay within its respective joystick area even if it is dragged away. The left joystick handles the throttle and rudder (rotation) of the copter. The right joystick handles the pitch (forward and backward) and the yaw (strafe left and right) of the copter. The right joystick also sets itself to the center or neutral value when released by the user so the copter will return to a neutral state. The left joystick will stay where the user set it by dragging for the throttle and rudder. The debugging values allows us to track what command values are being sent to the copter. The connect to bluetooth button is used when the controller and the copter are ready to be connected to each other and begin communications.

Data Gathering

As seen in figure 4.1 the processes of the system start by gathering data for the system use. There are two sources of data, on being the commands from the bluetooth controller and the other from the Inertial Measurement Unit. The Inertial Measurement Unit Provides the gyroscope data which can be used to determine the orientation of the drone along with the accelerometer and the barometer data which provide any speed changes applied to the copter and the temperature of the copter at the moment. The command data from the bluetooth controller is set by the user from the android application where the user can set for the copter to perform a specific action.

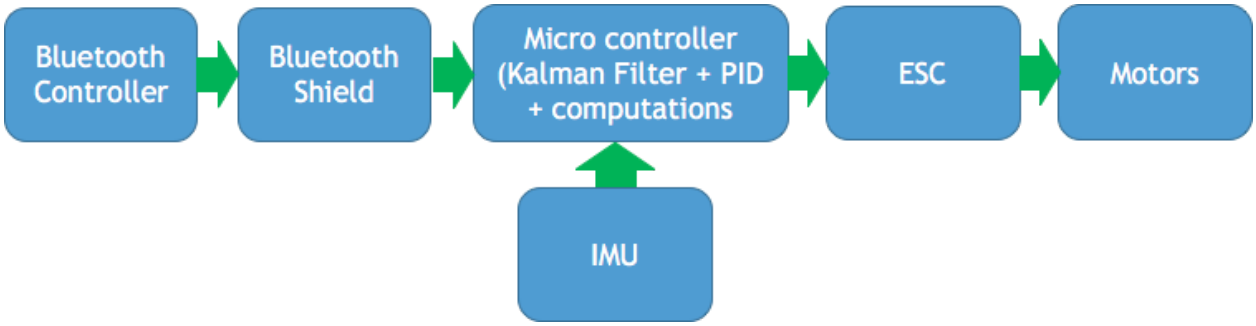


Figure 4.4 Process

To get the best results the system needs to calibrate the sensors before it can be used to collect data. The calibration involved finding the average “control” or base data from the sensors before system operation. This initial data is used as a baseline for further operations of the system and copter flight. This includes the gyroscope orientation. The sensors gathers data from the gyroscope over a few seconds and then finds the average value to be used as the basepoint of the system. When in flight the PID will use these values to come up with an appropriate response value to apply to the motors.

Kalman filter

The stabilization algorithm has many different steps and tasks to complete in order to be balanced and useful. Firstly the system must be able to receive consistent data from the IMU sensors and filter this data to ensure its reliability. The data from the IMU will be sent to the microcontroller and then Kalman filtering will be processed. The Kalman filter will be used to obtain a better data set from sensors that produce a lot of "noise". The filter will produce an educated guess or best estimate of the next data from the sensors. Then compares the best guess with the actual sensor data and returns the average result between to two data sets. This filter algorithm is best used when data is constantly changing from sensors and is most effective to rule out the "noise" from the sensors. This makes this algorithm best for a copter application since the data requires robust filtering from sensors to be able to come up with usable data. Some major equations that are involved are the following:

$$X_k = \mathbf{F_k} X_{k-1} + \mathbf{B_k} U_k$$

$$P_k = \mathbf{F_k} P_{k-1} \mathbf{F^T_k} + Q_k$$

X_k is the new best estimate state.

X_{k-1} is the current state.

P_k is the new best estimate position.

P_{x-1} is the current position.

F_k is the prediction.

B_k is the control matrix.

U_k is the unknown force vector

The new best estimate is a prediction made from previous best estimate, plus a correction for known external influences. And the new uncertainty is predicted from the old uncertainty, with some additional uncertainty from the environment.

$X^1_k = X_k + K^1 (Z_k - H_k X_k)$

$P^1_k = P_k - K^1 H_k P_k$

$K^1 = P_k H^T_k (H_k P_k H^T_k + R_k)^{-1}$

K^1 is the kalman gain matrix.

H_k is the sensor model matrix.

R_k is the covariance of sensor noise.

Z_k is the sensor data mean.

X^1_k is our new best estimate, and we can go on and feed it (along with P^1_k) back into another round of predict or update as many times as we like ((n.d.)). Since there are current arduino libraries available for application in sketches, the system will take advantage of the available libraries.

The data can then be used as crucial information about the current orientation and situation of the copter that can be applied to further calculations to produce the expected results. There is a Kalman filter library available for arduino programming which was used in this system developed by Kristian Lauszus, TKJ Electronics 2012 available at this link https://github.com/TKJElectronics/KalmanFilter. This library also support float math making it more precise for our application.

Proportional-Integral-Derivative processing

The Proportional-Integral-Derivative (PID) loop is used in our system to generate the optimal speed output for the motor compensation when the copter goes off balance. This process consists of three components, the proportional response, Integral response and the derivative response. The library Which is available at this website http://playground.arduino.cc/Code/PIDLibrary can be used by many micro controllers.

The proportional component depends only on the difference between the set point and the process variable. This difference is referred to as the Error term. The proportional gain (Kc) determines the ratio of output response to the error signal. For instance, if the error term has a magnitude of 10, a proportional gain of 5 would produce a proportional response of 50. In general, increasing the proportional gain will increase the speed of the control system response. However, if the proportional gain is too large, the process variable will begin to oscillate. If Kc is increased further, the oscillations will become larger and the system will become unstable and may even oscillate out of control.

The integral component sums the error term over time. The result is that even a small error term will cause the integral component to increase slowly. The integral response will continually increase over time unless the error is zero, so the effect is to drive the Steady-State error to zero. Steady-State error is the final difference between the process variable and setpoint. A phenomenon called integral windup results when integral action saturates a controller without the controller driving the error signal toward zero.

The derivative component causes the output to decrease if the process variable is increasing rapidly. The derivative response is proportional to the rate of change of the process variable. Increasing the derivative time (Td) parameter will cause the control system to react more strongly to changes in the error term and will increase the speed of the overall control system response. Most practical control systems use very small derivative time (Td), because the Derivative Response is highly sensitive to noise in the process variable signal. If the sensor feedback signal is noisy or if the control loop rate is too slow, the derivative response can make the control system unstable (PID Theory Explained. (n.d.)).

The PID processing provides a good output value for the motors to reduce over compensation and provide more stability for the copter in flight, however tuning and finding the right values for each component of the PID algorithm takes a lot of time and tedious experimentation to find the correct values for the best results.

Output Computation

The system we develop use a range of 0 to 180 for the speed controller values for our ESC components. The specific value that is sent to the motors are passed through multiple processes as mentioned. Firstly Kalman filter to provided better sensor data, then PID Processes the data to provide an optimal output and final adjustment computations for the speed values to avoid over compensation and unwanted movements. The tolerance level of the algorithm to 1 degree off center from the gyroscope angle that has been through the kalman filter. Once the copters orientation goes beyond the tolerance, the system

applies the algorithms output data to the ESCs which control the motor speed to compensate for the error in the copter orientation.

The motor speeds are initially set by the throttle input from the controller app. After the microcontroller receives the data from the controller it checks for the commands. The system adjust motor speeds depending on the desired command from the controller data and sends the speed values to each specified ESC respective to the command. When the command specifies that the drone should hover in place, the system checks the current orientation of the copter and calculates the necessary adjustments needed through the PID process algorithm. Then the system sends sends the PID return values to a mapping function. All values are mapped from their original values to a more appropriate value to provide better results. An example of the mapped values is the throttle. Where the values it receives from the command is 0 to 100 and is mapped to 70 to 130 to provide better control. Once the mapped value is ready it can then be sent to the ESC components which then set the speed of the motors.

Flowcharts

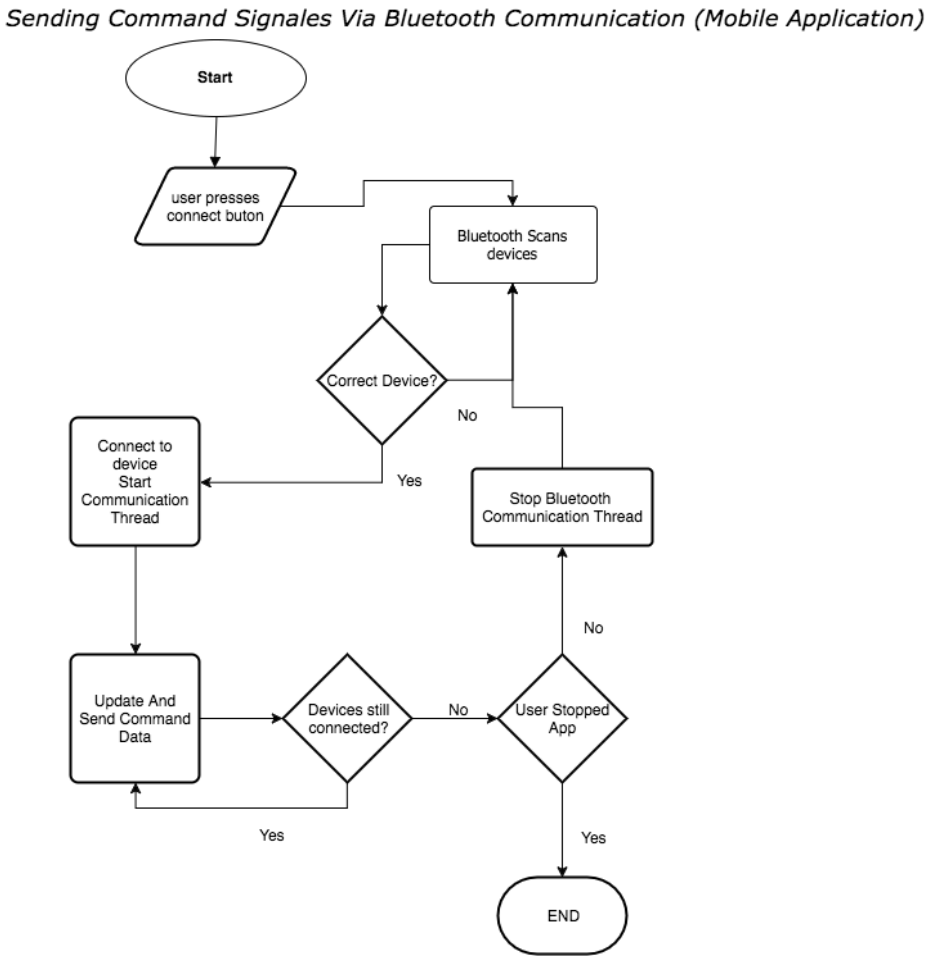


Figure 4.5 Bluetooth Connection (Mobile Application)

Figure 4.5 Illustrates the processes that the mobile application goes through while sending data to the bluetooth shield attached to the micro controller. This process begins when the user presses the connect button on the manual controller main screen of the application. It then begins testing all available devices in the area until the correct bluetooth device is found. Once the correct device has been found it begins the main bluetooth connection thread which constantly sends data to ensure a live connection between the devices at all time. If the connection has been interrupted the thread stops and

waits for the user to press the connect button to reconnect the two devices. If the user has closed the application the thread is cleared.

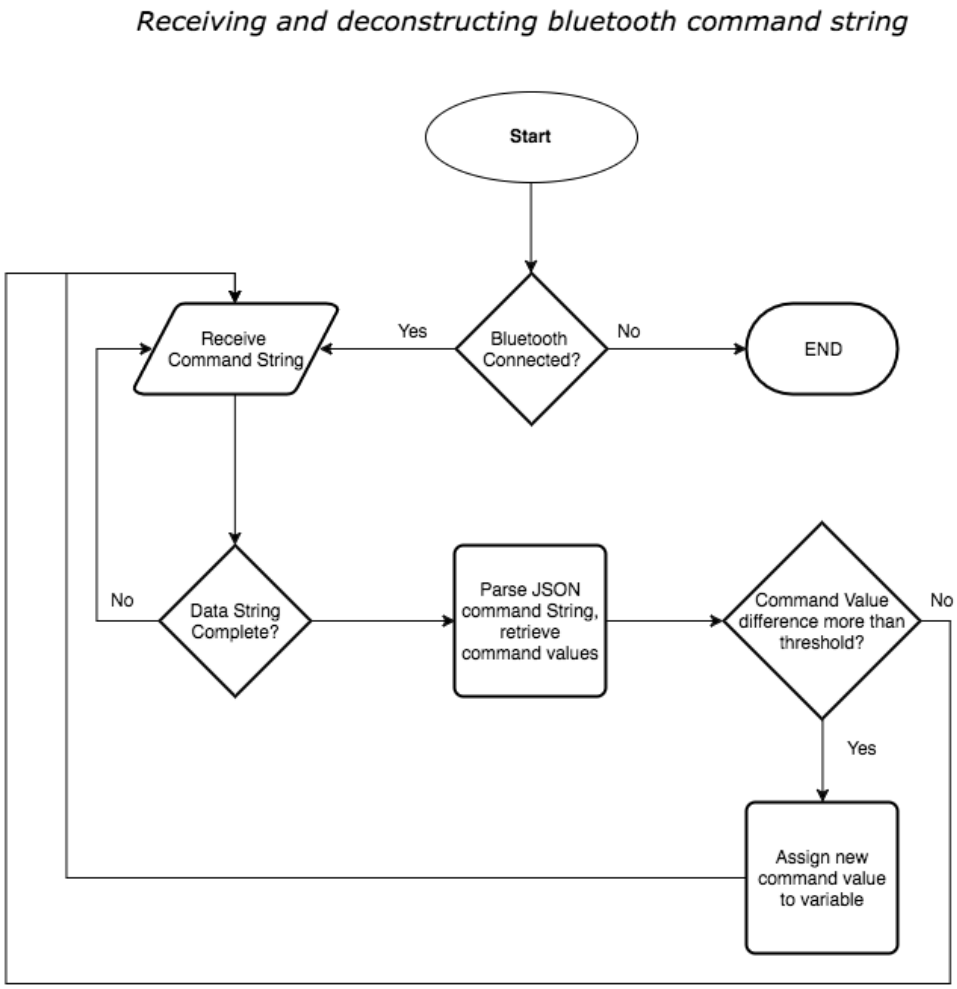


Figure 4.6 Receiving and Deconstructing Command String

Figure 4.6 illustrates the processes of receiving the command string from the bluetooth android application on the bluetooth shield. The bluetooth shield passes the data to the microcontroller which then processes the data. If the bluetooth module is not connected it will not receive any data. It also makes sure that the command data is also

complete and out of the threshold value before assigning the new command value to an array for further use in the system.

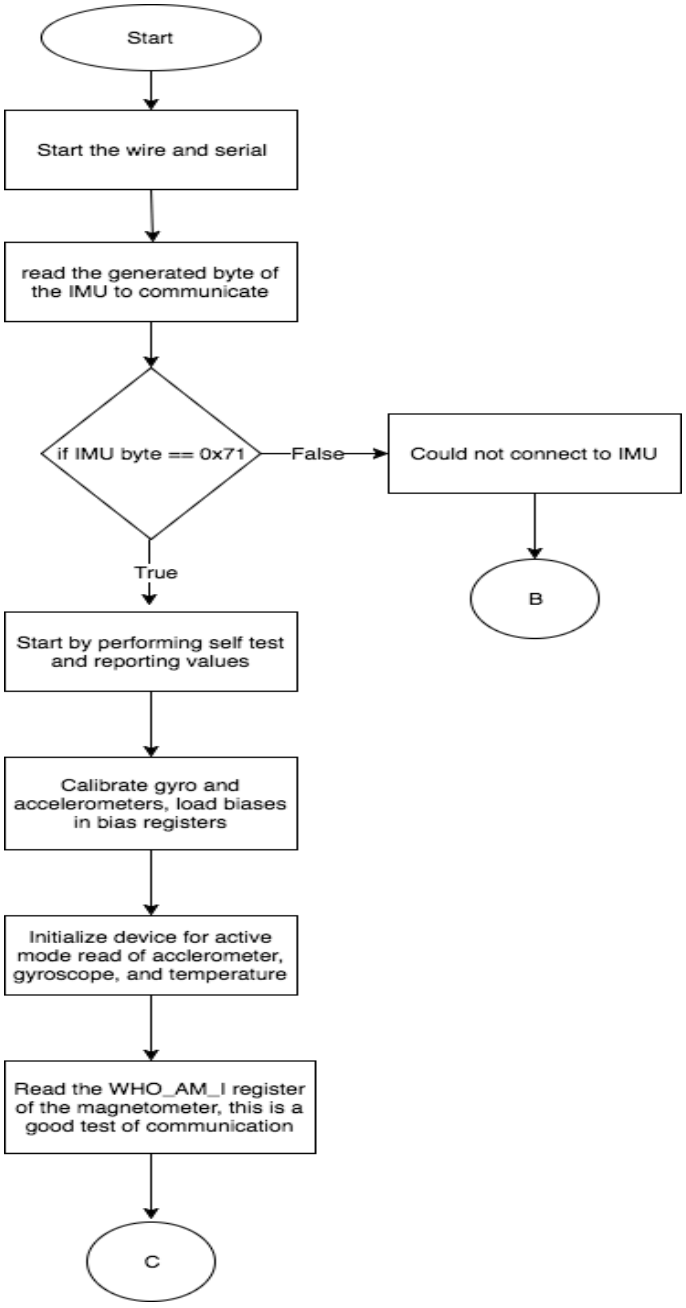


Figure 4.7 Setting up the IMU

Figure 4.7 illustrate the process of setting up the IMU. The microcontroller reads the IMU byte in determining if the IMU and microcontroller can communicate to each other. If the hexadecimal that passes the IMU is not equal to 0x71, it could not connect to the IMU else start the calibration and initialization of the gyroscope, accelerometer, temperature and magnetometer.

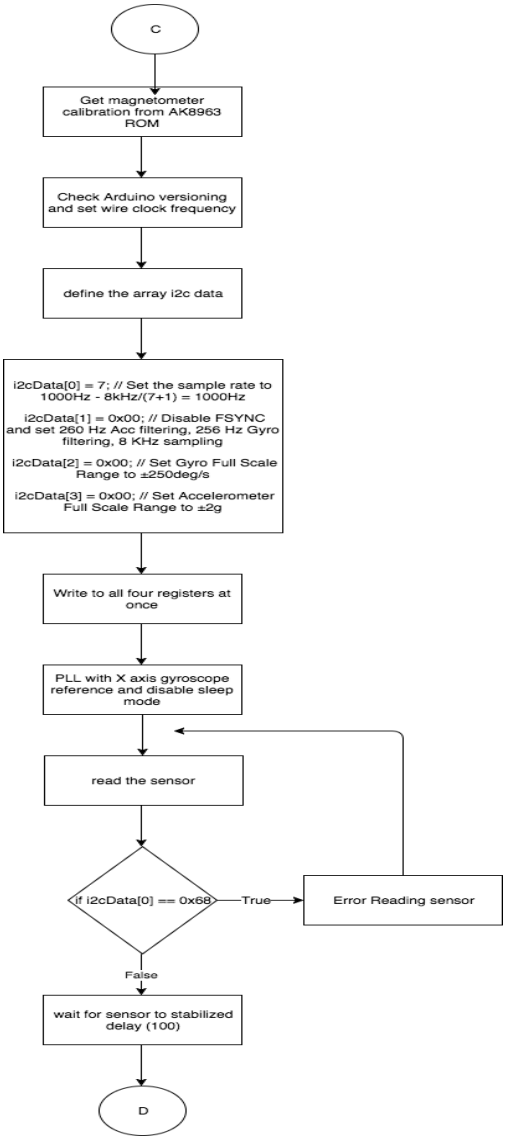


Figure 4.8 Reading the IMU data

Figure 4.8 illustrate the process in reading the IMU data from the microcontroller. The microcontroller set the wire clock frequency. The i2c array stored the full ranges of accelerometer and gyroscope,the filtering, and the sample rate. If the sample rate is equal to the hexadecimal of 0x68, it could the read the sensor else wait for 100 sec for the sensor to stabilized.

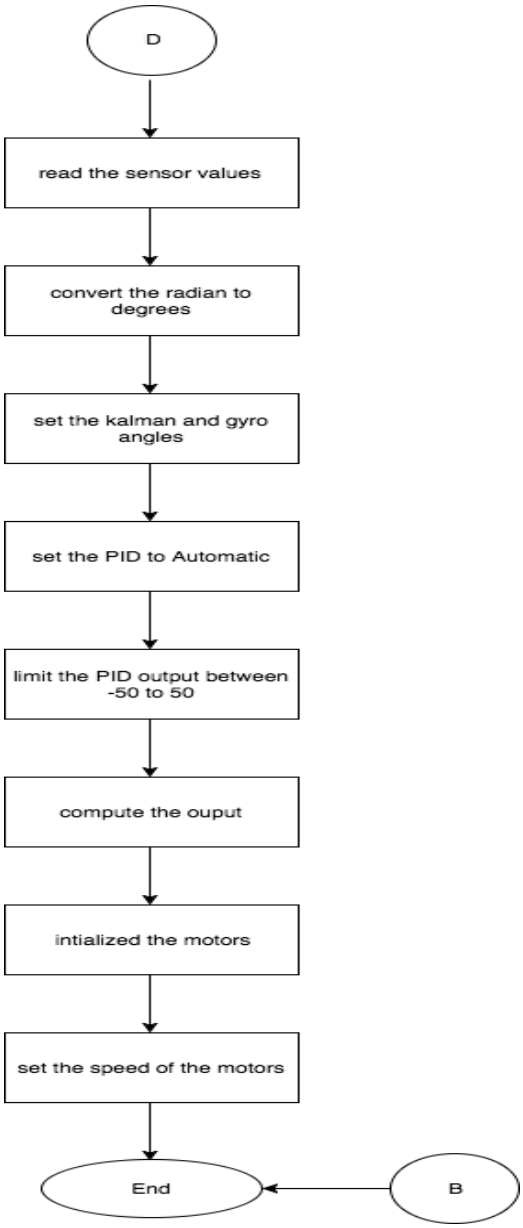


Figure 4.9 Setting up the motors

Figure 4.9 illustrate the process of setting the speed of the motors. After the microcontroller and the IMU is properly communicating, the microcontroller read the initial values of the sensors. The values is converted to degrees and set the converted values to the gyro and kalman variables. The PID mode is being set to AUTOMATIC mode and set the limit range between -50 to 50 the compute the initial values. The motors initialized and the speed is being set.

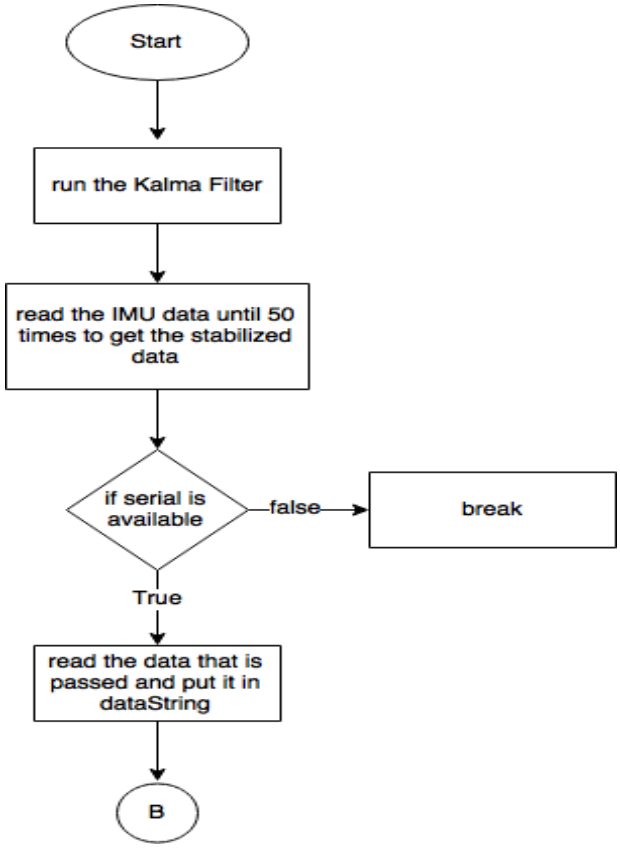


Figure 4.10 Serial Available

Figure 4.10 illustrate the process in getting the data that is pass from the bluetooth shield to the microcontroller. The data is being passed by character. When the serial is has a

data, the data is being added in dataString variable until serial is empty. The data is now a full text and can be used for the process.

CHAPTER V
Challenges

During development, the initial major obstacle the researchers encountered was developing a system that could maintain an efficient and stable bluetooth connection over the duration of the copter use. The researchers noticed that the main issue that caused connection interruption was when the controller was idle and not sending data to the copter. The researchers overcome this with a thread in the controller application the continuously checks for changes in the command data and sends the data to the copter each interval ensuring that the connection is never idle. However the connection was still being interrupted or disconnected even when sending data. The researchers found out that the bluetooth module wasn't receiving a sufficient power supply. This issue was quickly resolved when the power supply was upgraded from a lithium-ion to lithium polymer power supply. This new power supply not only solved the connection problem but also increased the flight time of the copter significantly.

The next obstacle we encountered was calibration of the ESCs. There are minimal documentation and data sheets that are available for the GAUI ESC, However the

researchers were able to come across a calibration manual which was used as a reference for the calibration. Calibration took some time and experimentation to find the optimal settings to conduct the research on the copter.

Sending a signal to the ESC components from a custom made software processed by the microcontroller. The researchers could not find any data sheets that provide information on what the required values are for proper use of the ESC components. The issue was solved through thorough experimentation and testing to find the optimal speed signal values that the ESC components and motors could respond to.

Using the GU-344 gyroscope flight controller was unsuccessful. The component cannot be programmed and the researchers were unable to receive feedback data. This became an unknown variable in the research that needed to be solved or removed. Research shows others have also failed in programming this component and had it replaced. Eventually this component was removed from the system since the IMU provided gyroscope information and testing would be more reliable with the feedback information available from the IMU.

Tuning and PID testing is one of the most tedious tasks during the research. The incorrect parameters for these algorithms will cause many problems for the copter. One of these is the copter over compensating to the changes in its orientation. Another issue is when the copter oscillates due to the over compensation. Solving this problem requires lots

of time and experimentation to find the optimal values to provide the best and most stable flight for the copter.

Testing and Results

Initial Test of the system showed that a more sufficient power supply was needed. The copter kept losing connectivity to the android controller and some motors would shut off during operation. Once we overcame the power supply issue the researchers then began the testing of the responsiveness to the controller commands. The copter would take moments to respond to changes in the command and was inefficient, The bluetooth thread solution proved to be a success in making a more efficient and stable connection. Once the thread was applied the responsiveness of the copter was enhanced significantly. Testing the main algorithm was a challenge. The researchers realized that the algorithm could not compensate for the physical weight alone and needed some assistance. We added a trim feature to the system as a solution to this issue. Then the algorithm is able to be more steady and compensate for angular changes.

The PID component of the algorithm needed rigorous testing to find the optimal values to prevent the drone from overcompensating and responding wildly to the changes. Testing each value is time consuming and tedious. Each test provided necessary data and adjustments were made making each test more positive and closer to the research objective. Below is a table with results of some of the test and changes we applied to the system during development.

Test 1 Roll trim 59 Negative Results Drone moves backward then goes right	Test 2 0 trims Negative Results Drone tries to balance but starts oscillating
Test 3 Changed pid D from .15 to .25 Negative Results Drone compensates wildly	Test 4 Changed pid P from .25 to .15 Negative Results drone compensates wildly
Test 5 Changed pid P from .25 to .35 Positive results Drone compensates less wildly	Test 6 Changed pid P from .35 to .45 Negative Results Drone compensates more wildly *Change pid P back to .35
Test 7 Added mapping pid values from -180 to 180 mapped -5 to 5 Changed pid I from .30 to .20 Negative Results mapping -180 is too high needs to be lowered	Test 8 Changed from map values from -180 & 180 to -50 & 50 *Trim fixed PID RESULTS Negative Results Drone overcompensates wildly
Test 9 Changes pid I from .20 to .40 Positive results Drone Is more stable but still oscillating	Test 10 To compensate about the twitch we changed the pid D from .15 to .25 Positive results Drone is more stable but still fluctuating
Test 11	Test 12

Changed pid I from .40 to .25 Positive results Drone is more stable less fluctuating	Changed pid P From .35 to .45 Positive results Drone is more stable but get off balance after a few moments
Test 13 Changed P from .45 to .35 Changed I from .25 to .15 Negative Results Drone has too much compensation	Test 14 Changed P from .35 to .45 Changed I from .15 to .25 Changed D from .25 to .65 Negative Results Drone has too much compensation
Test 15 Changed P from .45 to .05 Positive results Drone reacts quickly to changes but over compensates	Test 16 Changed I From .25 to .85 Negative Results Drone reacts at wrong times and overcompensates
Test 17 Changed I From .85 to .15 Reduced motor compensation speeds Positive results Drone reaction is more smooth	Test 18 Changed P from 1.0 to .80 Changed D from .65 to .35 Positive results Drone more stable and less compensation error

Table 5.1 Testing & Results

To calculate the accuracy percentage of the research we need to determine the positive and negative results. Then compute with the number of experiments to come up

with a percentage of the system research accuracy. The accuracy is tested in improvements in the stabilization, balance, reaction to angular changes and compensation during flight.

accuracy% = (#Tests - #Negative Results / #Tests) * 100

accuracy% = **38.89%**

Conclusion & Recommendation

This research found out that developing a flight controller stabilization system is no easy task. There are many things to consider in developing such system from all aspects, including electronics engineering, physics, aeronautics, and programming theories and applications. The results initially were very negative where the copter had almost no balance at all and gradually improved towards the end of the research time period, but it is still far from perfect. The research ended up with negative results due to the lack of time to test and to correctly tune the PID component of the algorithm. However with further testing and experimentation the algorithm may produce positive results once the optimal PID parameters can be obtained and minor tweaking to the algorithm is applied.

APPENDICES

APPENDIX A

ESC Calibration

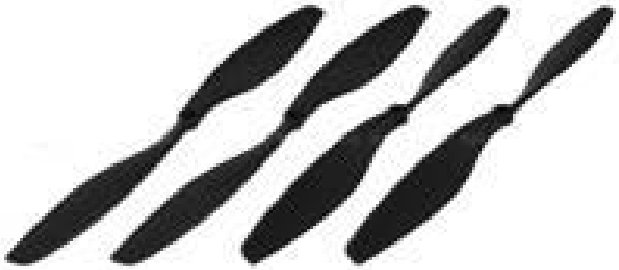
0. Switch off the Tx and GAUI
1. remove all ESC leads from the GU-344
2. remove the GU-344 gyro green wire from rx
3. set the gain on the GU-344 to minimal (ccw)
4. plug in the ESC lead of motor #1 in port #1 of the GU-344
5. set the throttle at 100%, switch on the tx and then the Gaui
6. wait for the beeps
7. move throttle to 0%
8. wait for the acknowledge beeps
9. switch off the GAUI
10. remove the ESC lead from the GU-344
11. repeat steps 4-9, replace the red numbers for the appropriate motor.
12. when all four have been calibrated, plug in all the ESC leads into the GU-344 and ditto the gyro lead into rx
13. adjust the gain of the GU-344 to nine/ten o'clock for starters.

ESC Calibration Gaui 330xs (using a Futaba 8FG)

1. While powered off, unplug the green wire going from the GU-344 to channel 5 of the Receiver ('Rx') on the 330xs.
2. Set the Gain dial on the GU-344 to Zero.

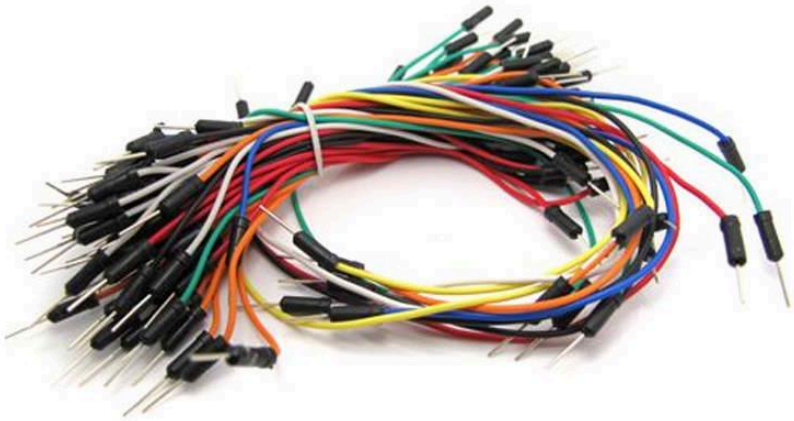
3. Turn on transmitter. Set the Throttle end points to 100 for both sides (Go into Linkage Menu, end point, THR, set the two middle numbers to 100)
4. Set the Gear end points to 50 for both sides (Go into Linkage Menu, End Point, GEAR, set the two middle numbers to 50)
5. From here, go to page 17 of the Gaui 500x manual found here: http://helitech-jp.com/download/500X_CH_EN110624.pdf
6. Follow steps 1-4 written in the the boxes shown on page 17.
7. When finished, unplug the 330xs.
8. Plug the green wire back in.
9. Set the Gain dial on the GU-344 to half way (50%).
10. Set the Throttle end points to 80 for both sides (Go into Linkage Menu, end point, THR, set the two middle numbers to 80).
11. THIS WILL VARY DEPENDING ON HOW THE GAUI REACTS. I set the Gear end points to 45 for both sides (Go into Linkage Menu, Endpoint, GEAR, set the two middle numbers to 45).

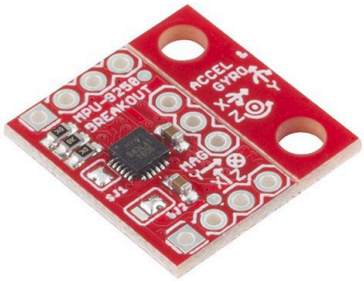
APPENDIX B
SUMMARY OF MATERIALS

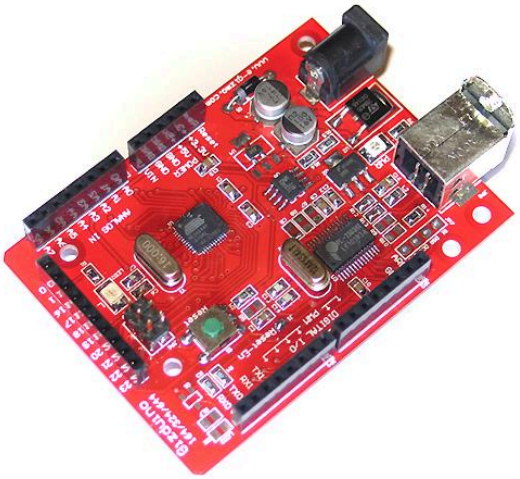



sportsmoto


GAUI




ACCEL
GYRO
MPU-9250
BREAKOUT
MAG



References

[1] Inertial Navigation System (INS). (n.d.). Retrieved July 11, 2016, from http://www.skybrary.aero/index.php/Inertial_Navigation_System_(INS)

Drones in logistics – internet of things, unmanned cargo, uav delivery. (n.d.). Retrieved July 20, 2016, from https://www.microdrones.com/en/applications/growth-markets/quadcopter-for-logistics/

News, A. (2016). Pinoy-invented drone seeks to improve search and rescue in PH. Retrieved July 20, 2016, from http://news.abs-cbn.com/global-filipino/02/02/16/pinoy-invented-drone-seeks-to-improve-search-and-rescue-in-ph

Beginner's guide to drone autopilots and how they work. (n.d.). Retrieved July 20, 2016, from http://www.dronetrest.com/t/beginners-guide-to-drone-autopilots-and-how-they-work/1380

Welch, G., & Bishop, G. (n.d.). An Introduction to the Kalman Filter. Retrieved August 5, 2016, from http://www.cs.unc.edu/~tracker/media/pdf/SIGGRAPH2001_CoursePack_08.pdf

PID controller. (2017, February 24). Retrieved March 02, 2017, from https://en.wikipedia.org/wiki/PID_controller

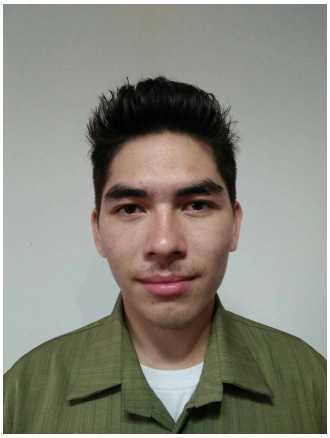
(n.d.). Retrieved March 02, 2017, from http://robotsforroboticists.com/kalman-filtering/%E2%80%A6...http%3A//home.wlu.edu/~levys/kalman_tutorial/

PID Theory Explained. (n.d.). Retrieved March 02, 2017, from http://www.ni.com/white-paper/3782/en/#toc2

Amato, A., Says, A., Says, C., Says, M., Says, K., & Says, R. (2015, April 16). Drone Sales Numbers: Nobody Knows, So We Venture A Guess. Retrieved March 02, 2017, from http://dronelife.com/2015/04/16/drone-sales-numbers-nobody-knows-so-we-venture-a-guess/

UNMANNED AERIAL VEHICLE IN LOGISTICS. (2014). Retrieved March 2, 2017, from http://www.dhl.com/content/dam/downloads/g0/about_us/logistics_insights/DHL_TrendReport_UAV.pdf
PUBLISHER DHL Customer Solutions & Innovation Represented by Matthias Heutger, Senior Vice President Strategy, Marketing & Development, DHL CSI 53844 Troisdorf, Germany

Curriculum Vitae



MEMBER'S PROFILE

Name: Jamie Joe M. Van Stone

Gender: Male

Age: 22

Date of Birth: October 14, 1994

Place of Birth: Red Deer, Alberta, Canada

Civil Status: Single **Citizenship:** Filipino - Canadian

Religion: Christian **Cellphone:** +639361041760

Address: N Junob, Dumaguete, Negros Island Region, Philippines

Email Address: jamiejoevanstone@gmail.com

Father: Mel Van Stone

Mother: Ana Maria Van Stone

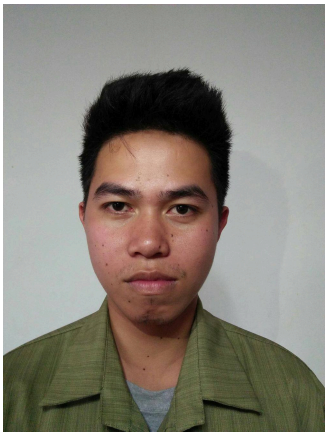
Education

Elementary: St. Patrick's Community School, Foundation University

High School: Foundation University

College: Foundation University

Course: Bachelor of Science in Computer Science



MEMBER'S PROFILE

Name: Jassam Alinsub

Gender: Male

Age: 23

Date of Birth: October 10, 1993

Place of Birth: Balayagmanok, Valencia Negros Oriental

Civil Status: Single **Citizenship:** Filipino

Religion: Roman Catholic **Cellphone:** 09754003811

Address: Balayagmanok, Valencia Negros Oriental

Email Address: jassam.slayer@gmail.com

Father: Buenaventura Alinsub

Mother: Cipriana Alinsub

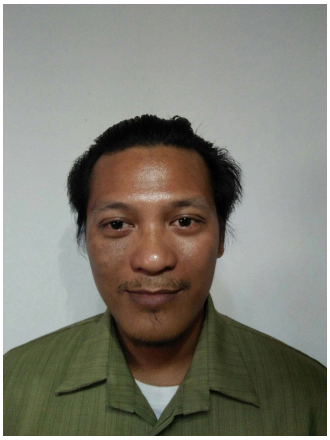
Education

Elementary: Bong-ao Elementary School

High School: Valencia National High School

College: Foundation University

Course: Bachelor of Science in Computer Science



MEMBER'S PROFILE

Name: Jeffrey Tubog

Gender: Male

Age: 34

Date of Birth: August 4, 1982

Place of Birth: Jolo, Sulu

Civil Status: Single **Citizenship:** Filipino

Religion: Roman Catholic **Cellphone:** +63 917 840 5333

Address: Bulak, Dauin

Email Address: jeffreytubog30@gmail.com

Father: Rosario Tubog

Mother: Mancia Fortuito

Education

Elementary: Moh. Tulawe Central School

High School: Notre Dame of Jolo For Boys

College: Foundation University

Course: Bachelor of Science in Computer Science