Autonomous Quadcopter Based Solution for Spraying of Fertilizers and Pesticides using Neural Networks for Flight Stabilization

Submitted in partial fulfillment of the requirements
of the degree of

Bachelor of Engineering

by

Siddhant Gangapurwala (60001120012)

Project Guide:
Prof. Pavankumar Borra

Electronics Engineering
Dwarkadas J. Sanghvi College of Engineering
Mumbai University
2015-2016

Certificate

This is to certify that the project entitled **"Autonomous Quadcopter Based Solution for Spraying of Fertilizers and Pesticides using Neural Networks for Flight Stabilization"** is a bonafide work of "**Siddhant Gangapurwala" (60001120012)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **"Bachelor Of Engineering"** in Electronics Engineering.

Internal Guide
Prof. Pavankumar Borra

Internal Examiner

Head of Department
Prof. Prasad Joshi

External Examiner

Principal
Dr. Hari Vasudevan

Project Report Approval for B. E.

This project report entitled ***Autonomous Quadcopter Based Solution for Spraying of Fertilizers and Pesticides using Neural Networks for Flight Stabilization*** by ***Siddhant Gangapurwala*** is approved for the degree of **Electronics Engineering**.

Examiners

1.--

2.--

Date:

Place:

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Siddhant Gangapurwala (60001120012)

Date:

Abstract

The past few years have been an evidence to the growing demand of Quadcopters and their applications in varied fields ranging from Military Operations to Geo Imaging Solutions (GIS). Several of these applications require an autonomous approach for its implementation inclusive of modules such as Navigation, Collision Avoidance, Flight Stabilization, Localization and Fault Detection.

This project aims at developing a solution for spraying of fertilizers and pesticides in agricultural colonies. A quadcopter, due to its flexibility of flight control, has been chosen as the Master Device. A GUI has been developed for PC systems as a control station to authorize the flight instructions to the Quadcopter. The control station is used to send instructions to the Quadcopter which determines the functional behavior of the device. The STM32F407 microcontroller is used as a flight controller for the Quadcopter. The device transmits status parameters to the Control Station, wirelessly using a ZigBee module, at a configurable interval. The use of a 3-Axis Accelerometer, 3-Axis Gyroscope, 3-Axis Magnetometer, Barometer and a Global Positioning System (GPS) module has been concluded to achieve flight control. The Kalman Filter for MARG Sensor Fusion shall be used for State Estimation. The Duty Cycle of the PWM wave, used as a DC Motor Speed Control Method, shall be a function of the State Parameters. Neural Networks shall be implemented to achieve stable flight characteristics. The end goal of the project is to design and develop an Autonomous, Self-Navigating Quadcopter to dump the Fertilizers and/or Pesticides at user defined Nodes.

Contents

CHAPTER NO.	SECTION NO.	TOPIC	PAGE NO.
		Certificate	**ii**
		Project Report Approval for B. E.	**iii**
		Declaration	**iv**
		Abstract	**v**
1.		**Introduction**	1
	1.1	Autonomy	2
	1.2	Function	3
	1.3	Design Approach	4
2.		**Review of Literature**	**5**
	2.1	Neural Network Control of a Quadcopter System	6
	2.2	Kalman Filter Design for State Estimation	7
3.		**Design and Control Components**	**9**
	3.1	Build Components	10
	3.2	Flight Control Components	12
	3.3	Communication	15
4.		**System Implementation**	**18**
	4.1	System Design	19
	4.2	Autonomous Behavior of Quadcopter	22
	4.3	Modular Assemblies as Containers for Fertilizers and Pesticides	26
5.		**Control System Design for Quadcopter**	**27**
	5.1	PID Control of a System	28
	5.2	Modelling Quadcopter System	30
6.		**Design of the Flight Control Board**	**35**
	6.1	Flight Control Board	36
7	7.1	**Conclusions & Scope**	**38**
8		**REFERENCES & BIBLIOGRAPGHY**	**40**
9		**Acknowledgments**	**42**

Chapter 1

Introduction

The success of Quadcopters lies in the huge Commercial Potential these devices possess. Currently, Quadcopters are being used for applications such as Mapping of Buildings, Bridge Inspection, Precision Farming, Factory Planning and even Transportation. However, despite the increase in performance of these devices, much of the solutions highly rely on Skilled Human Pilots. To further increase the adaptability of Quadcopters, a certain degree of Autonomy is necessary. One such application of an Autonomous Quadcopter is in the Agricultural domain.

Fertilizers and Pesticides have always been suspected of affecting the Human Health significantly upon continuous direct exposure. Currently, the agricultural sector relies heavily on the use of Synthetic Fertilizers and Pesticides. Children living near an Agricultural area have been observed to have a higher risk of Leukaemia. In Adults, this has resulted in a greater risk of Brain Cancer. Astrocytoma, a type of Brain Cancer, rose 6.7 times from the year 2003 following the increased use of Synthetic Fertilizers and Pesticides. The Toxicity of such chemicals can lead to permanent damages within people closely related to Agriculture.

An Autonomous Quadcopter for the purpose of Spraying Fertilizers and Pesticides could reduce the health effects on individuals and could further be used for Precision farming maximizing the yield of the Agricultural Land.

1.1 Autonomy

It is required of the Quadcopter to be Autonomous up to a certain extent so as to increase its usability and adaptability. The levels of Autonomy could be categorized as:

- Manual Flight – Automatic Attitude
- Semi-Autonomous – Height Control, Position Control, Obstacle Avoidance
- Fully-Autonomous – Waypoint Navigation, Path Planning, Exploration

The Quadcopter, for its implementation in the discussed field, is expected to be Fully-Autonomous in order to maintain as little human intervention as possible. The Quadcopter shall include features like Navigation, Collision Avoidance, Localization, State Estimation and Stabilization. It shall also Redirect its path to Home (Starting Location) in case there is no communication with the Control Station. Such a control requires a precise State Estimation Model in order to detect its Current location in a 3D vector Space. To achieve a high degree of Autonomy, the Quadcopter possesses authority over the control station. The Quadcopter has the ability to override the Control Station Instructions in order to service exceptions that have a higher priority than the Control Instructions.

1.1.1 Navigation

One of the main components of Autonomy include Waypoint Navigation. This enables the user to set a pre-defined location such that the device navigates to the destination without any user control. In Quadcopters, this is often done using 4D Matrix Translation methods in which the Motor speeds are a function of Yaw, Pitch and Roll. Navigation shall be further discussed in detail in the later chapters.

1.1.2 Exteroception

The device also needs to sense its environment in order to achieve Autonomy. This enables the device to function in Environments that may require calibrated control. For example, sensing the temperature is necessary so as to know the drift in the Barometer value that may have occurred. Another example may include change in Motor Speed in order to compensate of the Air Resistance.

1.1.3 Proprioception

This involves self-maintenance of the Device. For example, in case of a low battery, the Quadcopter to return back to the Starting Location and notify the Control Station about its low battery.

1.2 Function

The device shall dump the Fertilizers and Pesticides at User-Defined nodes as instructed to the device through the control station. An alternative to the approach could be to hover the Quadcopter over the periphery of the field and let the Quadcopter estimate the Nodes. This could be achieved by implementing equally spaced Nodes over the area of the field.

Furthermore, the Control Station shall log the Status Parameters transmitted by the Quadcopter to enable Machine Learning algorithms in order to better estimate the State Vector Position of the device.

1.3 Design Approach

The Device consists of a frame on which the microcontroller is mounted along with the necessary modules required to Navigate. The Motors are attached to the frame and are connected to the Microcontroller using Electronic Speed Controllers (ESC) that are used to drive the Motors. Module assemblies shall be used to connect the Fertilizers and Pesticides Containers to the Quadcopter. Actuators shall be used to dump the fertilizers and pesticides at a certain user-defined Node. For test purposes, small module assemblies shall be used such that the motors are capable of lifting the device off the ground.

For Portable Remote Control of the Quadcopter an Android based Control Station could be developed to deliver on-site control instructions to the Quadcopter.

The above mentioned approach shall enable an unskilled user to control the Quadcopter remotely with a portable device.

The main components used to determine the State Position of the device are:

- 3-Axis Accelerometer
- 3-Axis Gyroscope
- 3-Axis Magnetometer
- Barometer
- GPS

The various modules used for the development of the solution shall be discussed in the following chapters.

Chapter 2

Review of Literature

The project proposes an alternative to achieve Device Stabilization. Conventionally, the Quadcopter stabilization technique that has been widely used is based on the principles of PID Control. However, research in the field of Neural Networks has led to its implementation for Quadcopter Stabilization.

Several, Research Papers have been presented to include Neural Network Algorithms for device control. The chapter discusses and reviews these approaches for the control of Quadcopter and its application in the project.

2.1 Neural Network Control of a Quadcopter System

In the Research Paper presented by Authors 'C. Nicol', 'C.J.B. Macnab1', 'A. Ramirez-Serrano', titled 'Robust Neural Network Control of a Quadcopter Helicopter', it has been discussed in depth the need for the use of Neural Networks. Due to uncertainties in the Environment a Quadcopter could lose its Stability. Parameters such as Wind Speed, Temperature and Modelling Error could cause a significant drift in the State Vectors.

In order to address the issues that have been experienced in Quadcopter control, the Authors propose the use of a Cerebellar Model Articulation Controller (CMAC). The CMAC consists of an offset of Look-Up table divided into Hyper-Cube cells. Each input is a dimension in the CMAC structure. Each cell then has a Basis Function and a Weight associated with it. Much like the Radial Basis Function Network, the output can be expressed in terms of the input vector 'q' as the weighted sum of the basis function:

$$f'(q) = \Gamma(q).w$$

Here, 'Γ' represents the row vector of the values of the basis function associated with the activated cells and column vector 'w' corresponds to its associated weight. The control strategy for the flying/hovering modes involved the development of two loops. The outer loop is responsible for providing a desired roll and pitch (and the velocities) to the inner control loop in an effort to achieve desired X and Y coordinates (or velocities). The (faster) inner loop is responsible for achieving the desired roll and pitch.

2.1.1 Design of the System Model

The control strategy for the flying/hovering modes involved the development of two loops. The outer loop was responsible for providing a desired roll and pitch (and the velocities) to the inner control to achieve desired X and Y coordinates (or velocities). The (faster) inner loop was responsible for achieving the desired roll and pitch. The Research paper portrays, in great detail, the practical implementation of such a model. It proposes a Control Strategy using the discussed Neural Network for a given State Space Representation. The given description in the paper shall be used with a certain modification to apply its principles onto a 3-Axis control for Yaw, Pitch and Roll in the ongoing project.

2.1.2 Results and Conclusion

The method proposed involves complex mathematical solutions. Implementing such a model requires a high amount of processing. Hence, a suitable Microcontroller is required in order to compute at a higher rate. Also, the Neural Network architecture promises hopeful results in real prototype tests upon its comparison with alternate control procedures. There is a considerable reduction in the Weight drift and a far less error is achieved for a desired Attitude upon the use of CMAC. Therefore, use of such a mechanism for control of Quadcopter could provide a more stable response.

2.2 Kalman Filter Design for State Estimation

The Kalman filter has been widely used as a Sensor Fusion Technique for State Estimation. It has been incorporated in almost every device that uses an Accelerometer and a Gyroscope. In the paper presented by 'Péter Bauer', 'György Ritzinger', 'Alexandros Soumelidis' and 'József Bokor', the use of Kalman Filter for Servo Control has been discussed. Titled 'LQ Servo control design with Kalman filter for a quadrotor UAV', the paper presents the State Equations for direct control of Motors with Kalman Filter implied for State Estimation.

The paper presented a nonlinear model of a quadrotor UAV, and introduced a possible spatial trajectory tracking controller design technique. The nonlinear simulation model was implemented in a Simulation Program and Controller was designed for a Linear Model obtained from it in hovering mode. The controller consisted of an LQ Servo part with double integrator and a Kalman filter state observer. Weighting matrices were selected considering indoor constraints, data sheets and physical aspects. The resulted controller could track well a spatial position reference signal, however, it failed to control the Orientation Model. Thus, such a technique could be well implemented to determine the State Vectors of the Quadcopter. Simulations were done with the linear model and also with the whole nonlinear system using both the Servo controller and the Kalman filter. The designed Kalman filter was stable and gave satisfactory results during simulations.

2.2.1 Application of the Analysed Model

The Kalman filter provides a good estimate of the state position. Also, the paper has been successful in putting forward an acceptable Control Method for Servo Motors. Since it has been specifically designed for Quadcopters, it can be well used to achieve Autonomy in Quadcopters. The given model proposes a Linear Control mechanism. An Extended Kalman Filter uses a Non-Linear approach. The higher degree of Estimation is thus achieved using an Extended Kalman Filter. However, the Linear Model, for its simplicity as compared with the Extended Kalman Filter Model, could be well used to avoid any Delay in response from the controller.

2.2.2 Results and Conclusion

The linear controller provided good reference signal tracking even on the nonlinear system model. This is, because with slowly varying reference signal, the system could stay inside the linear region around hovering position. Thus the control of a Highly Non-Linear Quadrotor Model was proposed using a Linear Solution. Thus, the control of the Autonomous Quadcopter could be achieved using a Linear Model, reducing the complexity involved to compute its State Models.

Hence, in order to compensate for the Processing required for Neural Networks, the Project shall implement a Linear Model to control the Quadcopter, to reduce any delay so as to service the Motors. In turn, this shall reduce the amount of Oscillations observed at the state of hovering.

Chapter 3

Design and Control Components

Quadcopters generally use two pairs of identical fixed pitched propellers; two clockwise (CW) and two counter-clockwise (CCW). These use independent variation of the speed of each rotor to achieve control. By changing the speed of each rotor it is possible to specifically generate a desired total thrust; to locate for the centre of thrust both laterally and longitudinally; and to create a desired total torque, or turning force.

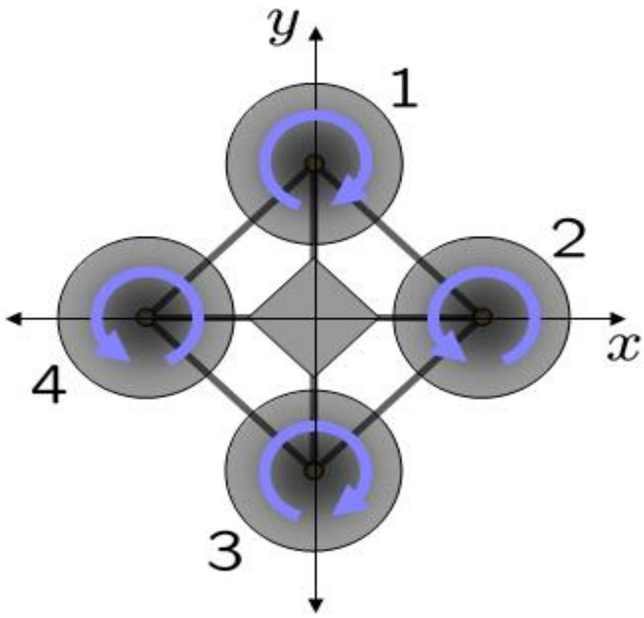

Figure 3.1: Motor Rotation Directions

Each rotor produces both a thrust and torque about its centre of rotation, as well as a drag force opposite to the vehicle's direction of flight. If all rotors are spinning at the same angular velocity, with rotors one and three rotating clockwise and rotors two and four, counter clockwise, the net aerodynamic torque, and hence the angular acceleration about the yaw axis, is exactly zero, which implies that the yaw stabilizing rotor of conventional helicopters is not needed. Yaw is induced by offsetting the cumulative thrust commands between the counter-rotating blade pairs. The design considerations while developing a Quadcopter are of utmost importance. To avoid Yaw imbalances, the Quadcopter is required to have a Symmetric Structure.

Quadcopters often can fly autonomously. Many modern flight controllers use software that allows the user to mark "way-points" on a map, to which the quadcopter will fly and perform tasks, such as landing or gaining altitude. This requires complex State Estimation Algorithms to provide a Stable Response to Navigate towards the Way-Point. This chapter discusses the design of the Quadcopter along with the Control Elements to direct the device.

3.1 Build Components

The Quadcopter requires certain basic components to function. Depending upon its application, these components can be substituted by a set of other components. The following subsections shall give a brief on the used Quadcopter components.

3.1.1 Frame

Figure 3.2 Carbon Fibre Frame

Every quadcopter or any other multirotor aircraft needs a frame to house all the other components. Things to consider while choosing a frame are the weight, size, and build material. While working on the project, Carbon Fibre Frame was used. It is strong, light, and has a built-in power distribution board (PDB) that allows for a clean and easy installation of ESCs.

3.1.2 Motors

The motors are used for spinning the propellers. Motors are rated by kilovolts, and the higher the kV rating, the faster the motor spins at a constant voltage. Also, the designer needs to consider the Power Consumed by motors during its operation. This shall in turn decide the Capacity of the Battery used for the Quadcopter.

3.1.3 Electronic Speed Controllers (ESC)

Figure 3.3 Electronic Speed Controller Running SimonK Firmware

The electronic speed controller, or ESC, is used to control the Motor Speed. The Quadcopter requires four ESCs, each connected to the one Motor. The ESCs are then connected directly to the battery through either a wiring harness or power distribution board. Many ESCs come with a built in battery eliminator circuit (BEC), which allows to power external components such as the flight control board and radio receiver without connecting them directly to the battery. Because the motors on a quadcopter must all spin at precise speeds to achieve accurate flight, the ESC is a very important component of the Build. Most of the ESCs have the SimonK firmware flashed into them. This allows for variable refresh rates and thus precise control of the Motors.

3.1.4 Propellers

A quadcopter has four propellers, two “normal” propellers that spin counter-clockwise, and two “pusher” propellers that spin clockwise. The pusher propellers are usually labeled with an ‘R’ after the size. The propellers are the most important components that determine the Thrust exerted by the Quadcopter. The Thrust is proportional to the square of the Diameter of the Propellers.

3.1.4 Battery and Battery Charger

Quadcopters typically use Lithium Polymer (LiPo) batteries which come in a variety of sizes and configurations. LiPo batteries have a C rating and a power rating in mAh (milliamps per hour). The C rating describes the rate at which power can be drawn from the battery, and the power rating describes how much power the battery can supply. Larger batteries weigh more so there is always a trade-off between flight duration and total weight. Charging LiPos is a complex process, because there are usually multiple cells within the battery that must be charged and discharged at the same rate. Therefore, a balance charger is often used. It is necessary that the balance charger be of a good quality since a low quality charger could even cause explosions.

3.1 Flight Control Components

It is necessary to have a set of components that govern the response of the Quadcopter to the instructions it receives from the Control Station. Most importantly, the level of Autonomy of the Quadcopter results from the available degrees of control. Considering the implementation of an Autonomous system, it is necessary to have a processor that can handle the complex Modelling Equations and Simultaneous have a control over the motors.

3.2.1 Flight Controller (STM32F407)

The flight controller is the brain of a Quadcopter. It is used to Control the motor speeds so as to maintain stability and in case of Autonomous behaviour, to Navigate and even avoid Collision. There are several flight control boards readily available in the market. However, most of these flight controllers are designed for Manual Flight control. For the project, the STM32F407 controller was chosen. It is based on the Cortex M4 architecture and has a Floating Point Unit (FPU) that allows for faster Mathematical Computations. It also offers a high degree of Interrupt Servicing so as to maintain the Autonomy of the Quadcopter and implement a basic Task Scheduling structure. Some of its features include:

- 2x USB OTG (one with HS support)
- Audio: dedicated audio PLL and 2 full duplex I²S
- Up to 15 communication interfaces (including 6x USARTs running at up to 11.25 Mbit/s, 3x SPI running at up to 45 Mbit/s, 3x I²C, 2x CAN, SDIO)
- Analog: two 12-bit DACs, three 12-bit ADCs reaching 2.4 MSPS or 7.2 MSPS in interleaved mode
- Up to 17 timers: 16- and 32-bit running at up to 168 MHz
- Easily extendable memory range using the flexible static memory controller supporting Compact Flash, SRAM, PSRAM, NOR and NAND memories
- Analog true random number generator
- The STM32F417 also integrates a crypto/hash processor providing hardware acceleration for AES 128, 192, 256, Triple DES, and hash (MD5, SHA-1)

3.2.2 MARG Sensors

The give a feedback to the controller, the Quadcopter requires sensors that provide an estimate of its current position. These sensors return the values of the Parameters that are used in the State Estimation using a Kalman Filter. A MARG sensor is a Magnetic, Angular Rate and Gravitational sensor. The term is used within the many publications concerning '9 Degrees of Freedom (DOF) Inertial Measurement Units (IMU)'. The 9 DOF is often used to refer to

- 3 – Axis Accelerometer
- 3 – Axis Gyroscope
- 3 – Axis Magnetometer

The values of these sensors are fused, referred to as ‘Sensor Fusion’, to obtain a more reliable Sensor Reading. These values are used as State Estimation Parameters.

An Accelerometer is a device that measures proper acceleration ("g-force"). When the object it’s integrated into goes from a standstill to any velocity, the accelerometer is designed to respond to the vibrations associated with such movement. It uses microscopic crystals that go under stress when vibrations occur, and from that stress a voltage is generated to create a reading on any acceleration.

A gyroscope is a device that uses Earth’s gravity to help determine orientation. Its design consists of a freely-rotating disk called a rotor, mounted onto a spinning axis in the centre of a larger and more stable wheel. As the axis turns, the rotor remains stationary to indicate the central gravitational pull.

Magnetometer is an instrument used for measuring the strength and sometimes the direction of a magnetic field. An important use of magnetometers is in measuring the earth's magnetic field so as to determine the Orientation of the Device it is mounted on.

3.2.3 Global Position System (GPS)

The Global Positioning System (GPS) is a space-based navigation system that provides location and time information in all weather conditions, anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites. The problem with GPS, however, is that in an indoor environment the GPS returns incorrect coordinates. Many times, the GPS Module is unable to connect to Satellites. This makes GPS an unreliable system for obtaining the current Position of the Device in applications where precision is necessary.

3.2.4 Barometer

A barometer is a scientific instrument used in meteorology to measure atmospheric pressure. Pressure tendency can forecast short term changes in the weather. Numerous measurements of air pressure are used within surface weather analysis to help find surface troughs, high pressure systems and frontal boundaries. In this project, the Barometer could be used to approximate the height of the Device. The barometer, however, is required to be recalibrated in order to suit various environmental changes. Thus, a barometer shall only be used for an approximate of the current altitude of the device.

3.2.5 Ultrasonic Sensors

Ultrasonic sensors transmit ultrasonic waves from its sensor head and receive the ultrasonic waves reflected from an object. By measuring the length of time from the transmission to reception of the sonic wave, the sensors can detect the position of the object from the sensor. This is used in order to detect any sort of an obstacle. In Quadcopters, it could be used to avoid collisions and is thus an important component for maintaining Autonomy.

3.3 Communication

The project focusses on the use of ZigBee protocol to communicate with the Quadcopter wirelessly from the Control Station.

3.3.1 Wireless Communication Modules

The project uses two XBee Modules to set up a communication between the Control PC and the Quadcopter.
XBee is the brand name from Digi International for a family of form factor compatible radio modules. The XBees can operate either in a transparent data mode or in a packet-based application programming interface (API) mode. In the transparent mode, data coming into the Data IN (DIN) pin is directly transmitted over-the-air to the intended receiving radios without any modification. Incoming packets can either be directly addressed to one target (point-to-point) or broadcast to multiple targets (star). This mode is primarily used in instances where an existing protocol cannot tolerate changes to the data format. AT commands are used to control the radio's settings. In API mode the data is wrapped in a packet structure that allows for addressing, parameter setting and packet delivery feedback, including remote sensing and control of digital I/O and analog input pins. For mere communication with the Quadcopter, the XBees are used in AT mode for this project.
The XBee modules used, work on the ZigBee protocol. ZigBee is an IEEE 802.15.4-based specification for a suite of high-level communication protocols used to create personal area networks with small, low-power digital radios. ZigBee is a low-cost, low-power, wireless mesh network standard targeted at wide development of long battery life devices in wireless control and monitoring applications. Zigbee devices have low latency, which further reduces average current. ZigBee chips are typically integrated with radios and with microcontrollers that have between 60-256 KB flash memory. ZigBee operates in the industrial, scientific and medical (ISM) radio bands: 2.4 GHz in most jurisdictions worldwide; 784 MHz in China, 868 MHz in Europe and 915 MHz in the USA and Australia. Data rates vary from 20 kbit/s (868 MHz band) to 250 kbit/s (2.4 GHz band).

3.3.2 Communication Protocol

To ensure a secured communication between the Control Station and the Quadcopter, a custom Communication Protocol was developed. This protocol has been adapted from the Credit Card Numbering System. Every instruction sent from the Control Station has a length of 8 characters.

- The first 3 characters consist of letters. These determine the type of instruction. Based on the priority of the instructions and the current device tasks, the device shall choose to obey the Control or Override it.
- The next 5 characters are numbers. The first 2 numbers are used to determine the Tag Value of the instruction. The tag value represents the parameter, the instruction needs to address.
- The next 2 numbers are used to address a sub-parameter.
- The last character is used as a validation number. This number is generated by the Control Station such that the sum of all the numbers in the instruction is equal to either 9, 18, 27, 36 or 45 whichever lowest possible.

For example, the instruction "PAR12636" is used to instruct the Quadcopter to send the current value of a certain Parameter (PAR). The (12) characters are used to address the Accelerometer. The next two characters (63) are used to refer to the X-Axis value. And the last character validates the string. Thus, the Quadcopter transmits the current X-Axis Value of the Accelerometer to the Control Station.

A basic block diagram of a general Quadcopter has been presented in figure 3.4.

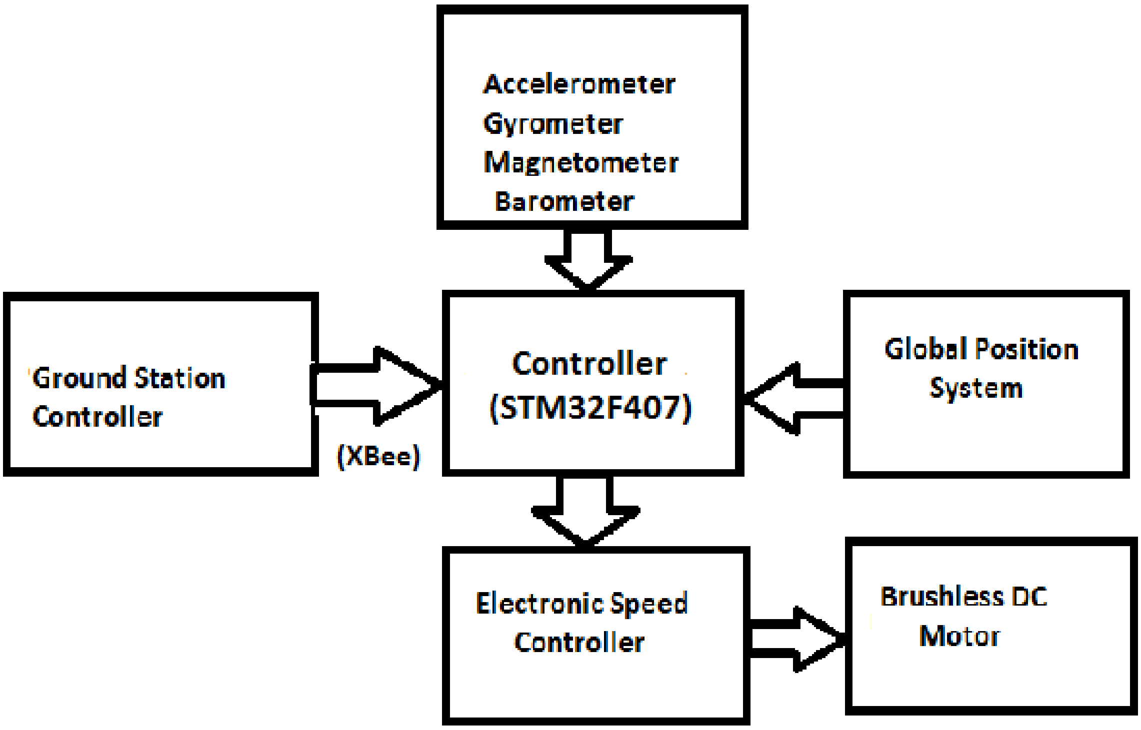

Figure 3.4: Block Diagram of a Quadcopter

Chapter 4

System Implementation

This chapter discusses the current progress of the project and the expected approach for its implementation. Currently, the project is in the implementation stage. The design of the system has been achieved, however, certain significant changes are expected with further revisions.

4.1 System Design

The components to be used for building the structure of the Quadcaopter have been mentioned and discussed in the previous chapter. The current pattern of the project appears to have the following elements.

4.1.1 PC Based Control Station

A Graphical User Interface (GUI) has been developed in Visual C# for Windows Platform to provide the User with an easy control of the Quadcopter. This particular interface is in its preliminary stages and would undergo quite a few changes upon further revisions so as to incorporate more features.

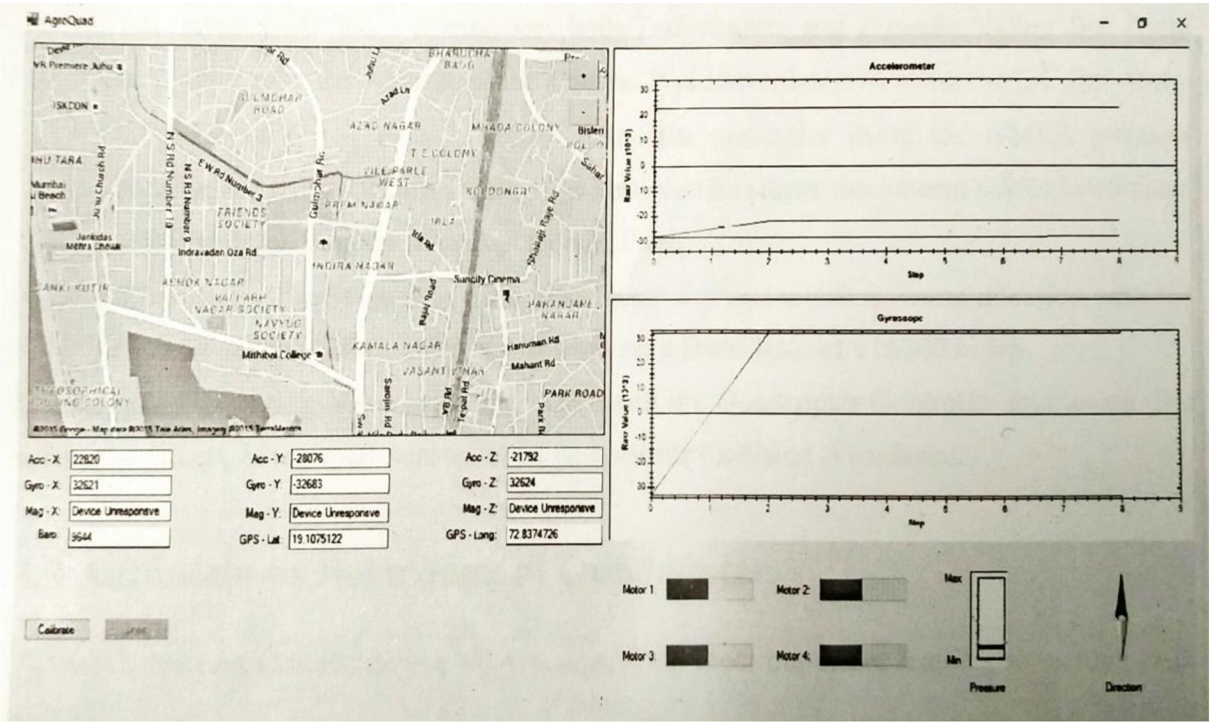

Figure 4.1: Control Window

Upon opening the application, the User is presented with the Welcome Windows which requires the user to select the COM Port where the XBee is attached and its Baud Rate for communication. Upon pressing the Connect button, the User is returned with the Connection Successful Window in case the connection between the PC and the XBee is successful. If not, an Error Message is returned. The application is then redirected to the Main Page where the Control Window is visible. The Control Window consists of all the status parameters and also gives users the ability to send instructions to the Quadcopter.

The Control Window displays the current parameter values. On the Upper Left Corner, the current GPS location is displayed on Google Maps. To the right, a log of Accelerometer and Gyroscope values if presented in the form of a chart. To the Bottom Right, the current speeds of the Motors, Pressure Value from the Barometer and the Direction as obtained from the Magnetometer are all displayed. The Left Bottom Space enables user to control the Quadcopter. There are a few features that are yet to be added to the application and shall be implemented upon further work on the project.

4.1.2 Android based Flight Control

The response of a Visual Studio based app is often slow due to the Heavy Library it implements and a virtual environment that is necessary to run the apps. The update interval observed between every internal timer was a 100ms which made the application non-real time. Hence, to address the issue and introduce a portable Flight Control Station, the Android Platform was used.

The Android app was developed using Android Studio. The design principle used was similar to the PC based Control Station. An XBee was serially connected to the Android Device using the OTG cable connected to the Micro-USB port. A library for Serial Communication was developed and was used to communicate with the XBee Module which in turn would communicate with the Flight Controller Board.

The programming language involved was Java. The Layout of the App was made using XML. The app focusses less on the design aspect and has been programmed considering the timing restrictions to maintain a proper control over the device.

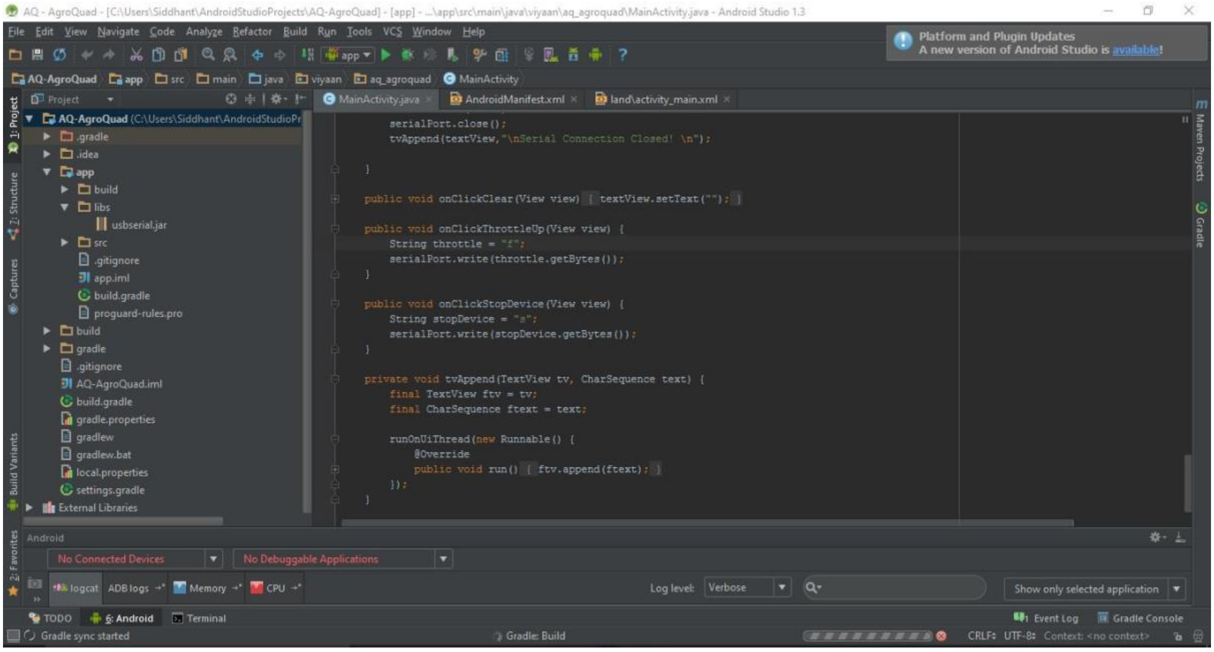

Figure 4.2: Android Studio – App Development Platform

The layout of the App is as shown in the Figure below. A pseudo-real time system has been implemented. It can be experimentally observed that the Andorid Device has a better control over the Quadcopter as compared to the PC based ground station.

Figure 4.3: Android App developed for Flight Control; AQ - AgroQuad

As seen in the *figure 4.3*, the design is highly basic and was used for experimental purposes. This was done in order to maintain the Flight Controller in constant control of the Device instead of servicing routines such as for communication. The App lets the User Enable communication with the device. It also lets user to send a custom command to the flight controller. The two basic functions include the Throttle Up and Stop buttons. As the names suggest, these are used to control the speeds of the Motors. The App also has a terminal pane on the bottom left of the layout which lets the user to observe the status of the Quadcopter.

4.1.3 Modules Connected to the Controller

As mentioned in the previous chapter, the Modules connected to the Controller include:

- 6 DOF (Accelerometer and Gyroscope) MPU6050 – It is interfaced with the controller using an I2C communication protocol. It returns the Accelerometer and Gyroscope Values.
- 3 Axis Magnetometer HMC5883L – It is also interfaced with the controller using an I2C Protocol. It returns the 3 Axis Magnetometer Values that help determine the Device's orientation
- Barometer BMP085 – It measures both Temperature and Pressure values that could help determine the Altitude of the Device. It also communicates using the I2C protocol.
- GPS Ublox 6M – It is interfaced with the controller using the NMEA Protocol specifically used for interfacing GPS Devices. It returns the current GPS Coordinates along with the number of connected satellites.
- XBee-ZB – The Module is used to establish a direct wireless communication with the Control Station. It is configured to work on a Baud Rate of 115200 bits/s.

Currently, these devices have been interfaced with the Quadcopter Controller and tested. For further revisions, Ultraviolet Sensors shall be used for Collision Avoidance.

4.2 Autonomous Behaviour of Quadcopter

To establish a considerable degree of Autonomy, the most important features to include in the system are:

- Way-Point Navigation
- Path Planning
- Rerouting
- Collision Avoidance
- Return Home on loss of Contact

To achieve such a complex structure, the use of Neural Network for Flight Stabilization and Kalman Filter for State Estimation have already been proposed in chapters 1 and 2. This section proposes the manner of implementations of such algorithms.

4.2.1 Flight Stabilization

Unlike the conventional method of implementing PID Control for Flight Stabilization, the project aims at using the CMAC algorithm for such a task. The use of such a Neural Network enables the Device to reduce the Error Values even due to the effect of uncertain Environmental Parameters. The Cerebellar Model Arithmetic Computer (CMAC) is a type of neural network based on a model of the mammalian cerebellum.

The CMAC has been extensively used in reinforcement learning and also for automated classification in the machine learning community. CMAC computes a function $F(X_1 \ldots X_N)$, where n is the number of input dimensions. The input space is divided up into hyper-rectangles, each of which is associated with a memory cell. The contents of the memory cells are the weights, which are adjusted during training. Usually, more than one quantisation of input space is used, so that any point in the input space is associated with a number of hyper-rectangles, and therefore with a number of memory cells. The output of a CMAC is the algebraic sum of the weights in all the memory cells activated by the input point.

A change of value of the input point results in a change in the set of activated hyper-rectangles, and therefore a change in the set of memory cells participating in the CMAC output. The CMAC output is therefore stored in a distributed fashion, such that the output corresponding to any point in input space is derived from the value stored in a number of memory cells. This provides generalisation.

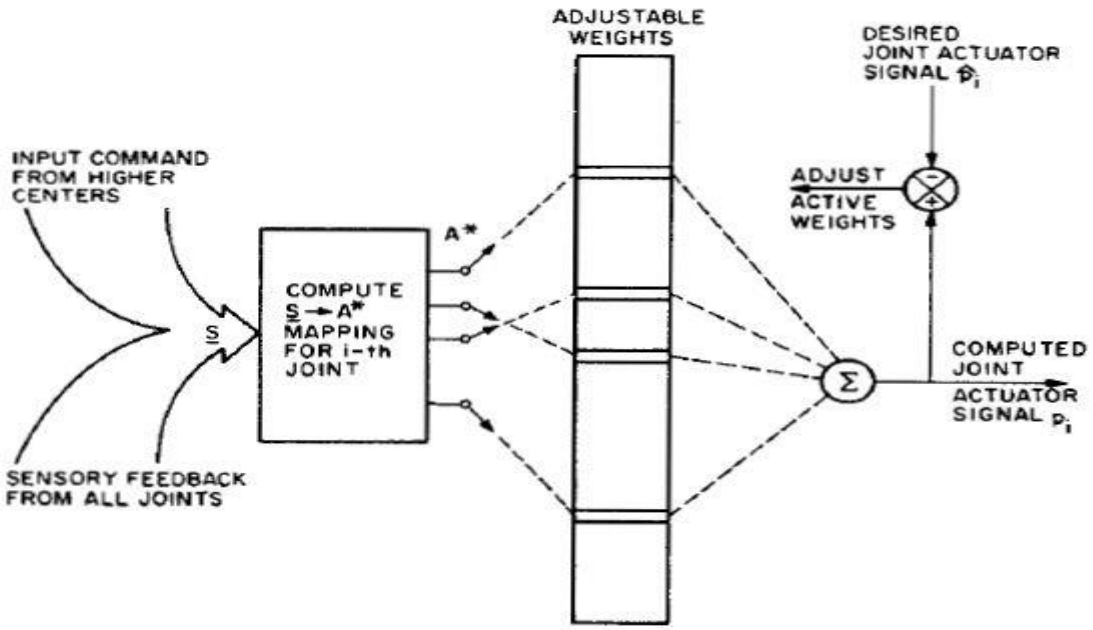

Figure 4.4: Block Diagram of a CMAC Network

4.2.1 State Estimation

The Kalman Filter is a state estimator which produces an optimal estimate in the sense that the mean value of the sum (in fact, of any linear combination) of the estimation errors gets a minimal value. In other words, the Kalman Filter gives a sum of squared errors.

The Kalman filter operates by producing a statistically optimal estimate of the system state based upon the measurement(s). To do this it will need to know the noise of the input to the filter called the measurement noise, but also the noise of the system itself called the process noise. To do this the noise has to be Gaussian distributed and have a mean of zero.

Step 1:

The apriori estimate of the angle is equal to the estimate of the previous state plus the unbiased rate times the delta time. Since, the bias cannot be directly measured, the estimate of the apriori bias is just equal to the previous one.

```
rate = newRate - bias;
angle += dt * rate;
```

Step 2:

```
P[0][0] += dt * (dt*P[1][1] - P[0][1] - P[1][0] + Q_angle);
P[0][1] -= dt * P[1][1];
P[1][0] -= dt * P[1][1];
P[1][1] += Q_gyroBias * dt;
```

Step 3:

```
y = newAngle - angle;
```

Step 4:

```
S = P[0][0] + R_measure;
```

Step 5:

```
K[0] = P[0][0] / S;
K[1] = P[1][0] / S;
```

Step 6:

```
angle += K[0] * y;
bias += K[1] * y;
```

Step 7:

```
float P00_temp = P[0][0];
float P01_temp = P[0][1];

P[0][0] -= K[0] * P00_temp;
P[0][1] -= K[0] * P01_temp;
P[1][0] -= K[1] * P00_temp;
P[1][1] -= K[1] * P01_temp;
```

The following variances have been used for the IMU:

Q_angle = 0.001;

Q_gyroBias = 0.006;

R_measure = 0.03;

The algorithm works in a two-step process. In the prediction step, the Kalman Filter produces estimates of the current state variables, along with their uncertainties. Once the outcome of the next measurement (necessarily corrupted with some amount of error, including random noise) is observed. These estimates are updated using a weighted average, with more weight being given to estimates with higher certainty. The algorithm is recursive. It can run in real time, using only the present input measurements and the previously calculated state and its uncertainty matrix; no additional past information is required.

The Kalman Filter does not require any assumption that the errors are Gaussian. However, the filter yields the exact conditional probability estimate in the special case that all errors are Gaussian-distributed.

Extensions and generalizations to the method have also been developed, such as the extended Kalman Filter and the unscented Kalman Filter which work on nonlinear systems. The underlying model is a Bayesian Model similar to a hidden Markov model but where the state space of the latent variables is continuous and where all the latent and observed variables have Gaussian Distributions.

4.3 Modular Assemblies as Containers for Fertilizers and Pesticides

The use of certain Modules could be made in order to achieve a direct implementation of dumping fertilizers and pesticides. The Modular Assemblies could be attached with the Quadcopter Frame and communicate with the Controller. The controller could thus recognize the Module Parameters such as Weight and Volume. In case the Module is sensed to be empty by the Quadcopter due to weight reduction, the Quadcopter could return Home for replacing the Module with another until the empty Module is refilled. Such a mechanism allows for a higher scope in case of future developments. Thus, the use of Modular Assemblies could be made.

Chapter 5

Control System Design for Quadcopter

This chapter discusses the use of the PID control loop and the timing considerations to execute the loop continuously in order to correct the System every 4ms.

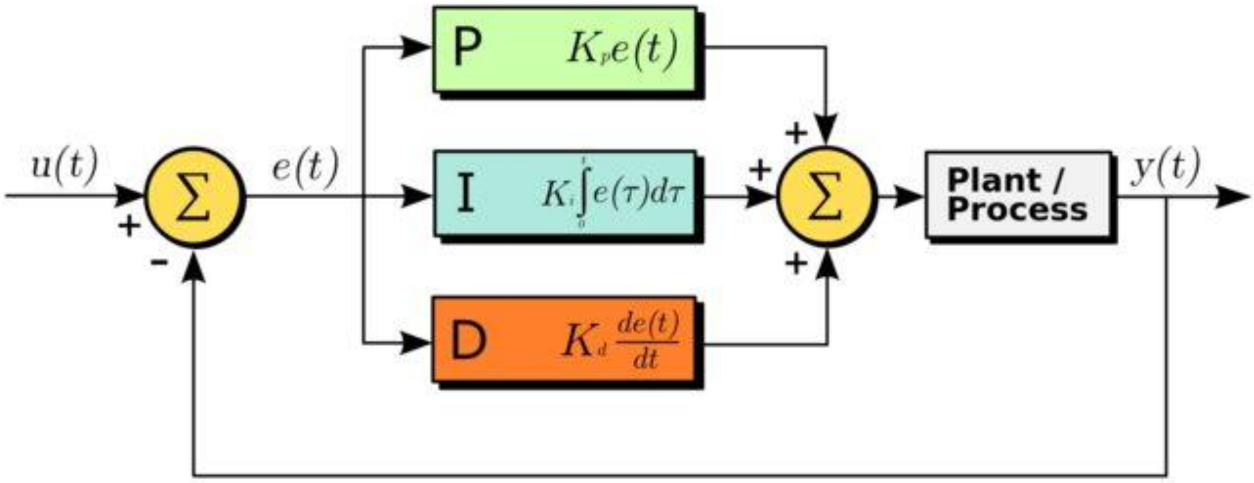

Figure 5.1: Block Diagram of a PID Controlled System

It is required in a Quadcopter System that the PID Gains are fine-tuned so as to witness a stable flight.

5.1 PID Control of a System

PID (proportional-integral-derivative) is a closed-loop control system that attempts to get the actual result closer to the desired result by adjusting the input. Most of the Quadcopters or Multicopters use PID controllers to achieve stability.

There are three algorithms in a PID controller, they are P, I, and D respectively. P depends on the present error, I on the accumulation of past errors, and D is a prediction of future errors, based on current rate of change. These controller algorithms are translated into software code lines.

To have any kind of control over the quadcopter or multicopter, it is needed to be able to measure the quadcopter sensor output (for example the pitch angle), so as to estimate the error. These output correction terms are then applied to the three control algorithms to the error, to get the next outputs for the motors aiming to correct the error.

There are three parameters that a pilot can adjust to improve better quadcopter stability. These are the coefficients to the 3 algorithms that were mentioned above. The coefficient basically would change the importance and influence of each algorithm to the output. The effects of these parameters on the flight are studied.

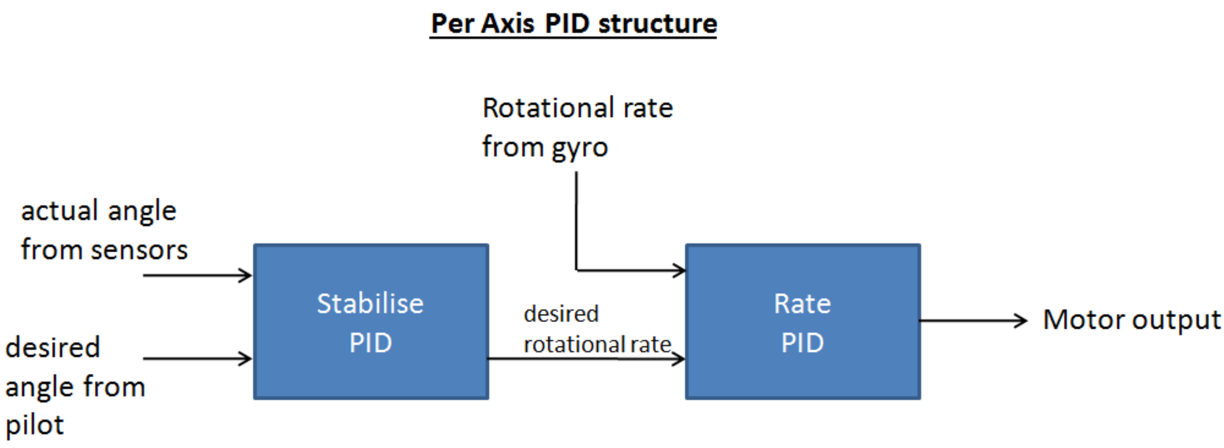

Figure 5.2: PID Loop for Single Axis

5.1.1 Effect of PID Gains

The variation of each of the gain parameters alters the effectiveness of the stabilization. Generally there are 3 PID loops with their own P-I-D coefficients, one per axis, so you as to set P, I and D values for each axis (pitch, roll and yaw).

- **Proportional Gain coefficient** – A quadcopter can fly relatively stable without other parameters but this one. This coefficient determines the significance of, human control or the values measured by the IMU. The higher the coefficient, the quadcopter seems more sensitive and reactive to angular change. If it is too low, the quadcopter will appear sluggish and will be harder to keep steady. The multicopter starts to oscillate with a high frequency when P gain is too high.
- **Integral Gain coefficient** – This coefficient can increase the precision of the angular position. For example when the quadcopter is disturbed and its angle changes by 20 degrees, in theory it remembers how much the angle has changed and will return 20 degrees. In practice if an external force makes the quadcopter go forward and then causes it to stop, the quadcopter will continue for some time to counteract the action. Without this term, the opposition does not last as long. This term is especially useful with irregular wind, and ground effect (turbulence from motors). However, when the I value gets too high the quadcopter might begin to have slow reaction and a decrease effect of the Proportional gain as consequence, it will also start to oscillate like having high P gain, but with a lower frequency.
- **Derivative Gain coefficient** – This coefficient allows the quadcopter to reach more quickly the desired attitude. Some people call it the accelerator parameter because it amplifies the user input. It also decrease control action fast when the error is decreasing fast. In practice it will increase the reaction speed and in certain cases an increase the effect of the P gain.

Aerobatic flight:

- Requires a slightly higher P
- Requires a slightly lower I
- Increase D

Gentle smooth flight:

- requires a slightly lower P
- Requires a slightly higher I
- Decrease D

5.1.2 Tuning Quadcopter PID Gains

For thorough tuning of the PID gains, a trial and error method is used once the Quadcopter is modelled.

For **P gain**, it is started from the lower limit and icreased to the point the Quadcopter appears to oscillate. The gain is then reduced to get a stable flight.

For **I gain**, again, start low, and increase slowly. It is to be noted that the Quadcopter should stabilize quickly with the help of this parameter.

For **D gain**, it can get into a complicated interaction with P and I values. When using D gain, it is needed to go back and fine tune P and I to keep the plant well stabilized.

5.2 Modelling Quadcopter System

The quadcopter is modelled so as to obtain an approximation for the PID Gain Parameters. In order to achieve baseline PID controller gains to control roll and pitch, the equations of motion for the quadcopter were found. The first step in determining the equations of motion is to find the inertia matrix:

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}$$

where I_{xx}, I_{yy}, and I_{zz} are the moments of inertia about the x, y, and z-axes, respectively, and I_{xy}, I_{xz}, I_{yz}, I_{yx}, I_{zx}, and I_{zy} are the products of inertia. Assuming perfect symmetry about the three axes, the products of inertia become zero, and the moments of inertia are defined by the following equations:

$$I_{xx} = \sum_i m_i(y_i^2 + z_i^2)$$

$$I_{yy} = \sum_i m_i(x_i^2 + z_i^2)$$

$$I_{zz} = \sum_i m_i(x_i^2 + y_i^2)$$

where m_i is the mass of a given particle (e.g. motor, esc, battery, etc.), and x_i, y_i, and z_i are the perpendicular distances from the mass to the specified axis. The moments of inertia, I_{xx}, I_{yy}, and I_{zz} are 0.0089, 0.0098, and 0.0187, respectively. After determining each moment of inertia, Lagrange's equation for rotational motion is applied using,

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}}\right) - \frac{\partial T}{\partial q} = 0$$

where *T* is the rotational kinetic energy of the system, and *q* represents the generalized coordinates. In an unconstrained quadcopter, there would be a total of six generalized coordinates, corresponding to x, y, z, φ (roll), θ (pitch), and ψ (yaw). More specifically, when strictly analyzing rotation, the generalized coordinates are reduced to just the Eulerian angles (φ, θ, ψ). However, the test rig used to stabilize the quadcopter constrains the system from moving translating in the x and y-directions, as well as rotating in yaw. Theoretically one can eliminate the yaw rotation without restricting the frame physically by spinning motors 1 and 3 in the clockwise direction and motors 2 and 4 in the counterclockwise direction (see Fig. 9), thus bringing the net moment about the z-axis to zero. During the stabilization process it was also assumed the quadcopter was not translating in the z-direction. Therefore, the only generalized coordinates of concern were φ and θ. Taking the coordinate axis to be at the center of mass, the rotational kinetic energy, T_{rot}.

$$T_{rot} = \frac{1}{2}I_{xx}(\dot{\varphi} - \dot{\psi}\sin\theta)^2 + \frac{1}{2}I_{yy}(\dot{\theta}\cos\varphi + \dot{\psi}cos\theta sin\varphi)^2 + \frac{1}{2}I_{zz}(\dot{\psi}cos\theta cos\varphi - \dot{\theta}sin\varphi)^2$$

Assuming ψ (and therefore dψ/dt) are zero, the following equations of motion are obtained:

$$\text{roll } (\varphi)\text{: } \ddot{\varphi} = \frac{M_\varphi - I_{yy}\dot{\theta}^2 cos\varphi sin\varphi}{I_{xx}}$$

$$\text{pitch } (\theta)\text{: } \ddot{\theta} = \frac{M_\theta + (I_{yy} - I_{zz})2\dot{\theta}\dot{\varphi}cos\varphi sin\varphi}{I_{yy}cos^2\varphi + I_{zz}sin^2\varphi}$$

where *M* is the applied moment. In order to design a controller to stabilize the quadcopter, the equations of motion must be linearized. This can be done by determining the jacobian matrix of the system about a set of specified initial conditions,

$$[\varphi, \theta, \psi, \dot{\varphi}, \dot{\theta}, \dot{\psi}, M_{\varphi}, M_{\theta}] = [0, 0, 0, 0, 0, 0, 1d, 1d]$$

where *d* is the distance from the center of mass to the motors in the y-direction. After applying these initial conditions, the following linearized state space model is obtained:

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

OR

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{1}{I_{xx}} & 0 \\ 0 & 0 \\ 0 & \frac{1}{I_{yy}} \end{bmatrix} [u]$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + [0][u]$$

Inputting the moments of inertia, matrix B becomes

$$B = \begin{bmatrix} 0 & 0 \\ \frac{1}{0.0089} & 0 \\ 0 & 0 \\ 0 & \frac{1}{0.0098} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 112.36 & 0 \\ 0 & 0 \\ 0 & 102.04 \end{bmatrix}$$

According to “Different Linearization Control Techniques for a Quadrotor System,” a control approach involves determining the decentralized equation of motion [Belkheiri]. The equations obtained here are nearly identical to those obtained in the decentralized control section of “Different Linearization Control Techniques for a Quadrotor System,” which further validates their accuracy. It is also important to verify the controllability and observability of the system. This can be done by checking the rank of the controllability and observability matrices

$$CB = [B\ AB\ A^2B\ A^3B]$$

$$OB = [C\ CA\ CA^2\ CA^3]^T$$

It is apparent that the phi and theta equations of motion are decoupled, allowing for two independent controllers to be designed to control roll and pitch individually. Converting the state space representation to transfer functions in the Laplace domain, the following transfer functions are obtained:

$$G_{roll} = \frac{\varphi}{M_{\varphi}} = \frac{1}{I_{xx}s^2} = \frac{1}{0.0089s^2}$$

$$G_{pitch} = \frac{\theta}{M_{\theta}} = \frac{1}{I_{yy}s^2} = \frac{1}{0.0098s^2}$$

If the actual quadcopter were indeed ideal, i.e. perfectly symmetrical and balanced, there would be no need for an integral term in the controller. However, since the quadcopter is not truly symmetrical, the controllers for both roll and pitch will be PID controllers of the form

$$C_{PID} = \frac{k_d s^2 + k_p s + k_i}{s} = \frac{kd(s+a)(s+b)}{s}$$

where k_d is the derivative gain and k_p is the proportional gain. An alternative method of PID controller design however is to first design a PD controller to meet a desired transient response, and then add a PI controller to improve steady-state response. Therefore, the first controller equation that will be used is that of a PD controller,

$$C_{PD} = k_d(s + a)$$

The open loop gain, L, can be determined. Using the equation for a closed loop feedback controller, the characteristic equation of the system can be determined, and set equal to the general form of the characteristic equation:

$$G_{cl} = \frac{L}{1+L} = \frac{\frac{k_d(s+a)}{I_{xx}s^2}}{1+\frac{k_d(s+a)}{I_{xx}s^2}}$$

$$I_{xx}s^2 + k_d s + k_d a = s^2 + 2\zeta\omega_n s + {\omega_n}^2$$

where *a* is the zero location of the PD controller, ζ is the damping ratio, and ω_n is the natural frequency. By selecting a desired damping ratio of 0.8 and a settling time of 0.4 seconds, values of 0.1788, 0.1954, and 7.1825 were obtained for k_d (roll), k_d (pitch), and *a*, respectively.

Chapter 6

Design of the Flight Control Board

6.1 Flight Control Board

A custom Flight Control Board was developed to interface all of the used modules with the Controller, which, in this case was the STM32F407VE based on the Cortex M4 architecture. The layout of the circuit is as shown:

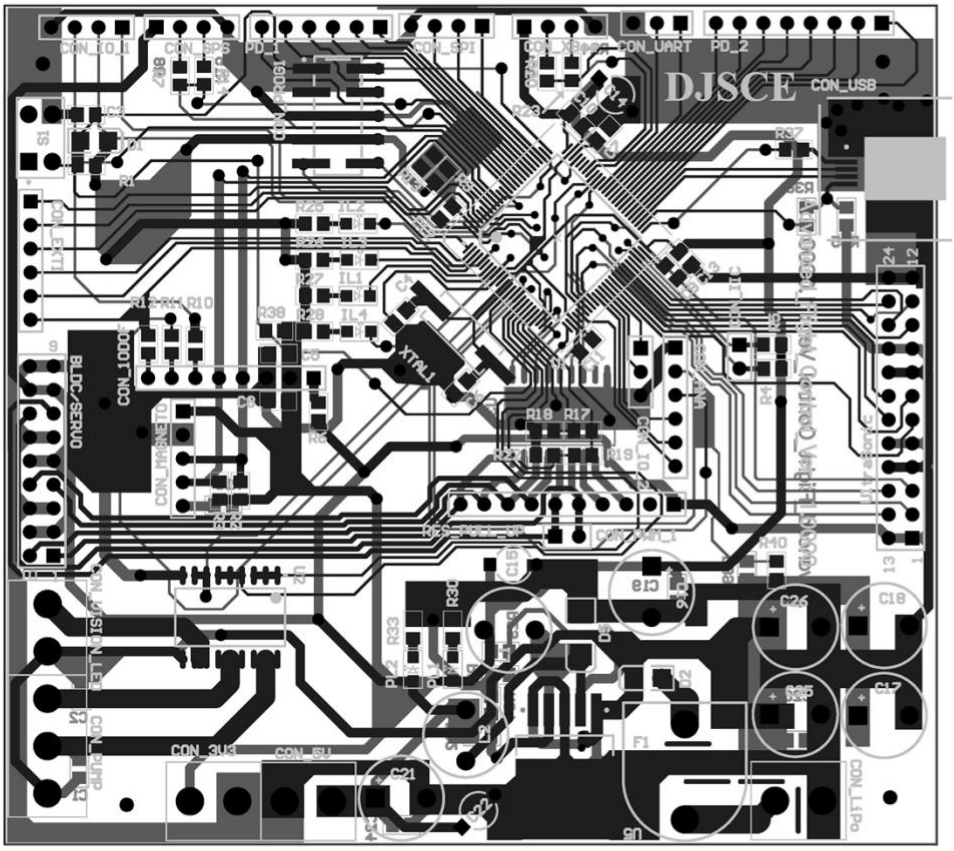

Figure 6.1: Layout for the PCB

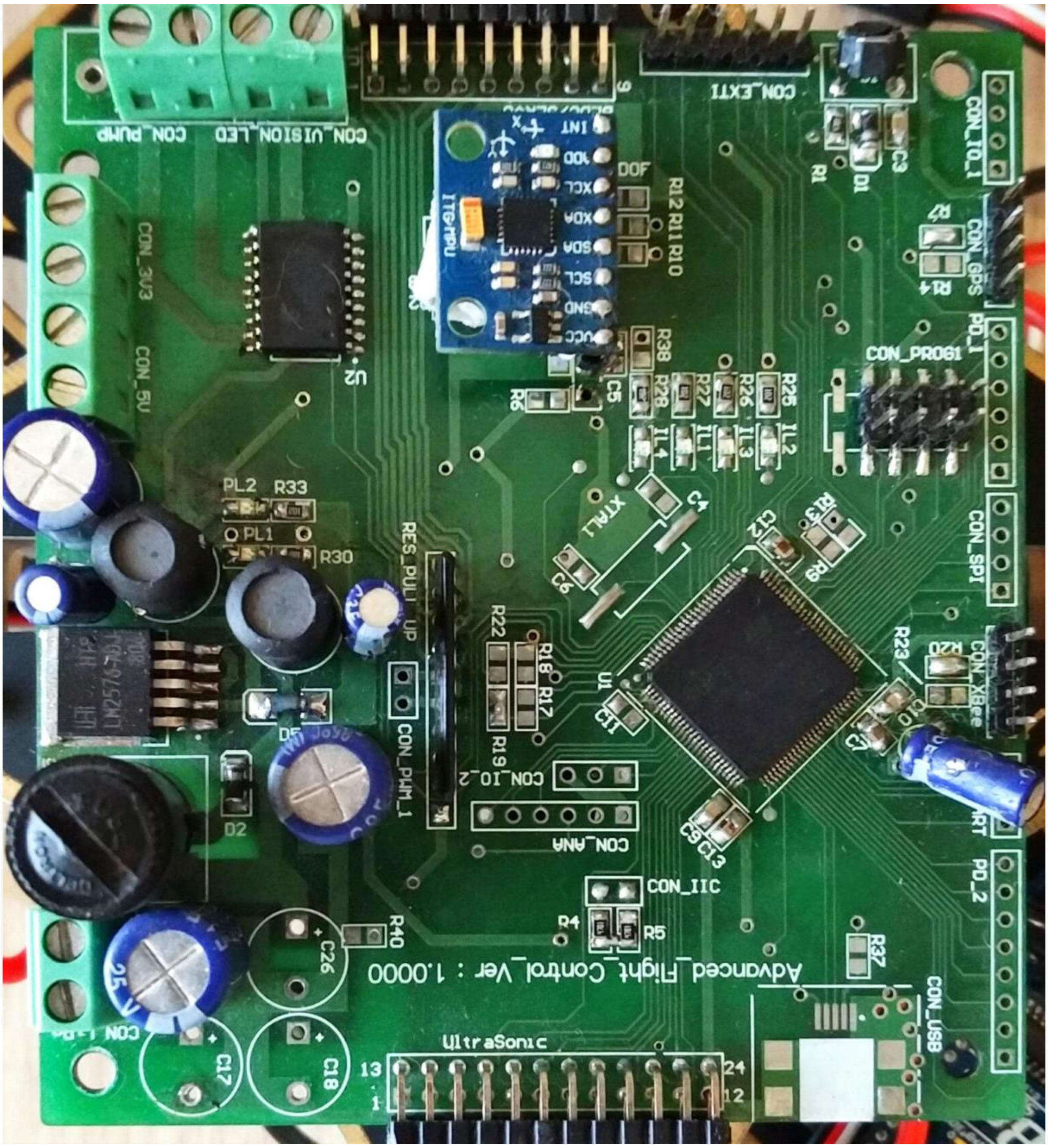

Figure 6.2: Flight Control Board

Chapter 7

Conclusion and Scope

Having studied the technical implications of the project, it could be well understood that a considerable amount of work has been required to achieve an Autonomous model. The State Estimation Algorithm and Flight Stabilization consume the maximum resources but become the base for the Autonomy of the device.

7.1 Conclusion and Scope

The implementation of the Kalman Filter and the CMAC Neural Network require high amount of Mathematical Computation. Hence, a significant amount of time is consumed, servicing and estimating, the state parameters. It is necessary to ensure that such algorithms are scheduled in a manner such that the components connected to the Controller undergo a flexible task scheduling. Failure of an efficient Task Scheduling mechanism results in instability of the Quadcopter.

One of the most important aspects of a Quadcopter is its Mechanical Structure and the ability of the Motors and Propellers to provide sufficient Thrust for a sooth flight. It is necessary to consider the weight of the whole assembly. It is also necessary to have a proper Battery powering the device. It should last for a high amount of time in order to service the Return Home feature.

There is a huge scope ahead for the development of the Quadcopters and its Commercialisation. The development of an Autonomous Quadcopter using the proposed methods could lead to an increase in the adaptability of these devices. The applications of these devices could range from Search and Rescue Operations to providing First Aid Support.

The project aims at achieving the proposed solutions for the Quadcopter. This could not just have a significant impact on the Agriculture industry but also encourage development of technologies for better implementation of these devices. It is expected that the industry award the Quadcopter developers and hobbyists with more efficient motors, compact batteries, and even faster controllers for a better flight experience.

References

[1] S. McGilvray and A. Tayebi, “Attitude stabilization of a vtol quadrotor aircraft,” in IEEE Trans. Control Systems Technology, 2006, vol. 14, pp. 562–571.

[2] E. Altug and B. Erginer, “Modeling and pd control of a quadrotor vtol vehicle,” in Proc. IEEE Intelligent Vehicles Symposium, Istanbul, Turkey, 2007, pp. 894–899.

[3] A. Benallegue and T. Madani, “Backstepping sliding mode control applied to a miniature quadrotor flying robot,” in Proc. IEEE Conf. on Industrial Electronics, Paris, France, 2006, pp. 700–705.

[4] S.Bouabdallah andR.Siegwart, “Backstepping andsliding-mode techniques applied to an indoor micro quadrotor,” in Proc. IEEE Int. Conf. on Robotics and Automation, Barcelona, Spain, 2005, pp. 2247–2252.

[5] E.Altug, J.P.Ostrowski, andC.J.Taylor, “Quadrotor control usingdual camera visual feedback,” in Proc. IEEE Conf. Robotics and Automation, Taipei, Taiwan, 2003, vol. 3, pp. 4294–4299.

[6] S.G. Fowers, D. Lee, B.J. Tippetts, K.D. Lillywhite, A.W. Dennis, and J.K. Archibald, “Vision aided stabilization and the development of a quad-rotor micro uav,” in Proc. IEEE Int. Sym. Computational Intelligence in Robotics and Automation, Jacksonville, Florida, 2007, pp. 143–148.

[7] P.C. Hughes, Spacecraft Attitude Dynamics, New York: Wiley, 1986.

[8] H. Voos, “Nonlinear state-dependent riccati equation control of a quadrotor uav,” in Proc. IEEE Int. Conf. Control Applications, Munich, Germany, 2006, pp. 2547–2552.

[9] S. Bouabdallah and R. Siegwart, Advances in Telerobotics, chapter Towards Intelligent Miniature Flying Robots, pp. 429–440, 2006.

[10] J.Albus, “Anew approach tomanipulator control: thecerebellar model articulation controller (CMAC),” J. Dyn. Sys. Meas. Contr., vol. 97, pp. 220–227, 1975.

[11] J. Albus, “Data storage in the cerebellar model articulation controller (CMAC),” J. Dyn. Sys. Meas. Contr., vol. 97, pp. 228–233, 1975.

[12] C.J.B. Macnab, “Robust direct adaptive control with associative memories in the presence of persistent oscillations,” Neural Information Processing Letters, vol. 10, no. 12, pp. 277–287, Dec. 2006.

[13] C. Coza and C.J.B. Macnab, “A new robust adaptive-fuzzy control method applied to quadrotor helicopter stabilization,” in Proc. North American Fuzzy Information Processing Society Conference, Montreal,Canada, 2006, pp. 454–458.