REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

FAR DISTANCE UNMANNED AERIAL VEHICLES CONTROL AND OBJECT DETECTION USING INTERNET OF THINGS NETWORK AND EMBEDED SYSTEMS

FAHAD YASSIR AL BAZZAZ

MSc. THESIS
DEPARTMENT OF COMPUTER ENGINEERING
PROGRAM OF COMPUTER ENGINEERING

ADVISOR
PROF. DR. HASAN HÜSEYİN BALIK

İSTANBUL, 2017

REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

FAR DISTANCE UNMANNED AERIAL VEHICLES CONTROL AND OBJECT DETECTION USING INTERNET OF THINGS NETWORK AND EMBEDED SYSTEMS

A thesis submitted by Fahad AL BAZZAZ in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE** is approved by the committee, on 18.12.2017 in Department of Computer Engineering, Computer Engineering Program.

Thesis Adviser
Prof. Dr. Hasan Hüseyin BALIK
Yıldız Technical University

Approved By the Examining Committee
Prof. Dr. Hasan Hüseyin BALIK
Yıldız Technical University

Doç. Dr. Sırma YAVUZ, Member
Yıldız Technical University

Doç. Dr. Metin ZONTUL, Member
İstanbul Aydin University

ACKNOWLEDGEMENTS

This thesis is the result of more than 8 months of work and it was supported by many people.
First of all, I would like to thank my supervisor, Prof. Dr. Hasan Hüseyin BALIK. Before being my thesis supervisor, he was one of my lecturers at Yildiz Technical University. As a lecturer, he taught me all the required materials to achieve successfully this project. As a supervisor, although his time schedule was very busy, he has made the effort to be as available as possible to solve my problems. And I want to thank him to his strong ability to listen to others, his answers were always consistent with my queries.
And I would like to dedicate this work to the spirit of my father Dr. Yassir AL BAZZAZ, who died six months ago, he has always provided me with support and assistance. Without him, I would not have reached this position in my life. I also thank my mother Prof. Dr. Nadia SALIM, who always supported me, taught me and give me the power to complete my Master. Also I will never forget to thank the love of my life -my dear wife- who was standing with me in all critical circumstances and gave me all the help that she could give. Moreover, my children who are the joy of my life.
In addition, I would like to thank Turkish people and government for helping us and opening their country and universities to live here as we were living in our country. Finally, I would like to thank everyone who helped me to complete this work.

December, 2017.

Fahad ALBAZZAZ

TABLE OF CONTENTS

LIST OF SYMBOLS

ω	The angular velocity vector in the body frame.
τ	The motor torque.
I	Input current.
I_0	The current when there is no load in the motor.
K_t	The torque relativity constant.
V	The voltage drop across the motor.
R_m	The motor resistance.
K_v	A proportionality constant (indicating back-EMF generated per RPM).
ρ	The density of the surrounding air.
A	The area swept out by the rotor.
C_D	Dimensionless constant.
R	The radius of the propeller.
b	Some appropriately dimensioned constant.
I_M	The moment of inertia about the motor Z-axis.
$\dot{\omega}$	The angular acceleration of the propeller.
b	Drag coefficient.
$\vec{x}$	The position of the UAV.
G	The acceleration due to gravity.
F_D	The drag force.
T_B	The thrust vector in the body frame.

LIST OF ABBREVIATIONS

AFHDS	Automatic Frequency Hopping Digital System
BEC	Battery Elimination Circuit
CCW	Counter Click Wise
CG	Center of Gravity
CW	Click Wise
ESC	Electronic Speed Controller
GPS	Global Positioning System
GUI	Graphical User Interface
IEEE	Institute of Electrical and Electronics Engineers
IMU	Inertia Measurement Unit
IoT	Internet of Things
LiPo	Lithium Polymer
PDB	Power Distributed Board
PID	Proportional-Integral-Derivative
PWM	Pulse Width Modulation
RC	Remote Control
UAV	Unmanned Aerial Vehicles
USV	Unmanned Surface Vehicles

LIST OF FIGURES

Page

LIST OF TABLES

Page

ABSTRACT

FAR DISTANCE UNMANNED AERIAL VEHICLES CONTROL AND OBJECT DETECTION USING INTERNET OF THINGS NETWORK AND EMBEDED SYSTEMS

Fahad AL BAZZAZ

Department of Computer Engineering

MSc. Thesis

Adviser: Prof. Dr. Hasan Hüseyin BALIK

The new generation of Wireless Sensor Networks, that is known as the Internet of Things (IoT) enables the direct connection of physical objects to the Internet using microcontrollers. In most cases these microcontrollers have very limited computational resources. The global connectivity provides great opportunities for data collection and analysis as well as for interaction of objects that cannot be connected to the same local area network.

The Internet of Things (IoT) is a technology that allows objects to be connected to the internet, enabling them with communication capabilities (with other objects and with people). By 2020 there will be over 50 billion things/objects connected to the internet, meaning this technology will completely revolutionize the world as we know it. IoT became an increasingly important research area around the world.

Unmanned Aerial Vehicles (UAV) are becoming more common in our modern world. UAVs are mostly associated with war due to the coverage of their use in the recent wars in Iraq and Syria, but have the ability to do much more. UAVs are helpful tools in assessing damage after a disaster, keeping rescuers safe while they help those in need. UAVs are useful tools in monitoring crops to ensure the maximum yield is realized. The use of UAVs is also being used for monitoring remote land areas that are difficult to reach by foot. However, uses of UAVs are endless.

This dissertation´s goal, is designing a system that uses internet of things (IoT) to make the control for UAV (drone) from an infant distance by simulating the signals coming from RF receiver and remote control, these signals will be provided from Raspberry Pi3 and Arduino Uno devices to apply the signals of Roll, Pitch, Yaw, and Throttle to take control of UAV, the system uses an interaction between raspberry pi and flight controller to overcome the complexity of stability and PID control calculations. The control is designed using python 2.7 GUI (Graphical User Interface) to take control for flight, also the design is containing the capability of tracking objects based on the BGR color moment calculations.

Key words: Internet of Things, Drone, PID (Proportional-Integral-Derivative).

YILDIZ TECHNICAL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

ŞEYLERİN İNTERNET AĞLARI VE GÖMÜLÜ SİSTEMLERİ KULLANAN UZAKTAN İNSANSIZ HAVA ARAÇLARI KONTROL VE NESNE ALGILAMA

Fahad AL BAZZAZ

Bilgisayar Mühendisliği Bölümü

Yüksek Lisans Tezi

Tez Danışmanı: Prof. Dr. Hasan Hüseyin BALIK

Things of Internet (IoT) olarak da bilinen yeni nesil Kablosuz Algılayıcı Ağları, fiziksel nesnelerin mikro denetleyicileri kullanarak Internet'e doğrudan bağlanmasını sağlar. Çoğu durumda bu mikro denetleyiciler çok sınırlı hesaplama kaynaklarına sahiptir. Küresel bağlantı, aynı yerel alan ağına bağlanamayan nesnelerin etkileşimi için olduğu kadar veri toplama ve analizinde de büyük fırsatlar sağlar.
Şeylerin İnternet'i (IoT), nesnelerin internete bağlanmasını, iletişim yetenekleri ile (diğer nesnelerle ve insanlarla) bağlantı kurmasını sağlayan bir teknolojidir. 2020 yılına kadar internete 50 milyarın üzerinde nesne / nesne bağlanacaktır, yani bu teknoloji dünyanın bilindiği gibi tamamen devrim yaratacaktır. IoT, dünyada giderek daha önemli bir araştırma alanı haline geldi.
İnsansız Hava Araçları (UAV) modern dünyamızda daha yaygın hale gelmektedir. UAV'ler, Irak ve Suriye'deki son savaşlardaki kullanımından ötürü çoğunlukla savaşla ilişkilidir, ancak daha fazlasını yapabilecek durumda. UAV'ler bir felaketten sonra hasarı değerlendirmede, kurtarıcıları ihtiyacı olanlara yardım ederken güvende tutmada yararlı araçlardır. UAV'ler, azami verimin alınmasını sağlamak için bitkileri izlemede yararlı araçlardır. UAV'lerin kullanımı, yürüyerek ulaşılması güç olan uzak arazilerin izlenmesi için de kullanılmaktadır. Bununla birlikte, UAV'lerin kullanımı sonsuzdur.
Bu tezin amacı, RF alıcısı ve uzaktan kumandadan gelen sinyalleri taklit ederek UAV (drone) kontrolünü bebeklik mesafeden yapmak için işlerin internet (IoT) kullanan bir sistem tasarlamakta olup bu sinyaller Ahududanı Pi3 ve Arduino Uno cihazlarının UAV kontrolünü ele geçirmek için Roll, Pitch, Yaw ve Throttle sinyallerini uygulamak için

sistem karanlık ve PID kontrol hesaplamaları karmaşıklığını aşmak için ahududu pi ve uçuş denetleyicisi arasında bir etkileşim kullanıyor. Kontrol, uçağın kontrolünü ele geçirmek için python 2.7 GUI (Grafik Kullanıcı Arayüzü) kullanılarak tasarlanmış ve ayrıca tasarım, BGR renk moment hesaplamalarına dayalı nesneleri izleme yeteneğini de içeriyor.

Anahtar Kelimeler: Bir şeylerin interneti (IoT), Drone, PID (Orantılı-İntegral-Türev).

CHAPTER 1

INTRODUCTION

The Internet offers an individual unified system, that qualifies devices to communicate globally by utilizing a set of criterion protocols and connecting several heterogeneous networks - business, academically, polity etc. Internet was initially comprised of email communication and static websites. Nowadays, several forms of Internet applications could be seen throughout, the fraction of numerous distinct part of our lives supports a great number of applications, services and seeking to convent every user's needs irrespective of place and time. There is major "secret" is cryptic beyond the digitalization of the user and all of the user-friendly and automated techniques.

Internet of Things represents a general concept that enables network contrivances to read and accumulate data information from the universe around us. They then participate that data over the Internet where it can be applied and processed for several motivating objectives.

1.1 Literature Review

The Internet of Things is already a well-known term nowadays and it is attractive increasingly important as various types of systems and sensors, it is advanced to assist people in their daily lives. Connected devices number continues to increase worldwide. And also the variety and the applications in the real world are enormous, making it an attractive manufacture to action on [1].

Many associations and firms are active and industrious on modernistic devices and new resolutions to transact with the increase of the joined devices, which is a rapidly rising industry. It has the ability to overturn any singular milieu such as transportations, seeding, industrializations, intelligent houses even entire metropolis. Associations and firms are seeking to make new devices for any conceivable scenario but they are also meliorating security as well as communication protocols. because of costs continue to decrease and claim continues to increase, It is rated that about 0 billion devices were connected in 2015, 20 to 30 bilion devices will be connected by 2020 [2].

With contemporary progressions in technology, it has become more potential to address a larger number of problems concerning accessibility and likewise to monitor systems more thoroughly and to interact with them. The capabilities of the system is established on the use of unmanned surface vehicle (USV) which help with nautical monitoring tasks by using multi-drone swarms [3] for an expanded spatial solutions. The system is able to discover things such as oil spills [4], or it may be used for the general monitoring of environmental conditions.

The Internet of Things can assist to settle many problems in various domains: including intelligent metropolis; IoT implementations are concerning with noise, parking issues, traffic, and illumination monitoring [5]; contingency system for earthquakes [6]; and thoroughness agriculture applications in culture operation optimization [7]. The Internet of Things (IoT) is used to convey information from the sensors and to the actuators.

In the very near future, millions of unmanned aerial vehicles (UAVs), as well known as drones, are prospective to swiftly prevail in various sections of our daily life. They will perform broad-ranging vigors, from conveying packages to submergence into the water for a particular underwater operation [8].

The application of UAVs can be separated into military and urbanized models. The antecedent is intended for non-governmental and governmental purposes; for example, the employment of UAVs in salvage operations and in recovery from large-scale catastrophe events, like as "the great East Japan earthquake" [9], "the natural disasters of Indonesia" [10], and "the earthquake of Nepal" [11].

Nevertheless, in the very near future, UAV will not only be applied for generic preservations and catastrophe assist operations [12, 13] but likewise into many other civilians, governmental and merchant services. Some kindly examples of these are reconnoitring and surveillance [14], native land security [15, 16], generic safety [17], ecological monitoring [18], forest combustion monitoring [19], safeness and border oversight [12], cultivation [13], "Internet delivery" [20, 21], architectural observation [22], "goods transportations" [23, 24] like "Amazon Prime Air" [25]; which is prepared to safely transfer packages to customers during 30 minutes by the use of small UAV. With their countless implementations, Drones will shortly be an influential part of our life and will comprise an indispensable technology that is identical to the smartphones of todays.

Furthermore, it is interesting to note that, from a technology perspective, UAVs are expected to form a significant component of a sophisticated cyber-physical IoT ecosystem [26]. The qualifier of the IoT is that its aim to enable the thing to be connected all the time and everywhere, and to ideally use any network to provide any service. The concept of the IoT permit UAVs to be a complementary portion of the IoT infrastructure. These are due to the actuality that UAVs have individual features; they are easy-to-prevail, dynamic, easy-to-reprogram over run-time, capable of gauge anything anywhere, and "capable of flying in a controlled air-space with a high degree of autonomy" [27].

1.2 Objective of Thesis

This thesis proposes a case study for designing a system that uses an Internet of Things (IoT) network to make the control for Unmanned Aerial Vehicle (drone) from an infant distance by simulating the signals coming from RF receiver and remote control.

These signals will be provided from raspberry pi device to apply the signals of **Roll**, **Pitch**, **Yaw**, and **Throttle** to take control of UAV. The system uses an interaction between raspberry pi and flight controller to overcome the complexity of stability and PID control calculations.

The control is designed using python 2.7 GUI (Graphical User Interface) to take control for flight, also the design is containing the capability of tracking objects based on the BGR colour moment calculations.

1.3 Original Contribution

This thesis proposes a case study for designing a system that uses an Internet of Things (IoT) to make the control for Unmanned Aerial Vehicle (drone) from an infant distance by simulating the signals coming from RF receiver and remote control, these signals will be provided from raspberry pi device to apply the signals of **Roll**, **Pitch**, **Yaw**, and **Throttle** to take control of UAV, the system uses an interaction between raspberry pi and flight controller to overcome the complexity of stability and PID control calculations. The control is designed using python 2.7 GUI (Graphical User Interface) to take control for flight, also the design is containing the capability of tracking objects based on the BGR color moment calculations.

1.4 Background to the IoT

Communication is the ultimate pivotal aspect of realizing the IoT as different devices need to be able to communicate if they are to interconnect. All other properties, such as the ability to capture, sense, maneuver, process data, and store, are needless; except if the device specifically seeks one of these properties. However, the mastery to communicate is fundamental in recognizing a device as an IoT device. This communication is outright minimal important, as the current materialistic with link layer communication then IoT could be recognized in various ways.

In Figure (1.1) status C demonstrates that devices are not permanently desired in order to communicate meanwhile a communication network. For epitome, if two devices are nigh to each other it may be easier to communicate straightly, for example, through radio or by anticipating technologies such as ZigBee or Bluetooth (protocols that enable frontal communication). By contrast, in status A in Figure (1.1) a device may communicate via a gateway by using one protocol and the gateway could then communicate by using other protocol (IPv4) through the communication network like the Internet. Status B in Figure (1.1) clarifies 2 devices that are directly communicating together without need a gateway, and where both devices are directly connected to the communication network and are therefore able to communicate, even if they are situated in various places.

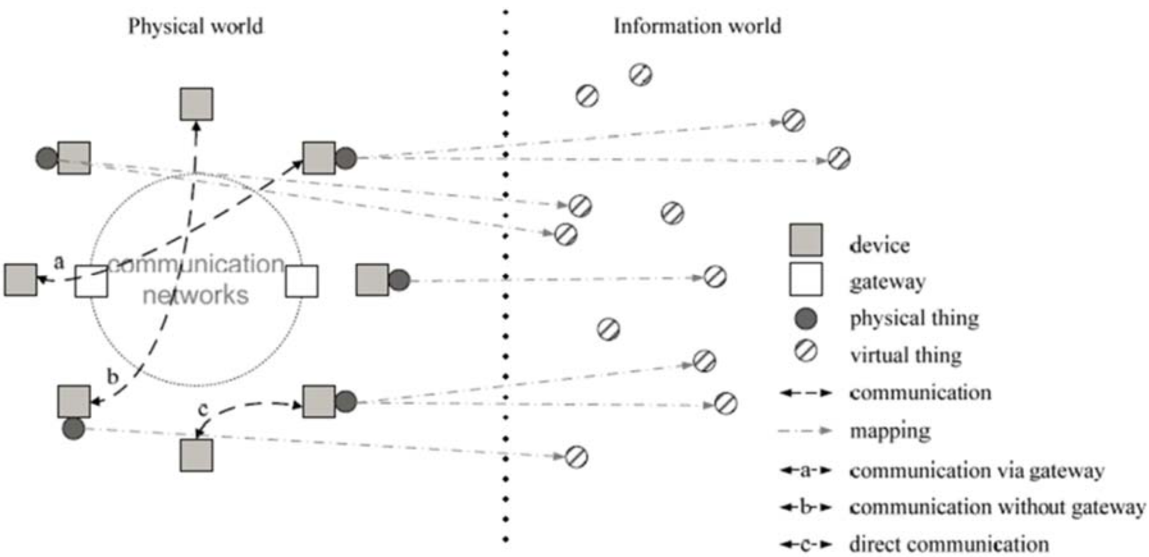


Figure 1.1 Overview of the Internet of Things [28].

A materialistic thing it could be mapped into the information sphere by one or more virtual things, whereas virtual things do not needs to be correlated with any materialistic thing and can subsist independently of any materialistic object. For epitome, a materialistic thing could implement various applications and just like that have vaeious identities in the virtual universe. Identically, a virtual thing could have many identities in the virtual universe too. For epitome, on a USB-drive a virtual thing it can be a video file. Like this file could have various file names that indecate to it, and also it could have multiple patterns (copies), which could likely have various encodings, decision, etc. Table (1.1) provides some substantial features of the IoT. These features might furnish an obvious representation of the genuine variations between IoTs and other devices [28].

Table 1.1 Features of the Internet of Things

Characteristic	Description
Interconnects	Everything could be attached to the universal information and telecommunication infrastructure.
Related - things services	It provides during the compulsion of things, like confidentiality and semantic uniformity between virtual and physical thing.
Heterogeneity	Devices in IoT have various hardware and anticipate various networks, but nevertheless, they can react with other devices over various networks.
Dynamic changes	The situation of a device can alternate dynamically, therefore the number of devices may differ.
Enormous scale	The number of operating devices and the number of communicating devices will be “greater than the number of devices in the existing Internet”.

The requisite feature of the IoT is interconnectivity since the entire concept is constructed onto the concept of being capable to interconnect everything.

In order to equip semantic uniformity, a physical thing that reports temperatures at some period it can be mapped to a virtual thing that seeks to evaluate the temperature among measurements, and it may therefore determine a distinct temperate value than the physical value. When the following gauge comes the virtual device may not update its appreciation in order to keep uniformity with the physical thing.

In Table (1.1) the greatest defy shall be to support heterogeneity as there are many diversified protocols in use. Reacting with multiple devices over multiple networks will be defying from both technical perspectives and a security, as the protocols may vary depending on the interface through which the device is communicating. Therefore, certain requirements are pertinent for IoT, like privacy protection and security. This is particularly true as there will be more data and services available and an increasing number of an effectiveness will depend on this information. Security also implicates privacy considerations, since data collected by means of a sensor, for example, may include sensitive personal information [28].

For privacy reasons, a thing that reports the geographical location can insert noise to its position, which means that the physical location can differ in comparison to the virtual location. This prevents an exact location being mapped to the device, which protects its spatial privacy.

1.5 Unmanned Aerial Vehicles

Drone, as a definition is an aircraft without human on board, drones some times are called unmanned aerial vehicles or unmanned aircraft system, in another point of view it is a robot that flies.

These aircrafts may be controlled remotely or can be autonomous through flight system that use software on an embedded systems using sensors and GPS.

In the past years, the main use of drones was related military applications,

The drones were used mainly in anti-aircraft target practice and weapon platforms , now days they are used in civilian applications like search, rescue, weather monitoring , traffic monitoring, delivery services …etc.

The drone (plane shaped) called "Queen Bee" is considered as the first drone used, which was supplied with a radio signal for controlling the servos-operated parts.

The plane could be piloted from the seat in front, but it flews unmanned and it was shot by gunners in training.

In 2012, a man called Chris Anderson retired from his work to dedicate himself to a new drone company, 3D Robotics. The company started by constructing a personal drone for hobbyist. Nowadays it is marketing its solutions for film and photography companies, telecom businesses…etc.

In 2013, amazon was a leader in the use of drones in delivery activities, since then the others started to use the drones for the same purpose like State University and Virginia Polytechnic Institute in testing Project Wing, to make deliveries starting with burritos produced.

The most common use for drones in drone surveillance and drone journalism, the most important thing in drone is it can be used to access locations that is impossible for human to get and it can do things a human can not do.

The process of drone education is expanded, for example Embry-Riddle Aeronautical University now offers a Bachelor degree in unmanned systems and it is applications, also it gives a master degree for unmanned systems.

In 2016 Business Insider BI Intelligence forecasted that an increase in drone sectors in revenues and shipment by 2021, will reach 29 million shipments in worldwide.

Integration between drones technology with the IoT sensor networks can produce a great help for agricultural companies to monitor areas, lands and crops, also it can produce a great achievement to the energy companies for monitoring power lines and operational equipment. Insurance companies can take benefit from drones in properties monitoring and claims.

1.6 Thesis Outline

This thesis contains of five chapters. The first chapter about general concepts of IoT and related works with ours. The second chapter about hardware components that been used in our work. The third chapter we explained which materials used in our work and how assembled. In fourth chapter the quad copter simulation and PID parameter are showed. And the fifth chapter is the conclusion of thesis.

CHAPTER 2

HARDWARE COMPONENTS

This thesis proposes a case study for designing a system that uses an Internet of Things (IoT) to make the control for Unmanned Aerial Vehicle (drone) from an infant distance by simulating the signals coming from RF receiver and remote control, these signals will be provided from raspberry pi device to apply the signals of **Roll**, **Pitch**, **Yaw**, and **Throttle** to take control of UAV, the system uses an interaction between raspberry pi and flight controller to overcome the complexity of stability and PID control calculations. The control is designed using python 2.7 GUI (Graphical User Interface) to take control for flight, also the design is containing the capability of tracking objects based on the BGR color moment calculations.

2.1 UAVs Components

There are many different types of frames and formations that used to create UAVs.

2.1.1 UAV Frame Types

The most famous types of UAV frames are:

1. **Tricopter:** In this type, the UAV has 3 arms, every arm of which is tied to one motor. The frontage of the UAV inclines to be among two of the arms (Y3). The angle among the arms can be any degree, but usually and most likely is to be 120 degrees, as shown in Fig (2.1).
   The Advantages of this type is that flying is similar to an airplane in the forward motion. The price is theoretically the lowest as it utilizes the lowest number of brushless motors. The disadvantages are that the design utilizes a normal servo motor to rotate the rear motor. Which means that it is not symmetrical due to tha

copter. The design is less modest than that of several other multi-rotors. The rearward arm is more intricate because the servo motor requests to be riding along the axis. In integration, this configuration are not supported in all flight controllers.

Figure 2.1 Tricopter frame.

2. **Quadcopter:** In this type, a UAV has 4 arms, with each arm tied to one motor. The frontage of the UAV is generally among two of the arms (× configuration), but it can to be likewise over an arm (+ configuration), as shown in Fig (2.2). The advantages of this type is that it is the simplest construction and is the most commonly found. In the standard configuration, the arms per motors are symmetrically arranged around two axes. All the flight controllers in the markets are able to work with this UAV design, which is the most popular UAV design. The disadvantage of this type is that there is no redundancy, which means that the craft will probably crash if there is a deficiency anywhere in the system, particularly in a motor or fan.

Figure 2.2 Quadcopter frame.

3. **Hexacopter:** In this type, UAV has 6 arms, and each arm is tied to one motor. The frontage of the UAV may be among two of the arms, but it may also be along one arm, as shown in Fig 2.3 . The advantage of this type is that it increases the total thrust available, which means that the UAV can elevate more payload. In addition, the UAV still has a chance of landing rather than crashing if a motor fails. Virtually “all flight controllers support this configuration”. The disadvantage of this design is that it utilize extra components. This makes it larger and more expensive that the quadcopter, which utilizes a minimum number of components bur has the same motors and fans. These supplemental motors and components increase the weight to the UAV, which means that the battery also has to be larger in order to achieve the same flight time as a quadcopter.

Figure 2.3 Hexacopter frame.

4. **Octocopter:** The UAV octocopter design has eight arms, each of which is tied to one motor. The frontage of the UAV tends to be among any two neighboring arms, as shown in Fig (2.4). The advantages of this type is that it has more thrust and also increased redundancy because of uses more motors. The disadvantages are that it is more expensive and a larger battery is needed.

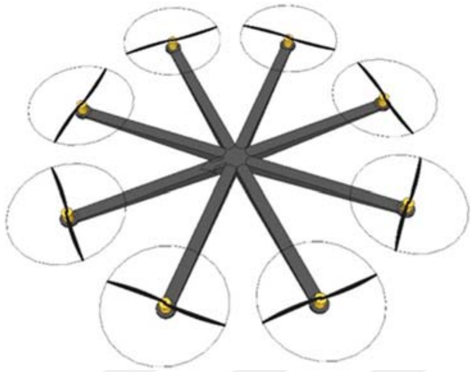


Figure 2.4 Octocopter frame.

2.1.2 UAV Frame Size

There are many different sizes of UAVs frames, from "Nano" which is smaller than your hand, to "Mega", which can only be conveyed by the truck. For most users who are getting commenced in the field, a good size range which offers the most flexibility its size is between 350mm to 700mm. "This measurement represents the diameter of the largest circle which intersects all of the motors" [33].

In this thesis we used quadcopter frame with size 450mm, because of simplest construction, symmetric around two axis, uses a minimum number of parts, cheaper from other frames, low in weight and easy to find it in local markets.

2.1.3 Motors

The motors used have significant effect on the payload (or extreme load) that the UAV can prop, and also on the flighting time. We vigorously propose utilizing the same (propulsion) motor everywhere. It should be noted that even if a couple of motors are in the same trademark and model, and from the same production run, their speeds may partially differ, which is the flight controller will attend to. Figure 2.4 demonstrates some of the types of motors available.

Figure 2.5 Some types of motors.

2.1.3.1 Brushless compared to Brushed

Brushed motors spinning the coil inside a case, with fine-tuned magnets mounted concerning the outside of the casing. Brushless motors do the antagonistic; their coil is fine and are tuned to either the outer casing or to the inner casing whilst the magnets are spun. In ultimate statuses, one only considers brushless DC motors. "Pancake" "brushless motors have a larger diameter and are essentially flatter", and are often used sanction for higher torque and lower KV (for further details, see Section 2.1.3.3). More minute UAVs (conventionally to the size of the palm of your hand) incline for utilize minuscule-brushed motors due to their minimal price and flat to the two-wire controller. While brushless motors come in an assortment of different sizes and specifications, culling more minute brushless motor infrequently designates it will be minimal extravagant. Brushed DC motors own 2 connectors: one for positive and the other is for negative. Inverting the wires means that reverses the rotation of the motor. Brushless DC motors owns 3 connectors. Section 2.1.5 on ESC descrides how to connect that wires and how to invert the direction of rotation.

2.1.3.2 In runner vs Out runner

There are a slight kinds of brushless DC motors:

- **In runner:** These have finely-tuned coils mounted to the outward casing, while the magnets are mounted to the armature shaft that spins inner the casing (they are likely to be utilized on RC cars due to the high KV).
- **Out runner:** In these the magnets are mounted on the outward casing, which is spun around the finely-tuned coil in the center position of the motor casing (the lower mounting of the motor is finely-tuned).

- **Hybrids out runner:** Out runners technically have a immobile outward shell around them in order to make them look homogeneous in comparison to in runners.

In runner brushless DC motors are likely to be utilized in helicopters and airplanes due to their high KV. İn addition, they may be prepered down to increment the torque. Out runners are likely to have more torque.

2.1.3.3 KV

KV it refers to a motor’s rpm constant, and it “is the number of rotations per minute that the motor will turn when 1V is applied with no load annexed to the motor”. In summary, we called it “revs per volt” but do not cerebrated and one will obtain those revs when one affixes a fan, and distinctly “the revs will be reduced” by the load. KV is similar to a motor’s potency or more helpfully, its torque level of a motor. It is tenacious by the number of winds on the armature “or turns as we sometimes call it” and the vigor of the magnets, there are a lot of variables with electric motors. KV enables one to determine how to gain a handle on the torque one could anticipate from a specific motor. The KV is rating per value of a motor imputes to how expeditiously it will rotate at a given voltage. A low KV (for example, from 500 to 100) is required for most multirotor aircraft as this assists with stability. However, for acrobatic flights one may consider a KV of among 1000 and 1500, and one would consider utilizing more minute diameter fans. If the KV classifying of a specific motor is 650 rpm per V, then at 11.1V, the motor will be rotating at 11.1V x 650 = 7215 rpm. If one operates the motor at a minimal voltage (verbalize, 7.4V), the rpm will be 7.4V x 650 rpm / V = 4810 rpm. It is consequential to note that utilizing a minimal voltage is likely to make the current draw will be higher “power = current x voltage “.

2.1.3.4 Thrust

Several brushless motor manufacturers provide a designation of the motor’s thrust that corresponds to some propeller options (and are often approaching in a table). The “unit of thrust is often Kg, Lbs or N”. For epitome, if one is need to build a quadcopter and finds that a categorical motor can provide up to 0.5 Kg of thrust with an 11 inch fans, that signifies that 4 of those motors can hoist 0.5 Kg * 4 = 2 Kg at extreme thrust. Ergo if one’s quadcopter weights just less than 2 Kg, it will only slip off at maximum thrust. One will therefore be required to either cull a motor + propeller amalgamation, which will furnish more thrust, or else minimize the weight of the aircraft. If the propulsion system “all motors and props” can provide 2 Kg of thrust “max” then one’s unharmed copter should not be more than about half this weight (1Kg, inclusive the weight of the motors themselves).

2.1.4 Propeller (Fans)

Fans "propellers" for multi-rotor aircraft are convenient from the "propellers" used in RC airplanes. The material(s) utilized to make the propellers can have a equinoctial influence on the flight features, but safety should be the primary consideration, as can be seen in Figure 2.5 . There are three types of propellers (plastic, Fiber-Reinforced Polymer, Natural such as wood).

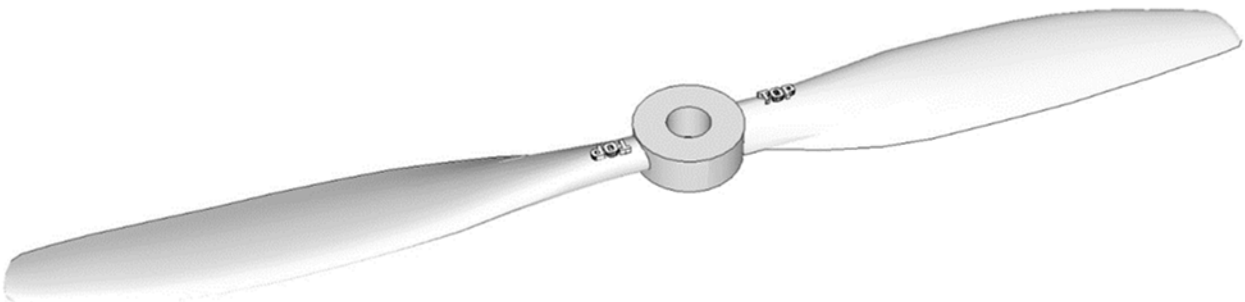


Figure 2.6 UAV Fan.

2.1.4.1 Blades & Diameter

Ultimate multi-rotor aircraft have either 2 or 3 rotor blades, with two being the ultimate common. İt should not be surmised that integrating more blades will automatically means more thrust. Each blade must follow in peregrinate through the wake of the one, which precedes it, which means that the more bladesthere are, the more predominant the wake will be. A propeller with a more diminutive diameter propeller has less inertia and is ergo more facile to expedite and decelerate, which avails in acrobatic flights.

2.1.4.2 Rotation

Propellers are either intended to “rotate clockwise (CW) or counter-clockwise (CCW)”. It is consequential to know which component of the propeller is designed to face upwards (as the top flatness is curved outwards). The top of the propeller should always face the sky.

2.1.5 ESC (Electronic Speed Controller)

An ESC “Electronic Speed Controller” enables the flight controller to control a motor’s haste and direction. The ESC must be able to treat the maximum current, which the motor might consume, and must be able to furnish it at correct and right voltage, Figure 2.6 shows an ESC.

Figure 2.7 an ESC.

Connectors of an ESC:

An ESC may initially be embarrassing as it has numerous wires that exit on 2 sides. These include the following:

- **The Power Inputs:** The 2 solid wires (normally red and black) are intended to gain power from the “power distribution board”, which itself get its power straightly from the main battery.
- **3 black wires connectors:** These pins connect to the 3 pins on the brushless motor.
- **3 - pin R / C servo connector:** This connector accepts RC signals, however requiring 5V on the black and red pins, most of the time an internal BEC provides 5V to power the electronics.

Battery Elimination Circuit “BEC”:

Ultimate ESCs comprise what is called a “Battery Elimination Circuit or BEC”. This is because emanates from the fact that historically, only one brushless motor was needed for a given RC vehicle, and rather than of splitting the battery, it simply had “to be connected to the ESC”, which may have “an onboard voltage regulator to power the electronics”. It is consequential to know the current which an ESC’s BEC can provide, although this is normally in the domain of 1A or more, and is virtually always 5 V. In a multi-rotor, all the ESCs have to be connected the flight controller, but only one BEC is required, which must have power emanating from multiple sources. However, the fact that these are all fed to the same lines can potentially purpose issues. Given that "there is normally no way to deactivate a BEC on an ESC", it is best to take off the red wire and enwrap it with electrical tape for all but one ESC. It remains consequential to leave the black (ground) wire in place for "common ground".

2.1.6 Battery

Chemistry: The batteries utilized in UAVs are now virtually exclusively "Lithium polymer (LiPo)", with some more exotic batteries being "Lithium-Manganese" and other "Lithium variations". "Lead acid" is simply not an option and "NiMh / NiCd" are still too cumbersome for their capability and predominately cannot provide the high discharge rates needed. "LiPo" offer low weight with high capacity, and high discharge rates.

Voltage: in reality only one battery pack should be considered for an UAV, and this "battery's voltage should correspond with the motors chosen". Virtually all the batteries currently utilized are "lithium-based and incorporate a number of 3.7V cells", where 3.7V = 1S. Consequently, a battery that is labeled as 4S is likely 4 x 3.7V = 14.8V nominal. However, providing the number of cells will help one to determine which charger to utilize. A "single cell high capacity battery" may look physically very similar to a "low capacity multi-cell battery".

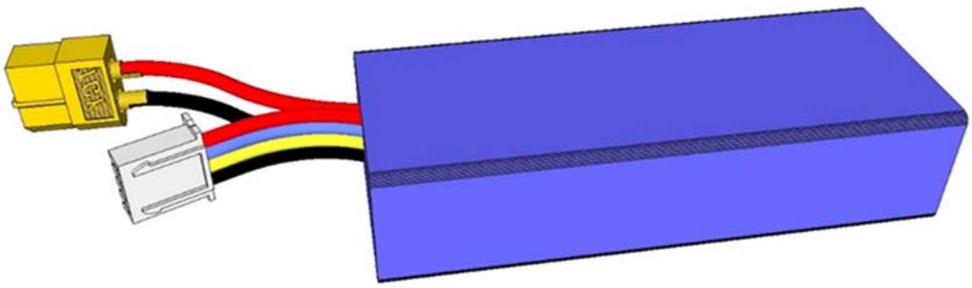


Figure 2.8 UAV Battery.

2.1.7 Flight Controller

A flight controller for a multi-rotor UAV is an integrated that is usually composed of a "sensors, microprocessor, and input / output pins". A flight controller, as shown in Figure 2.8, does not automatically know one's categorical UAV type or configuration. This means that certain set parameters are required in the software program, and the configuration is then uploaded to the board. This is perferred to simply comparing the flight controllers that are currently available.

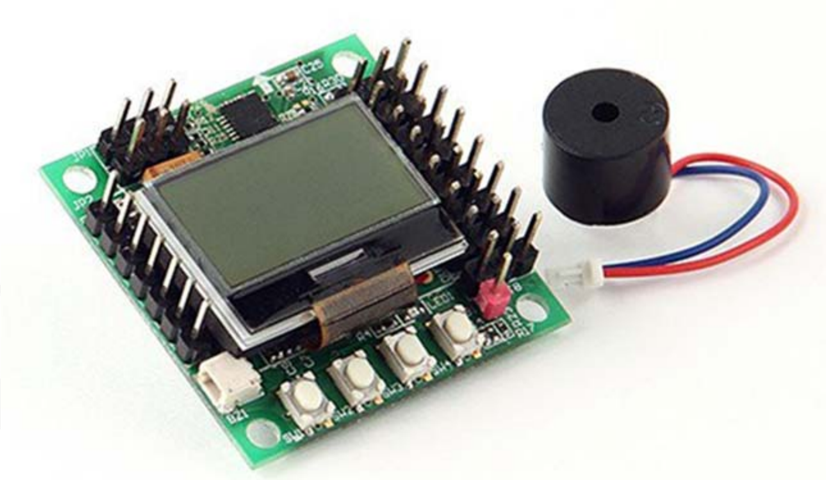


Figure 2.9 Flight Controller.

2.1.7.1 Main Processor

“8051 vs AVR vs PIC vs ARM”: These microcontroller families compose the fundamentals of ultimate existing flight controllers. Arduino is “AVR based (ATmel)” and the community appears to prefer “MultiWii” as its preferred code. Microchip is the essential manufacturer of PIC chips. It is complicated to say that one is better than the other, and it is the software’s abilities that are really more important. ARM (for epitome, STM32) uses “16/32-bit architecture”, while “AVR and PIC” tend to use “8 / 16-bit”. As singular board computers become minimal costly, one can anticipate to see a new generation of flight controllers that are able to “run full operating systems such as Linux and Android”.

“CPU”: These are usually in multiples of 8 “(8-bit, 16-bit, 32-bit, 64-bit)” and refer to the size of the CPU’s primary registers. “Microprocessors can only process a set (maximum) number of bits in memory at a time”. “The more bits a microcontroller can handle, the more accurate (and faster) the processing will be”. For example, “processing a 16-bit variable on an 8-bit processor is rather slow”, while “on a 32-bit processor it is very fast”. It is important to note that the “code also needs to work with correct number of bits”.

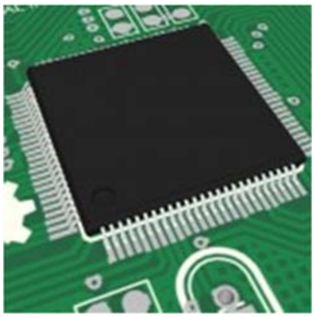


Figure 2.10 Main Processor in flight controller.

“Operating frequency”: This refers to the “frequency at which the main processor operates”. The frequency is measured in “Hertz (cycles per second)”, which is also “commonly referred to as the clock rate”. The higher the operating frequency, the faster data can processed.

“Program Memory / Flash: The flash memory is essentially where the main code is stored”. If the program is complex, it may occupy considerable space. Obviously, “the greater the memory, the more information it can store”. Memory is also useful when “storing in-flight data such as GPS coordinates, flight plans, automated camera movement“, etc. The code loaded to the flash memory remains on the chip even if the power is cut.

“SRAM”: SRAM stands for “Static Random-Access Memory”, and is “the space on the chip that is used when making calculations”. The data stored in RAM is lost when the power is cut. The “higher the RAM, the more information” will be “readily available” for calculations at any given time.

“EEPROM: Electrically Erasable Programmable Read-Only Memory (EEPROM) is normally used to store information which does not change in flight, such as settings, unlike data stored in SRAM which can relate to sensor data” etc.

“Additional I/O Pins: Most microcontrollers have a lot of digital and analog input and output pins, and on a flight controller, some are used by the sensors, others for communication and some may remain for general input and output. These additional pins can be connected to RC servos, gimbal systems, buzzers and more”.

"A/D converter: Should the sensors used onboard output analog voltage (normally 0-3.3V or 0-5V), the analog to digital converter needs to translate these readings into digital data. Just like the CPU, the number of bits which can be processed by the A/D determines the maximum accuracy. Related to this is the frequency at which the microprocessor can read the data (number of times per second) to try to ensure no information is lost. It is nevertheless hard not to lose some data during this conversion, so the higher the A/D conversion, the more accurate the readings will be, but it is important that the processor can handle the rate at which the information is being sent".

2.1.7.2 Power

"There are generally two voltage ranges qualified in the spec sheet of a flight controller, the first being the voltage input range of the flight controller itself (most operate at 5V), and the second being the voltage input range of the main microprocessor's logic (3.3V or 5V). Since the flight controller is an obviously integrated unit, and attention must be taken to the input range for the flight controller itself. Most UAV flight controllers operate at 5V since that is the voltage provided by a BEC".

2.1.7.3 Sensors

In terms of hardware, "a flight controller is basically a normally programmable microcontroller but has specific sensors onboard. At a minimum, a flight controller will have a three-axis gyroscope, but as such will not be able to auto-level. Not all flight controllers will have all of the sensors below and maybe include a combination of there".

The sensors:

1. **Accelerometer:** Is measure linear acceleration in up to three axes (let's call them X, Y and Z), Fig (2.10) shows the accelerometer axis's. The units are in "gravity (g) which is 9.81 meters per second per second, or 32 feet per second per second". The output of an accelerometer can be "integrated twice to give a position, though because of losses in the output, it is subject to drift". A very important characteristic of three axis accelerometers is that they detect gravity, and as such, can know which direction is "down". This plays a major role in "allowing multirotor aircraft to stay stable". The "accelerometer should be mounted to the flight controller so that the linear axes line up with the main axes of the UAV".

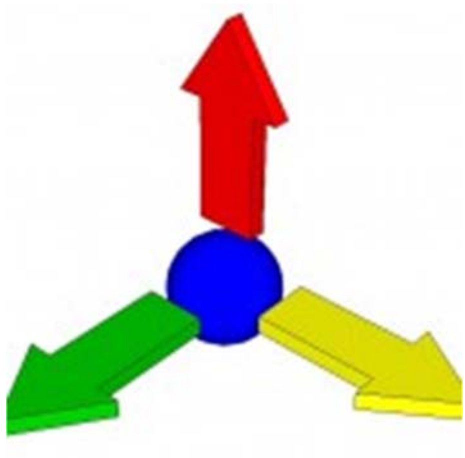


Figure 2.11 Accelerometer.

2. **Gyroscope:** A gyroscope measures the rate of angular change in up to three angular axes (let's call them alpha, beta and gamma), Fig (2.11) shows the gyroscope angular axis's. The units are often degrees per second. Note "that a gyroscope does not measure absolute angles directly, but we can iterate to get the angle which, just like an accelerometer, is subject to drift. The output of the actual gyroscope tends to be analog or I2C, but in most cases you do not need to worry about it since this is handled by the flight controller's code". The gyroscope should be mounted so that its rotational axes line up with the axes of the UAV.

Figure 2.12 The Gyroscope Angular Axis's

3. **Inertia Measurement Unit (IMU):** An IMU is "essentially a small board which contains both an accelerometer and gyroscope (normally these are multi-axis), as shown in Fig (2.12). Most contain a three axis accelerometer and a three-axis gyroscope, and others may contain additional sensors such as a three axis magnetometer, providing a total of 9 axes of measurement".

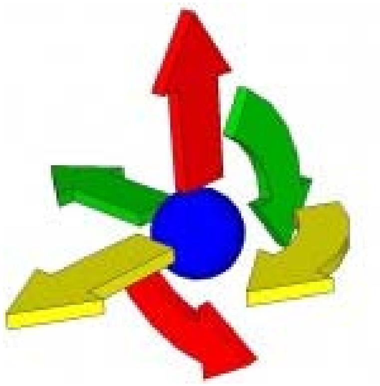


Figure 2.13 Inertia Measurement Unit (IMU).

4. **Compass / Magnetometer:** An electronic magnetic compass is able to measure the earth's magnetic field and used it to determine the UAV's compass direction (with respect to magnetic north), Fig (2.13) shows Magnetometer. This sensor is "almost always present if the system has GPS input and is available in one to three axes".

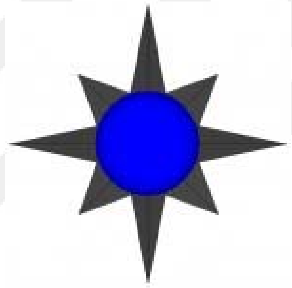


Figure 2.14 Magnetometer.

5. **Pressure / Barometer:** Since atmospheric pressure changes according to the altitude from sea level, "a pressure sensor can be used to give you an accurate reading of the UAV's height. Most flight controllers take input from both the pressure sensor and GPS altitude to calculate a more accurate height above sea level. Note that it is better to have the barometer covered with a piece of foam to minimize the effects of wind over the chip".
6. **GPS:** Global Positioning Systems (GPS) use "the signals sent by a number of satellites in orbit around the earth in order to determine their specific geographic location". A flight controller can either have onboard GPS or one which is connected to it via a cable. The GPS antenna should not be confused with the GPS chip itself, and can look like a small black box or a normal "duck" antenna. In

order to get an accurate GPS lock, “the GPS chip should receive data from multiple satellites, and the more the better”.

7. **Distance:** Distance sensors are “being used more and more on drones since GPS coordinates and pressure sensors alone cannot tell you how far away from the ground you are (think hill, mountain or building) or if you will hit an object”. The distance of downward-facing sensor is based on ultrasonic, laser or lidar technology (infrared has issues in sunlight). Slight flight controllers include distance sensors as part of the standard package.

2.2 Raspberry Pi

The first model (Raspberry Pi 1 Model B) was produced in 2012. Then it was followed by another model, it was simple and cheap in comparing to the previous Model A. In early 2014, the Foundation produced another board with more improved design, which is Model B+. the main advantage of these boards is that they approximately credit-card sized , another generation is A+ and B+ models were produced in one year later. After that, another Module was produced in early 2014 and it is used for embedded applications. A new generation Raspberry Pi 2 produced in early 2015 added more RAM for it.

In early 2017, Raspberry Pi 3 Model B was introduced as the newest version of Raspberry Pi.

All models includes “an on-chip graphics processing unit (GPU, a Video Core IV), ARM compatible central processing unit (CPU) and CPU speed starts from 700 MHz and ends at 1.2 GHz for the Pi 3 and on board memory starts from 256 MB and ends at 1 GB RAM. A Digital (SD) cards are used to store the program memory and operating system in Micro SDHC sizes. Most boards have one or more (four) USB slots, composite video output, HDMI, and a 3.5 mm jack for audio output or input. A number of GPIO pins provide Lower level output, which supports common used protocols like I^2C. The B-models have an Ethernet port and the Pi 3 has an on board Wi-Fi 802.11n and Bluetooth”.

The Raspberry Pi hardware can be summarized in the following diagram:

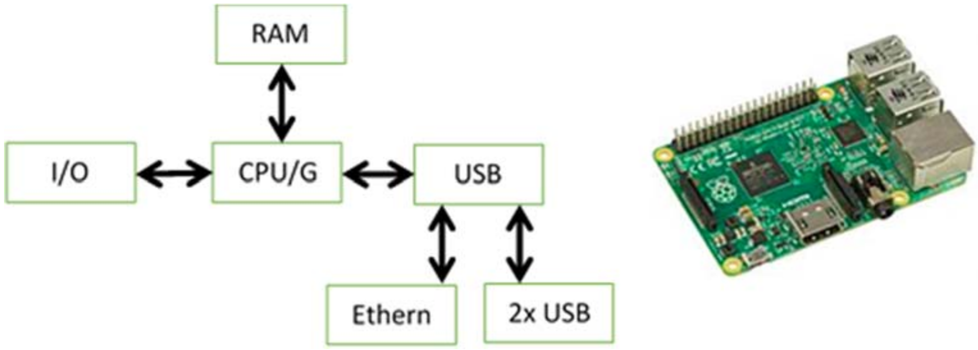


Figure 2.15 Raspberry Pi Hardware Components and B+ Model.

2.2.1 Processor

The main processor of raspberry pi 3 is a 64-bit quad-core ARM Cortex-A53 with 1.2 GHz a Broadcom BCM2837 SoC.

2.2.2 Performance

The Raspberry Pi 3, using a quad-core Cortex-A53 processor performance is 10 times the performance of a Raspberry Pi 1.[33] Benchmarks showed that Raspberry Pi 3 is approximately 80% faster than the Raspberry Pi 2 in parallelized applications.[34]

Raspberry Pi 2 includes a 1 GB RAM quad-core Cortex-A7 CPU,900 MHz. It the Benchmarks showed that it is 4–6 times more efficient than its previous processor. its GPU is the same as the original.[35] In parallelized benchmarks, the Raspberry Pi 2 is said to be up to 14 times faster than its previous model.[36]

2.2.3 RAM

On the older beta Model B boards, “128 MB was allocated by default to the GPU, leaving 128 MB for the CPU”. [37] On the first 256 MB release Model B (and Model A), three different splits were possible. The default split was 192 MB (RAM for CPU), which should be enough for standalone “1080p video decoding, or for simple 3D, but probably not for both together. 224 MB was for Linux only, with only a 1080p framebuffer, and was likely to fail for any video or 3D. 128 MB was for heavy 3D, possibly also with video decoding (e.g. XBMC)”. [38] Comparatively “the Nokia 701 uses 128 MB for the Broadcom VideoCore IV”. [39]

For the later Model B with 512 MB RAM "initially there were new standard memory split files released(arm256_start.elf, arm384_start.elf, arm496_start.elf) for 256 MB, 384 MB and 496 MB CPU RAM (and 256 MB, 128 MB and 16 MB video RAM). But a week or so later the RPF released a new version of start.elf that could read a new entry in config.txt (gpu_mem=xx) and could dynamically assign an amount of RAM (from 16 to 256 MB in 8 MB steps) to the GPU, so the older method of memory splits became obsolete, and a single start.elf worked the same for 256 and 512 MB Raspberry Pis".[40]

"The Raspberry Pi 2 and the Raspberry Pi 3 have 1 GB of RAM".[41][42] The Raspberry Pi Zero and Zero W have 512 MB of RAM.

2.2.4 Peripherals

Figure 2.16 Raspberry Pi 3.

In order connect peripherals, the current Model B boards uses four USB ports. "Raspberry Pi can be operated with generic USB computer keyboard or mouse". [43] It can also be used with USB to MIDI converters, USB storage, and any other devices with USB capabilities. Other devices can be connected through the various pins and connectors on the Raspberry Pi.[44]

2.2.5 Real-time clock

The available Raspberry Pi models do not have a real-time clock, so they are cannot keep tracking the time of the day independently. Some methods can be used to overcome this problem such as, some programs running on the Pi may have the ability to retrieve the time from a network time server or from the user at boot time, thus knowing the time while powered on.

To insure correct real-time tracking "A real-time hardware clock with battery backup, such as the DS1307, which is binary coded, could be added to the system. Raspberry Pi 1 Models A+ and B+, Pi 2 Model B, Pi 3 Model B and Pi Zero (and Zero W) GPIO J8 have a 40-pin pinout".[45, 46]

Table 2.1 Raspberry Pi pinout.

GPIO#	2nd func.	Pin#		Pin#	2nd func.	GPIO#
	+3.3 V	1		2	+5 V	
2	SDA1 (I²C)	3		4	+5 V	
3	SCL1 (I²C)	5		6	GND	
4	GCLK	7		8	TXD0 (UART)	14
	GND	9		10	RXD0 (UART)	15
17	GEN0	11		12	GEN1	18
27	GEN2	13		14	GND	
22	GEN3	15		16	GEN4	23
	+3.3 V	17		18	GEN5	24
10	MOSI (SPI)	19		20	GND	
9	MISO (SPI)	21		22	GEN6	25
11	SCLK (SPI)	23		24	CE0_N (SPI)	8
	GND	25		26	CE1_N (SPI)	7
(Pi 1 Models A and B stop here)						
EEPROM	ID_SD	27		28	ID_SC	EEPROM
5	N/A	29		30	GND	
6	N/A	31		32		12
13	N/A	33		34	GND	
19	N/A	35		36	N/A	16
26	N/A	37		38	Digital IN	20
	GND	39		40	Digital OUT	21

"Model B rev. 2 also has a pad (called P5 on the board and P6 on the schematics) of 8 pins offering access to an additional 4 GPIO connections". [47]

Table 2.2 Model B additional 4 GPIO connections.

Function	2nd func.	Pin#		Pin#	2nd func.	Function
N/A	+5 V	1		2	+3.3 V	N/A
GPIO28	GPIO_GEN7	3		4	GPIO_GEN8	GPIO29
GPIO30	GPIO_GEN9	5		6	GPIO_GEN10	GPIO31
N/A	GND	7		8	GND	N/A

Models A and B provide "GPIO access to the ACT status LED using GPIO 16. Models A+ and B+ provide GPIO access to the ACT status LED using GPIO 47, and the power status LED using GPIO 35".

2.2.6 Operating systems

Figure 2.17 Back view of Raspberry Pi 3 shows MicroSD.

The operating system on raspberry pi can be installed on MicroSD card, we can see from figure (2.16) above the MicroSD slot on the bottom of raspberry pi 2 or 3 board.

The most recommended operating system for raspberry pi by its foundation are Raspbian, a Debian-based Linux operating system. Also a third party operating systems such as include Ubuntu MATE, Snappy Ubuntu Core, Windows 10 IoT Core, RISC OS are available for raspberry pi2 and 3. Many other operating systems can also run on the Raspberry Pi.

2.3 Arduino

Arduino is an open-source hardware. Layout and production files for all versions of the hardware are available in company website. The source code for the IDE is released.

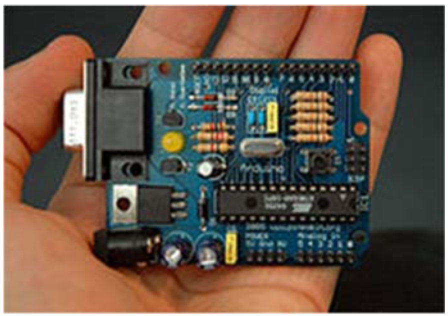


Figure 2.18 RS-232 serial interface.

“An early Arduino board” [18] started with a RS-232 serial interface, which can be seen in Figure 2.17 above. It uses an Atmel ATmega8 microcontroller chip (the black chip in the lower left side in Figure above). The 14 digital I/O pins are located at the top, while the six analog input pins are located on the lower right hand side, and the power connector is found on the lower left hand side of the figure above.

Arduino microcontrollers can be programmed using “a boot loader that simplify” the process of uploading programs to the on-chip flash memory. The boards are programed “with the code via a serial connection to a computer”. Some Arduino boards there are a level shifter circuit to convert between RS-232 logic levels and (TTL) level signals. The Current Arduino boards now days are programmed using (USB), the connection between arduino and computer is made using cable, in some cases standard AVR in-system programming (ISP) programming is used.

Figure 2.19 Arduino Uno.

An official Arduino Uno R2 with descriptions of the I/O locations

The Arduino Uno[c] board provides 14 digital I/O pins, six of them can produce pulse-width modulated signals, and it has six analog inputs, also it can be used as six digital I/O pins. These pins are located on the top of the board, via female 0.1-inch (2.54 mm) headers.

A program for Arduino is “written in any programming language for a compiler that produces binary machine code for the target processor. Atmel provides a development environment for their microcontrollers, AVR Studio and the newer Atmel Studio”. [48, 49, 50]

The Arduino project provides “the Arduino integrated development environment (IDE), which is a cross-platform application written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus”.

A program written with “the IDE for Arduino is called a sketch. [51] Sketches are saved on the development computer as text files with the file extension .ino. Arduino Software (IDE) pre-1.0 saved sketches with the extension” .pde.

Using specific rules of code structuring Arduino IDE supports the languages C and C++. The Arduino IDE supplying a software library from the Wiring project, “which provides a lot of input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in the hexadecimal encoding that is uploaded into the Arduino board by an uploader program in the board's firmware. There are many free public libraries for developers to use to boost their projects”.

CHAPTER 3

ASSEMBLY AND PROCEDURE

3.1 UAV chosen components

According to the information given in the previous chapter (2), in this thesis we have chosen F450 Quadcopter frame with 1400KV brushless motors, EMAX ESC 30A, KK2 Flight controller, RC remote control 6 channels and polycarbonate propellers.

3.2 UAV Assembly

In F450 frame we get 4 arms and two boards as shown in Fig (3.1).

Figure 3.1 F450 UAV Frame Used.

Lower board is printed board (Power Distributed Board (PDB)) and we connect the main input power from battery to the PDB and connect all ESC's to the PDB too. The PDB have 4 connectors as an output and we connect it to the 4 ESC's and we connected the input to the battery, the PDB used mostly to split the main battery to each of the ESCs. The voltage is supplied to the ESCs at the same level, so there is no need to increase it (step up) or decrease it (step down) as shown in Fig (3.2) and Fig (3.3).

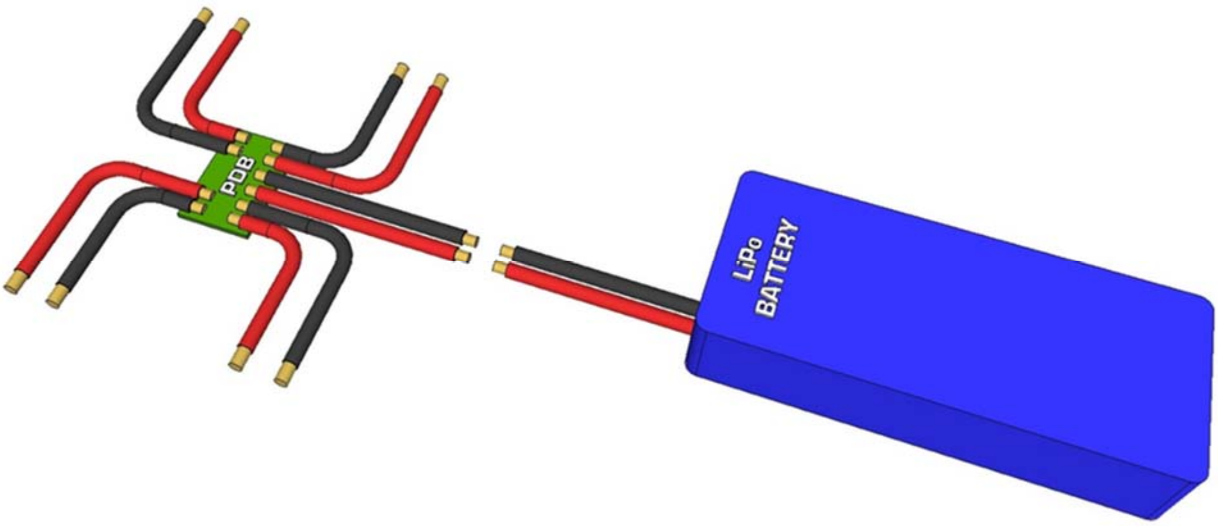


Figure 3.2 Power Distribution Board to Battery.

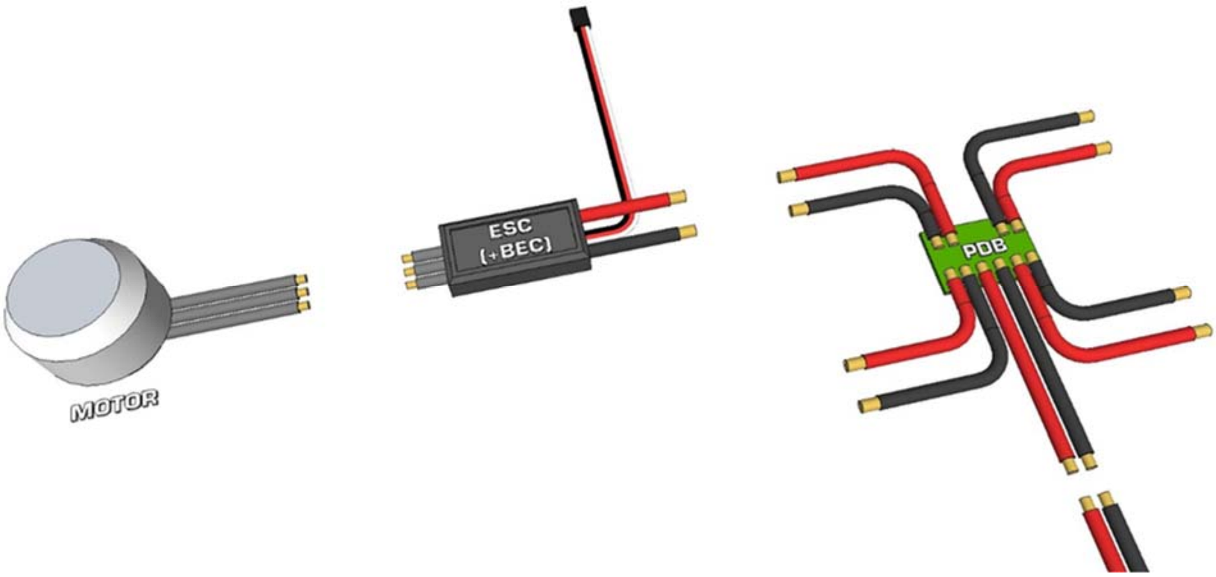


Figure 3.3 Motor to ESC (with BEC) to Power Distributed Board.

After attaching the motors on the frame body, we connect it to the ESCs, each motor to its related ESC. The direction of motor is important for that reason every two opposite motors are in the same direction and the other two in the reverse direction [52], we have two direction clockwise (CW) and counter-clockwise (CCW). For the CW direction, the connection between motor and ESC is direct connect that mean left side from motor goes to left side from the ESC, the middle from the motor goes to the middle from the ESC and right side from the motor goes to the right side from the ESC. For the CCW direction, the connection between motor and ESC is in revers attach that mean left side from motor side goes to right side from the ESC, middle side from the motor goes to the middle side from the ESC, and right side from the motor goes to left side from the ESC as shown in fig (3.4).

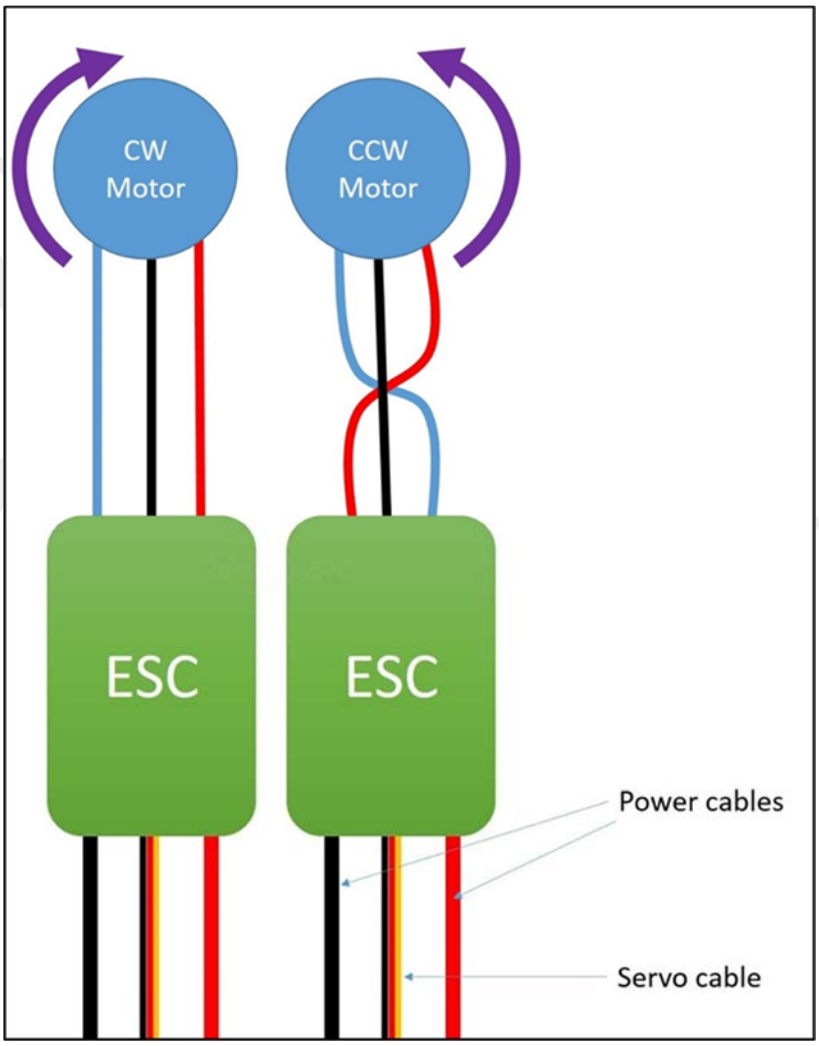


Figure 3.4 Connection types between motor and ESC.

If the motors are all in the same direction and due to Newton’s third low of motion “**for every action, there is an equal and opposite reaction.**” As of it, the body of the quadcopter will tend to spin in the opposite directional to the motors. Therefore, the direction of all motors must be as in Fig (3.5).

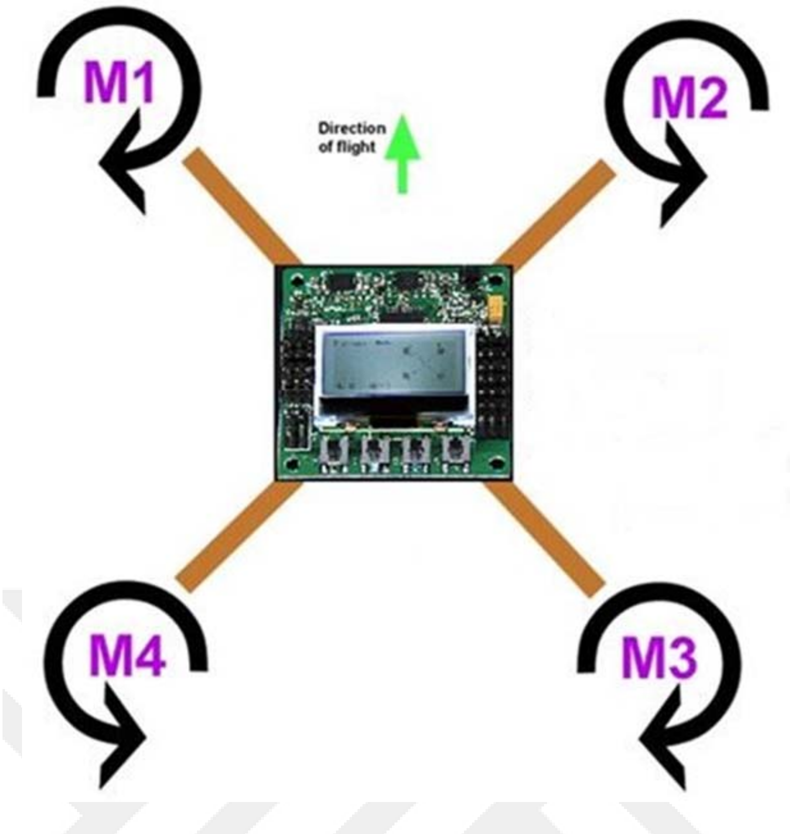


Figure 3.5 direction of UAV motors.

In our UAV we used An EMAX ESC (Electronic Speed Controller) as shown in Fig (3.6) below.

Figure 3.6 EMAX ESC 30 A.

The 3 pins servo connectors goes to flight controller, these wires goes to flight controller and receives the PWM (Pulse Width Modulation) from flight controller to control the rotational speed of the motor.

3.3 Pulse Width Modulation (PWM)

PWM is a technique uses digital circuits to control analog circuits, by generating analog signals from digital devices. PWM used in a wide variety of applications, measurements, communications, conversion and power controls.

Analog systems used to “generate a lot of heat because they are basically variable resistors carrying a lot of currents. While digital systems do not generate much of heat, approximately all the generated heat by a switching device is during the transition, while the device is neither on nor off, but in between”. This is because power follow the following formula:

$$\boldsymbol{P = E \times I} \quad \text{Or} \quad \boldsymbol{Watts = Voltage \times Current} \qquad (3.1)$$

If any of “voltage or current is near zero” then the “power will be near zero too”, PWM takes advantage of this fact. A PWM signal can defined its behavior by two main components duty cycle, and frequency. The PWM waveform are shown in Fig (3.7) below.

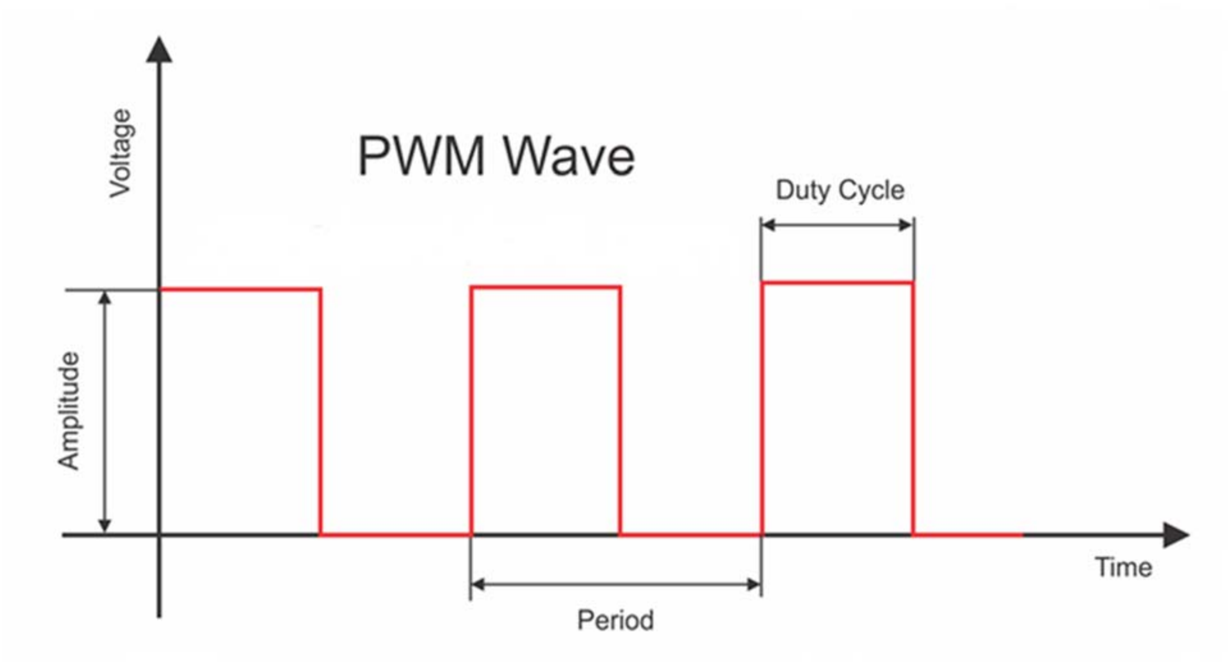


Figure 3.7 A PWM Waveform.

The duty cycle describes the time of signal in high (ON) status as a percentage of a total time of it takes to complete one cycle. One period pulse consist of time ON and time OFF, Fig (3.8) illustrate pulse timing. The frequency describes how fast the PWM complete a cycle (i.e. 500 Hz would be 500 cycles per second).

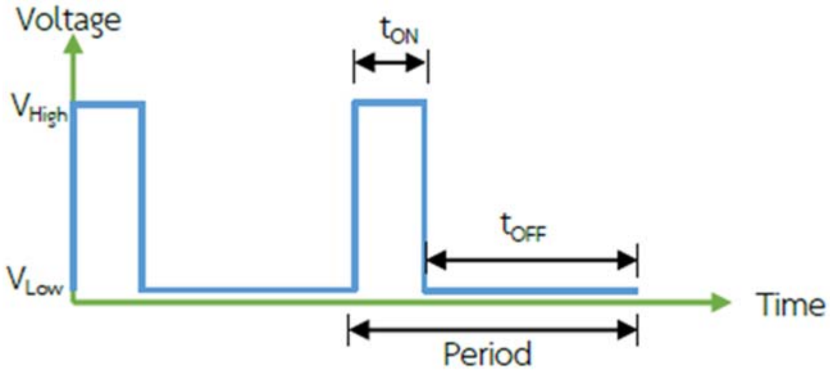


Figure 3.8 Period pulse timing.

The duty cycle can be calculated from the following formulas:

$$\boldsymbol{Duty\ Cycle} = \frac{\boldsymbol{t_{On}}}{\boldsymbol{T}} \tag{3.2}$$

Where T: is the total time ($t_{on} + t_{off}$).

Or

$$\boldsymbol{Duty\ Cycle} = \boldsymbol{PWM\ (in\ Sec) \times Frequency\ (in\ Hz) \times 100} \tag{3.3}$$

Fig (3.9) shown below are examples of a 0%, 25%, 50%, 75% and 100% duty cycle. While the frequency is the same for each.

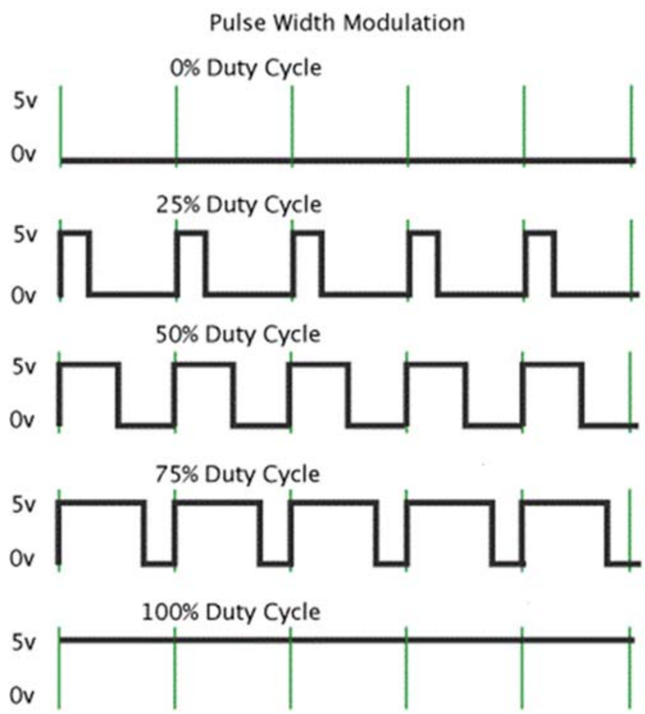


Figure 3.9 Examples of duty cycle.

3.4 Installing and connecting Flight Controller

Flight controller is the most important component in the UAV, it must be in the center of the frame (center of gravity) and with the same level of motors, in addition to its direction. Stability and controlling UAV is from flight controller responsibilities.

Controlling the motion of the Quadcopter (UAV) is by three main things, the Yaw (Rudder), Pitch (Elevator) and Roll (Ailerons), in addition to the Throttle, which is the distance from ground level.

3.4.1 Yaw (Rudder)

It is the rotating / deviation of the quadcopter (UAV) to right or to left, by rotating around the virtual axis Z. As shown in Fig (3.10) below.

Figure 3.10 Yaw axis.

3.4.2 Pitch (Elevator)

It is moved the UAV to the front or to the back, by rotating around the virtual axis Y. as shown in Fig (3.11).

Figure 3.11 Pitch axis.

3.4.3 Roll (Ailerons)

It moves the UAV to the sideward either to right or to left, by rotating around the virtual axis X. As shown in Fig (3.12) below. Many people are confusing between Yaw and Roll, Yaw is change the direction of the UAV fly but Roll is move the UAV to right or left.

Figure 3.12 Roll axis.

3.4.4 Center of Gravity (CG)

Is the effective point where all axes of flight (Roll, Pitch and Yaw) meet on it, also all weight is considered to be, Fig (3.13) shows the CG of UAV. CG point does not change in any aircraft but some times moves forward or backward along the longitudinal axis, depending on how the aircraft is loaded.

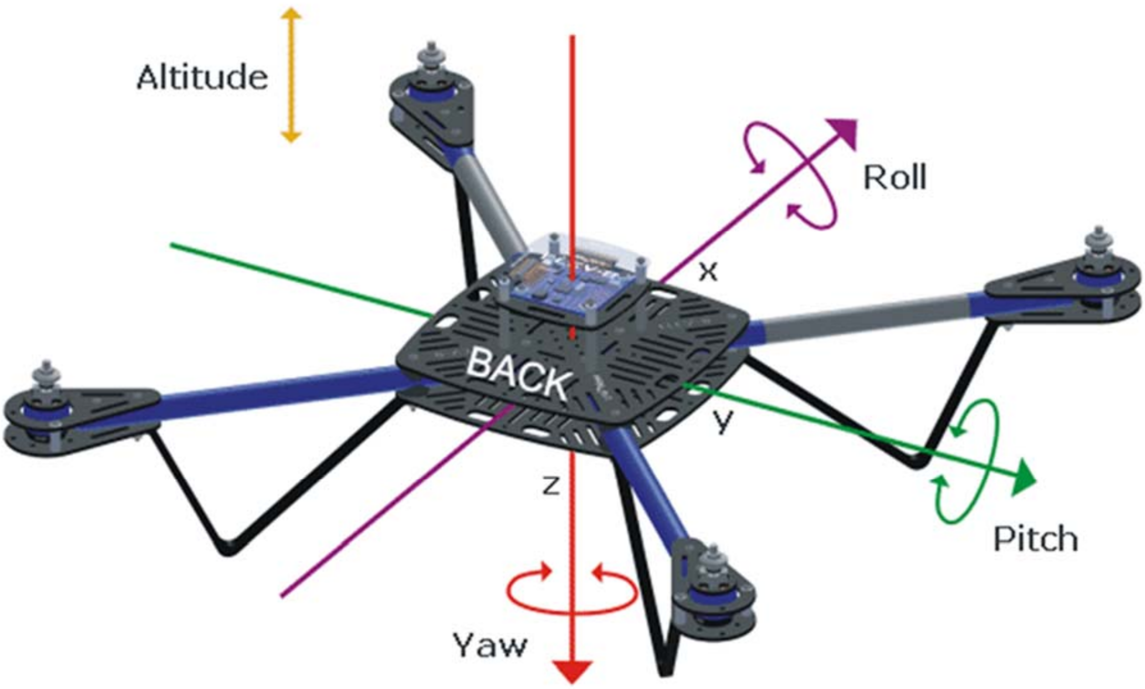


Figure 3.13 Center of Gravity of UAV.

3.4.5 PID

It is fundamentally a method utilized in programming and if made settings properly, can be incredibly effective and delicate. PID stands for Proportional Integral Derivative, three separate components joined together, though sometimes we do not require all three. For example, we could instead have just P control, PI control, PD control or PID control. Many flight controller software allow users to adjust PID values to get better performance of flight. PID is a function in the flight controller; it reads data from the sensors and inform the motors how fast they need to run. Finally, this is how the stability is obtained on UAV.

Proportional-Integral-Derivative (PID) is a closed loop control system that effort to get the actual result near or closer to the required result by regulating the input. The error is fed back to the beginning, and repeats the process, as shown in Fig (3.14) below.

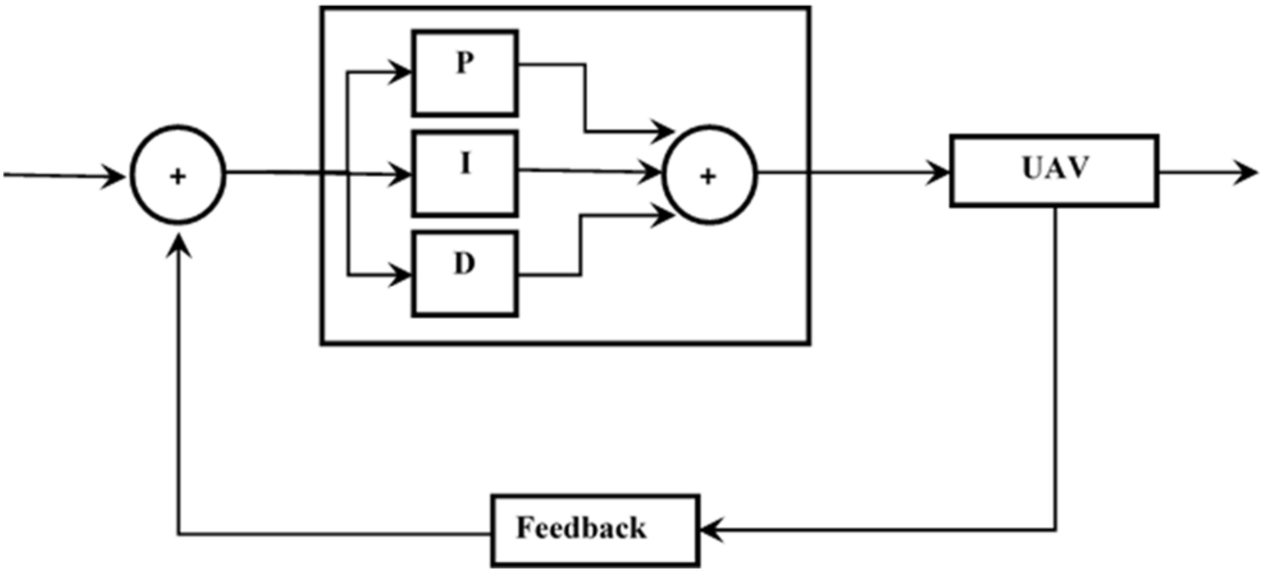


Figure 3.14 PID controller diagram.

UAV control is a mainly “difficult and interesting problem. With six degrees of freedom in which three are translational and three are rotational and only four autonomous inputs which are rotor speeds, UAVs are severely underactuated. To achieve six degrees of freedom rotational and translational motion are mixed. The produced dynamics are highly nonlinear, particularly after accounting for the complicated aerodynamic effects. Finally, unlike ground vehicles, UAVs have very little friction to restrain their motion, so they must furnish their own damping in order to stop moving and stay stable. Simultaneously, these factors create a very motivating control problem”.

3.4.5.1 UAV Dynamics

We will commence deriving UAV dynamics by introducing “the two frames in which will operate. The inertial frame is defined by the ground, with gravity pointing in the negative z-direction. The body frame is defined by the orientation of the UAV, with the rotor axes pointing in the positive Z-direction and the arms pointing in the X and Y directions”. Figure (3.15) shows the UAV body frame and inertial frame.

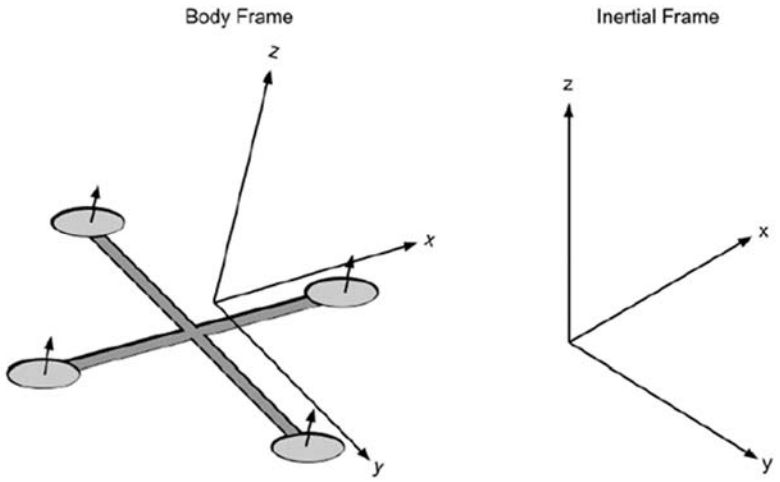


Figure 3.15 UAV Body frame and Inertial frame.

3.4.5.2 Kinematics

Let us formalize the kinematics in the body and inertial frames before delving into the physics of UAV motion. Position and velocity of UAV defined in the inertial frame as $position = (x, y, z)^T$and $elocity = (\dot{x}, \dot{y}, \dot{z})^T$. Similarly, the Yaw, Roll, and Pitch angles in the body frame was defined as $\theta = (\phi, \theta, \psi)^T$, with identical angular velocity as $\dot{\theta} = \left(\dot{\phi}, \dot{\theta}, \dot{\psi}\right)^T$. However, with the consideration that the angular velocity vector $\omega \neq \dot{\theta}$. The angular velocity is a vector pointed along the rotational axis, while $\dot{\theta}$ is the time derivative of Roll, Yaw and Pitch. "To convert these angular velocities into angular velocity vector", we can utilize the following relation:

$$\omega = \begin{bmatrix} 1 & 0 & -S_\theta \\ 0 & C_\phi & C_\theta S_\phi \\ 0 & -S_\phi & C_\theta C_\phi \end{bmatrix} \dot{\theta} \tag{3.4}$$

Where

ω is the angular velocity vector in the body frame.

The body frame can related with inertial frame by a rotation matrix R that goes from the body frame to the inertial frame. This matrix is determined by utilizing the ZYZ Euler angle conventions and successively "undoing" the Pitch, Roll and Yaw.

$$R = \begin{bmatrix} C_\phi C_\psi - C_\theta S_\phi S_\psi & -C_\psi S_\phi - C_\phi C_\theta S_\psi & S_\theta S_\psi \\ C_\theta S_\psi S_\phi + C_\phi S_\psi & C_\phi C_\theta C_\psi - S_\phi S_\psi & -C_\psi S_\theta \\ S_\phi S_\theta & C_\phi S_\theta & C_\theta \end{bmatrix} \tag{3.5}$$

For a given vector $\vec{v}$ in the body frame, the identical vector is given by $R\vec{v}$ in the inertial frame.

3.4.5.3 Physics

In order to fairly model the dynamics of the system, "an understanding of the physical properties that govern it is needed. We will start with detailing of the motors being utilized for our UAV, and then use energy considerations to drive the forces and thrusts that these motors produced in the entire UAV. All motors on the UAV are indistinguishable, so we can investigate a single one without loss of generality".

3.4.5.4 Motors

All UAV applications are used brushless motors. The torque produced for our electric motors is given by:

$$\tau = K_t(I - I_0) \tag{3.6}$$

Where

τ : the motor torque.

I : Input current.

I_0 : The current when there is no load in the motor.

K_t : The torque relativity constant.

The voltage across the motor is the sum of the some resistive loss and the back-EMF:

$$V = IR_m + K_v\omega \tag{3.7}$$

Where

V: the voltage drop across the motor.

R_m : the motor resistance.

ω : the angular velocity of the motor.

K_v : a proportionality constant (indicating back-EMF generated per RPM).

The following description can be used to calculate the power consumes of our motors. The power is:

$$P = IV = \frac{(\tau + K_t I_0)(K_t I_0 R_m + \tau R_m + K_t K_v \omega)}{K_t^2} \tag{3.8}$$

Assuming negligible motor resistance. Then, the power becomes proportionate to the angular velocity:

$$P \approx \frac{(\tau + K_t I_0) K_v \omega}{K_t} \tag{3.9}$$

For further simplification, assuming that $K_t I_0 \ll \tau$. Since I_0 is the current when there is no load, even that is too small. This approximation is good enough. Thus, the final simplified equation for power is:

$$P \approx \frac{K_v}{K_t} \tau \omega \tag{3.10}$$

3.4.5.5 Forces

By keeping of energy, we realize that “the energy of the motor consumes in a given duration is equal to the force generated on the propeller times the distance that the air it displaces moves ($P.dt = F.dx$)”. Equivalently, the *power* equals to the thrust times the air velocity $\left(P = \frac{dx}{dt}\right)$.

$$P = \mathrm{T}v_h \tag{3.11}$$

Assuming low vehicle speed, so v_h is the air velocity when flight. And also assuming that the free stream velocity, v_∞, is zero “the air in the surrounding environment is stationary proportional to the UAV”. Momentum theory gives us the equation for flight velocity as a function of thrust.

$$v_h = \sqrt{\frac{\mathrm{T}}{2\rho A}} \tag{3.12}$$

Where

ρ : the density of the surrounding air.

A : the area swept out by the rotor.

By using simplified equation of power:

$$P = \frac{K_v}{K_t}\tau\omega = \frac{K_v K_\tau}{K_t}\mathrm{T}\omega = \frac{\mathrm{T}^{\frac{3}{2}}}{\sqrt{2\rho A}} \tag{3.13}$$

In the general case, $\tau = \vec{r} \times \vec{F}$; in this case, the torque is relative to the thrust T by some constant ratio K_τ determined by the code configuration and parameters. For solving the thrust magnitude *T*, thrust is “proportional to the square of angular velocity of the motor” obtained:

$$T = \left(\frac{K_v K_\tau \sqrt{2\rho A}}{K_t}\ \omega\right)^2 = k\omega^2 \tag{3.14}$$

Where k is some fitly dimensioned constant. Summing over all motors, we find that the total thrust on the UAV (in the body frame) is given by:

$$T_B = \sum_{i=1}^{4} T_i = k \begin{bmatrix} 0 \\ 0 \\ \sum \omega_i^2 \end{bmatrix} \tag{3.15}$$

3.4.5.6 Torques

Since we have processed the powers on the UAV, we might like to figure the torques. "Every rotor contributes some torque about the body Z-axis. This torque is the torque required to keep propeller turning and providing thrust; it makes the immediate angular acceleration and defeats the frictional drag forces". From fluid dynamics the drag equation gives us the frictional force:

$$F_D = \frac{1}{2}\ \rho C_D A v^2 \tag{3.16}$$

Where

ρ: the surrounding fluid density.

A: the reference area (propeller cross-section, not area swept out by the propeller).

C_D: dimensionless constant.

This, while just exact in some at times, is good enough for our motivations. This infers the torque because of drag is given by:

$$\tau_D = \frac{1}{2} R_\rho C_D A v^2 = \frac{1}{2} R_\rho C_D A (\omega R)^2 = b\omega^2 \tag{3.17}$$

Where

ω: the angular velocity of the propeller.

R: the radius of the propeller.

b: some appropriately dimensioned constant.

Note that we've expected that all the force is applied at the tip of the propeller, which is certainly inaccurate; in any case, the main outcome that issues for our motivations is that the drag torque is corresponding to the square of the angular velocity. We would then be able to compose the entire torque about the Z-axis for the *i*th motor:

$$\tau_z = b\omega^2 + I_M\dot{\omega} \tag{3.18}$$

Where

I_M : the moment of inertia about the motor Z-axis.

$\dot{\omega}$: the angular acceleration of the propeller.

b : drag coefficient.

Note that in stable status flight (not landing or taking off) $\omega \dot{\approx} 0$, since most of the time the propellers will be preserving a constant (or nearly constant) thrust and won't be accelerating. Thus, simplifying the entire expression to:

$$\tau_z = (-1)^{i+1} b\omega_i^2 \tag{3.19}$$

Where the $(-1)^{i+1}$ negative for the *i*th propeller if they spinning CCW and positive if its spinning CW. The sum of all the torques from each propeller gives us the total torque about the Z-axis.

$$\tau_\psi = b(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \tag{3.20}$$

From standard mechanics, the Pitch and Roll torques are derived. The *i* =1 and *i* =3 motors arbitrarily chosen to be on the Roll axis, so

$$\tau_\phi = \sum r \times \mathrm{T} = L(k\omega_1^2 - k\omega_3^2) = Lk(\omega_1^2 - \omega_3^1) \tag{3.21}$$

Correspondingly, a similar expression gives the Pitch torque:

$$\tau_\theta = Lk(\omega_2^2 - \omega_4^2) \tag{3.22}$$

Where

L: the distance from the center of the UAV to any of the propellers. So, the torques in the body frame are:

$$\tau_B = \begin{bmatrix} Lk(\omega_1^2 - \omega_3^2) \\ Lk(\omega_2^2 - \omega_4^2) \\ b(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} \tag{3.23}$$

3.4.5.7 Equations of Motion

The acceleration of the UAV is "due to thrust, gravity, and linear friction in the inertial frame. The thrust vector in the inertial frame can be obtained by using the rotational matrix R to map the thrust vector from the body frame to the inertial frame". So, linear motion can summarized as:

$$m\ddot{x} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R\mathrm{T}_B + F_D \tag{3.24}$$

Where

$\vec{x}$: the position of the UAV.

g: the acceleration due to gravity.

F_D: the drag force.

T_B: the thrust vector in the body frame.

While it is advantageous to "have the linear equations of motion in the inertial frame, the rotational equations of motion are helpful to us in the body frame, so we can express rotations about the center of the UAV rather than about our inertial center". We determined the rotational equations of movement from Euler's equations for inflexible body dynamics. Expressed in vector form, Euler's

Equations are written as:

$$I\dot{\omega} + \omega \times (I\omega) = \tau \tag{3.25}$$

Where

ω : the angular velocity vector.

I : the inertia matrix.

τ : vector of external torques.

We can rewrite this as :

$$\dot{\omega} = \begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = I^{-1}\big(\tau - \omega \times (I\omega)\big) \tag{3.26}$$

We can show our UAV as two thin uniform rods crossed at the source with a point mass (motor) at the end of each. In view of this present, obviously, the symmetries result in a diagonal inertia matrix of the frame

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{3.27}$$

Thus, we obtain our last result for the body frame rotational equations of movement

$$\dot{\omega} = \begin{bmatrix} \tau_{\emptyset} I_{xx}^{-1} \\ \tau_{\theta} I_{yy}^{-1} \\ \tau_{\psi} I_{zz}^{-1} \end{bmatrix} - \begin{bmatrix} \frac{I_{yy}-I_{zz}}{I_{xx}} \omega_y \omega_z \\ \frac{I_{zz}-I_{xx}}{I_{yy}} \omega_x \omega_z \\ \frac{I_{xx}-I_{yy}}{I_{zz}} \omega_x \omega_y \end{bmatrix} \tag{3.28}$$

3.4.5.8 Control

The motivation behind deriving a numerical model of a UAV is to help with creating controllers for physical UAVs. The inputs to our framework comprise of the angular velocities of every rotor since everything we can control is the voltages over the motors. Note that we just utilized the square of the angular velocity, ω_i^2 and "never the angular velocity itself, ω_i. For notational effortlessness, let us present the inputs $\gamma_i = \omega_i^2$. Since we can set ω_i we can obviously set γ_i also. With this, we can compose our system as a first order differential equation in state space. Suppose x_1 be the position in space of the UAV, x_2 be the UAV linear velocity, x_3 be the Roll, Pitch, and Yaw angles, and x_4 be the angular velocity vector. (al of these are 3-vectors.) With these being our state, we can compose the state space equations for the development of our state".

$$\dot{x}_1 = x_2 \tag{3.29}$$

$$\dot{x}_2 = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \frac{1}{m} R\mathrm{T}_B + \frac{1}{m} F_D \tag{3.30}$$

$$\dot{x}_3 = \begin{bmatrix} 1 & 0 & -S_\theta \\ 0 & C_\phi & C_\theta S_\phi \\ 0 & -S_\phi & C_\theta C_\phi \end{bmatrix}^{-1} x_4 \tag{3.31}$$

$$\dot{x}_4 = \begin{bmatrix} \tau_\emptyset I_{xx}^{-1} \\ \tau_\theta I_{yy}^{-1} \\ \tau_\psi I_{zz}^{-1} \end{bmatrix} - \begin{bmatrix} \frac{I_{yy}-I_{zz}}{I_{xx}} \omega_y \omega_z \\ \frac{I_{zz}-I_{xx}}{I_{yy}} \omega_x \omega_z \\ \frac{I_{xx}-I_{yy}}{I_{zz}} \omega_x \omega_y \end{bmatrix} \tag{3.32}$$

3.4.5.9 PD Control

In order to control UAV, we will utilize a PD control, with “a component proportional to the error between our coveted path and the observed path, and a component proportional to the derivative of error. Because our UAV only have a gyro, so we only be able to utilize the angle derivatives $\dot{\phi}$, $\dot{\theta}$, and $\dot{\psi}$ in our controller; the measured values gives us the derivative of error, and their integral provides us the actual error. We might want to stabilize the UAV in a level position (horizontal), so our desired velocity and angles will all be zero. Torques are associated to angular velocities by $\tau = I\ddot{\theta}$, so we might want to set the torques proportional to the output of controller, with $\tau = Iu(t)$”. Thus,

$$\begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} -I_{xx}\left(K_d\dot{\phi} + K_p \int_0^{\mathrm{T}} \dot{\phi}\, dt\right) \\ -I_{yy}\left(K_d\dot{\theta} + K_p \int_0^{\mathrm{T}} \dot{\theta}\, dt\right) \\ -I_{zz}\left(K_d\dot{\psi} + K_p \int_0^{\mathrm{T}} \dot{\psi}\, dt\right) \end{bmatrix} \tag{3.33}$$

We have beforehand determined the relationship between torque and inputs, so we realize that

$$\tau_B = \begin{bmatrix} Lk(\omega_1^2 - \omega_3^2) \\ Lk(\omega_2^2 - \omega_4^2) \\ b(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} = \begin{bmatrix} -I_{xx}\left(K_d\dot{\phi} + K_p \int_0^{\mathrm{T}} \dot{\phi}\, dt\right) \\ -I_{yy}\left(K_d\dot{\theta} + K_p \int_0^{\mathrm{T}} \dot{\theta}\, dt\right) \\ -I_{zz}\left(K_d\dot{\psi} + K_p \int_0^{\mathrm{T}} \dot{\psi}\, dt\right) \end{bmatrix} \tag{3.34}$$

This gives us an arrangement of three equations with four unknowns. We can compel this by enforcing the constraint that our information sources (inputs) must keep the UAV aloft:

$$\mathrm{T} = mg \tag{3.35}$$

Note that this equation eliminates "the fact that the thrust won't be pointed straightforwardly up. This will restrict the applicability of our controller, however, should not cause real problems for small deviations from stability. If we had a way of deciding the current angle precisely, we can recompense. If our gyro is precise enough, we can integrate the values obtained from the gyro to obtain the angles θ and ϕ. In this case, we can compute the thrust necessary to keep the UAV aloft by projecting the thrust *mg* onto the inertial z-axis". We find that

$$\mathrm{T}_{proj} = \mathrm{mg} \cos\theta \cos\phi \tag{3.36}$$

Therefore, with an exact angle measurement, we can rather uphold the necessity that the thrust be equal to

$$\mathrm{T} = \frac{mg}{\cos\theta \cos\phi} \tag{3.37}$$

In which case the component of the thrust pointing along the positive z-axis will be equivalent to *mg*. We realize that the thrust is relative to a weighted sum of the inputs:

$$\mathrm{T} = \frac{mg}{\cos\theta \cos\phi} = k \sum \gamma_i \implies k \sum \gamma_i = \frac{mg}{k \cos\theta \cos\phi} \tag{3.38}$$

With this additional imperative, we have a set of four linear equations with four unknowns γ_i. By solving for each γ_i, the following input values obtained:

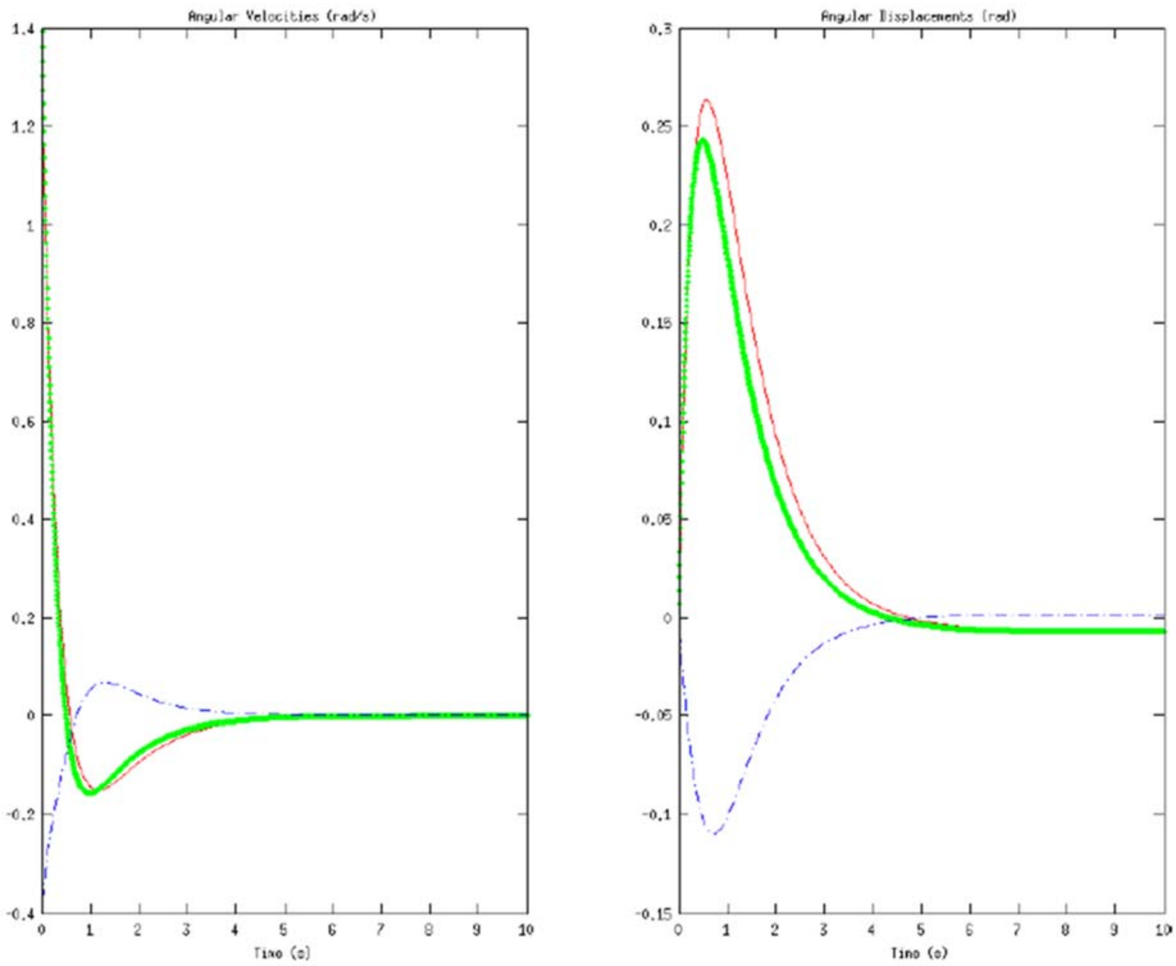
$$\gamma_1 = \frac{mg}{4k\cos\theta\cos\emptyset} - \frac{2be_{\emptyset}I_{xx}+e_{\psi}I_{zz}kL}{4bkL} \tag{3.39}$$

$$\gamma_2 = \frac{mg}{4k\cos\theta\cos\emptyset} + \frac{e_{\psi}I_{zz}}{4b} - \frac{e_{\theta}I_{yy}}{2kL} \tag{3.40}$$

$$\gamma_3 = \frac{mg}{4k\cos\theta\cos\emptyset} - \frac{-2be_{\phi}I_{xx}+e_{\psi}I_{zz}kL}{4bkL} \tag{3.41}$$

$$\gamma_4 = \frac{mg}{4k\cos\theta\cos\emptyset} + \frac{e_{\psi}I_{zz}}{4b} + \frac{e_{\theta}I_{yy}}{2kL} \tag{3.42}$$

This is an entire detailing for PD controller. The controller drives the angular velocities and angles to zero, as shown in Fig (3.16) bellow.



Angular velocities **Angular displacements**

Figure 3.16 Angular velocities and angular displacements. ϕ, θ, ψ are coded as red, green, and blue.

Nevertheless, note that the angles are not entirely driven to zero. “The average steady state error is approximately 0.3 o . This is a prevalent problem with utilizing PD controllers for mechanical systems and can be partially alleviated with a PID controller”.

In addition, take note of that since we are just “controlling angular velocities, our positions and linear velocities don't go to zero. Nonetheless, the z position will stay consistent, in light of the fact that we have obliged the aggregate vertical thrust to be such that it keeps the UAV perfectly aloft, without ascending or descending”.

3.4.5.10 PID Control

PD controllers are suitable in their straightforwardness and simplicity of implementation, but they are “frequently deficient for controlling mechanical systems. Particularly in the presence of noise and disturbances, PD controllers will often lead to steady state error. A PID control is a PD control with another term included, which is corresponding to the integral of the process variable. Including an integral term makes any remaining steady-state error to develop and enact a change, so a PID controller ought to have the ability to track our path (and stabilize the UAV) with an essentially smaller steady-state error”. The equations stay “identical to the ones displayed in the PD case, but with an extra term in the error”:

$$e_{\phi} = k_d\dot{\phi} + k_p \int_0^{\mathrm{T}} \dot{\phi} dt + \ k_i \int_0^{\mathrm{T}} \int_0^{\mathrm{T}} \dot{\phi} \ dt \ dt \qquad (3.43)$$

$$e_{\theta} = k_d\dot{\theta} + k_p \int_0^{\mathrm{T}} \dot{\theta} dt + \ k_i \int_0^{\mathrm{T}} \int_0^{\mathrm{T}} \dot{\theta} \ dt \ dt \qquad (3.44)$$

$$e_{\psi} = k_d\dot{\psi} + k_p \int_0^{\mathrm{T}} \dot{\psi} dt + \ k_i \int_0^{\mathrm{T}} \int_0^{\mathrm{T}} \dot{\psi} \ dt \ dt \qquad (3.45)$$

However, PID controls come with their own inadequacies. One trouble that ordinarily happens with “a PID control is known as integral windup. In some cases, integral wind-up can cause stretched oscillations instead of settling. In other cases, wind-up may indeed cause the system to become unstable, instead of taking longer to reach a steady state”, as shown in Fig (3.17) below.

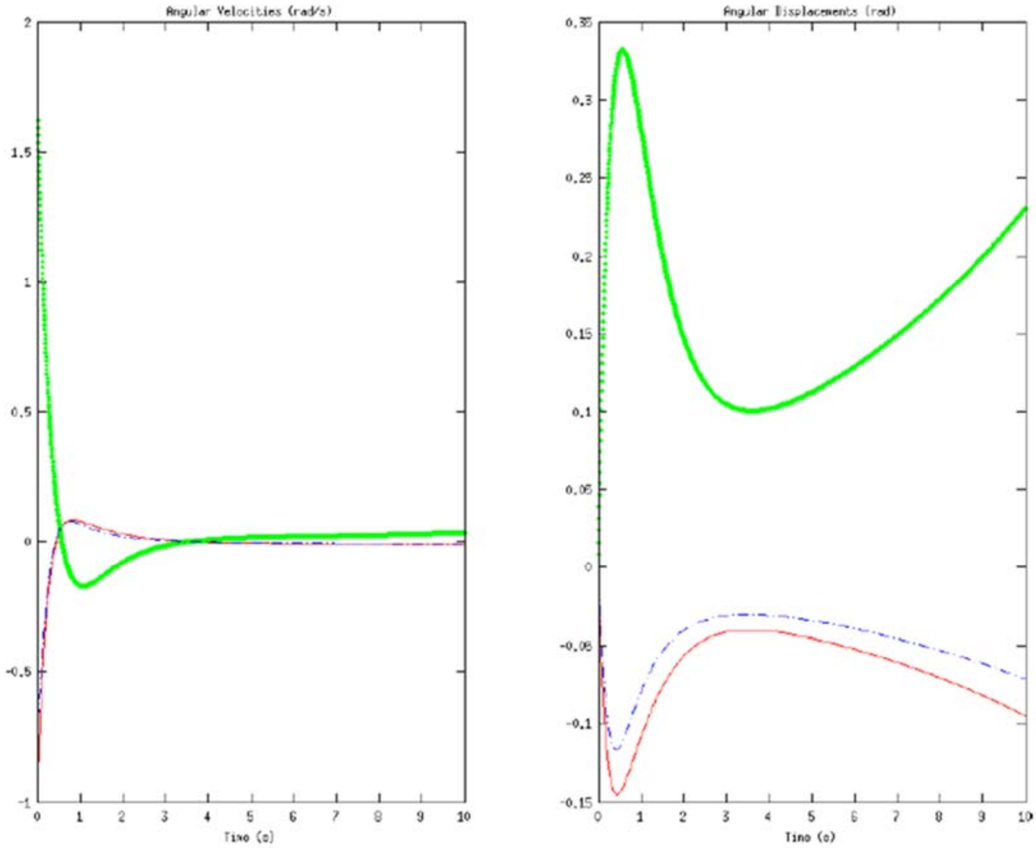


Figure 3.17 PID controller.

If there is a large trouble in the process variable, this large trouble is integrated over time, becoming a still bigger control signal (due to the integral term). However, even once the system stabilizes, the integral is still big, therefore making the controller overshoot its objective. It may then start “a series of dieing down oscillations, become unstable, or basically take an incredibly long time to reach a steady state. In order to avert this, we disable the integral function until we reach something near to the steady state. When we are in a controllable region near the desired steady state, we turn on the integral function, which pushes the system towards a minimal steady-state error”, as shown in Fig (3.18).

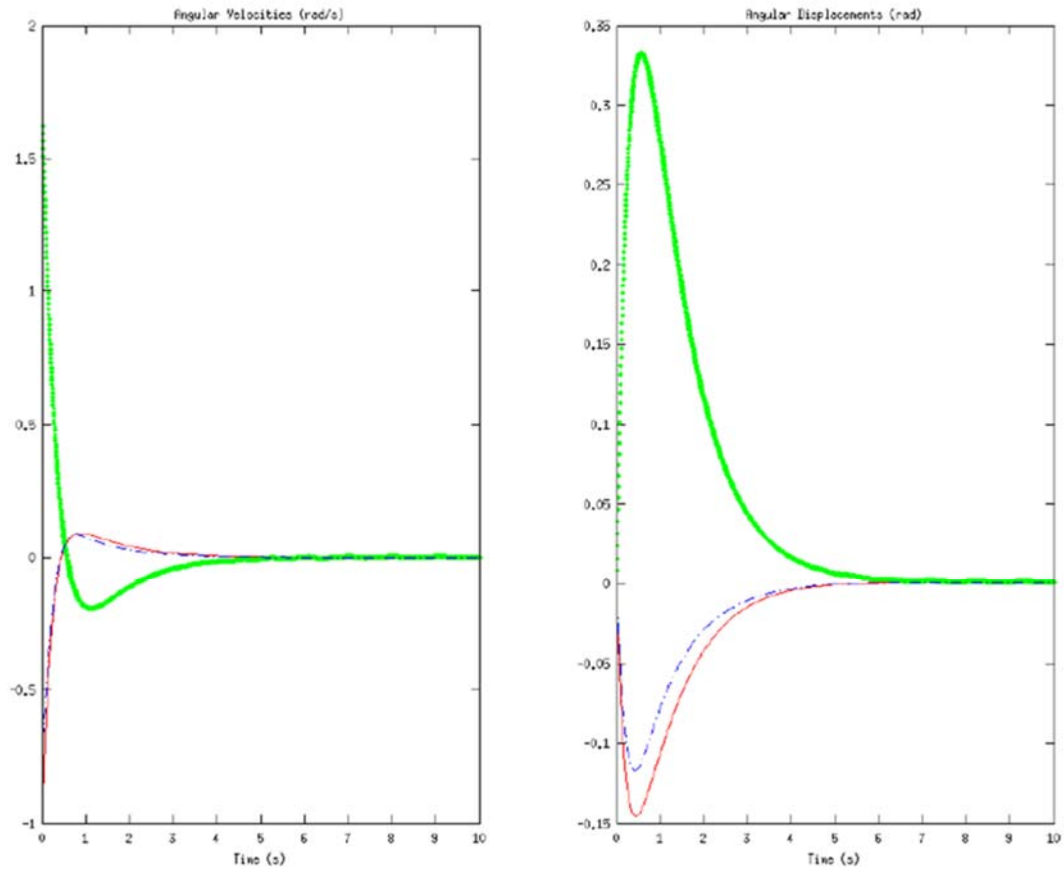


Figure 3.18 With a properly implemented PID, we achieve an error of approximately 0.06 after 10 seconds.

3.5 Remote Control (RC)

There are many types of remote controls in markets, in our thesis we chose Fly Sky – T6 6 channels 2.4 GHz AFHDS computerized digital proportional R/C airplane and helicopter system.

AFHDS: Stands for "Automatic Frequency Hopping Digital System". This highly developed radio transmission system will assure a long range, jamming free and long battery life experience. [53]

RF specifications:

Our radio system (remote control) works in the frequency range 2.4000 to 2.4835 GHz. 500 KHz channel bandwidth, this band has been divided into "160 independent channels". Each radio system uses "16 different channels and 160 different types of hopping algorithm". And uses less than 20 dBm (100 mV) from RF power and GFSK modulation type, with -1058 dBm of RX sensitivity. This radio system uses a high gain and high quality multidirectional antenna. It covers the whole frequency band, also assure a jamming free long radio transmissions. Fig (3.19) shows fly-sky remote control (transmitter and receiver).

Figure 3.19 Fly – Sky remote control.

In this thesis, we connect the receiver of remote control to the UAV by using servo wires; we connect it with the flight controller as shown in Fig (3.20) below.

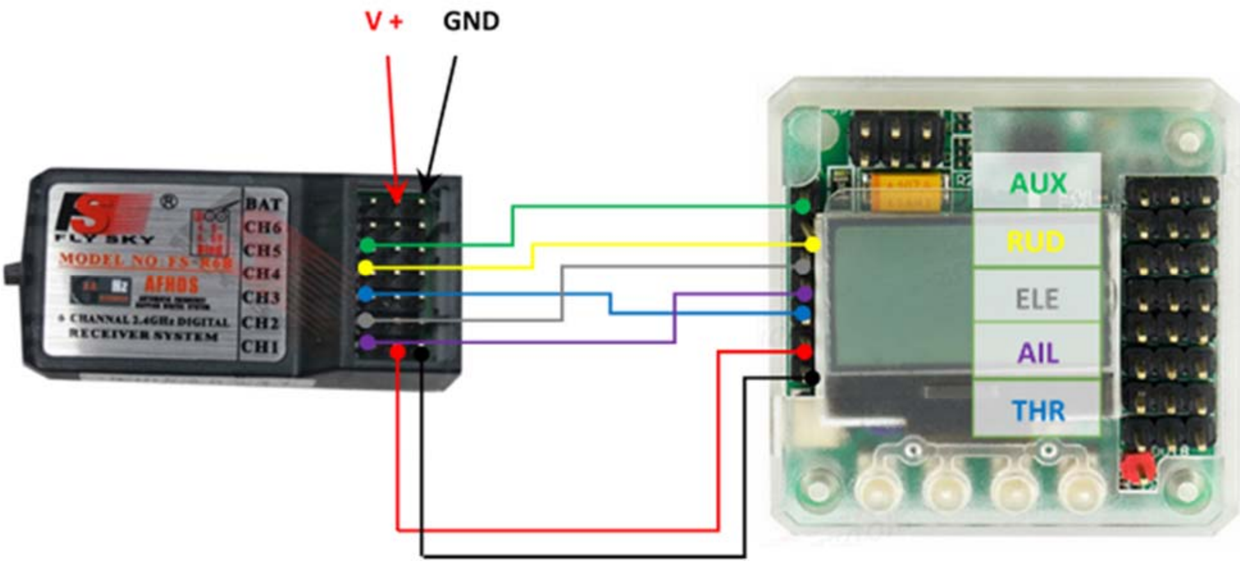


Figure 3.20 diagram shows the connection between RC receiver and flight controller.

3.6 Reading signals from RC receiver

Our contribution is designing a system that uses internet of things to make the control for Unmanned Aerial Vehicle (drone) from an infant distance by simulating the signals coming from RF receiver and remote control. So, we need to read the signals from RF receiver, actually we read it in two types:

First type: we connect the receiver to the oscilloscope and take signals and all information from that signals, but these signals not accurate enough as shown in Fig(3.21).

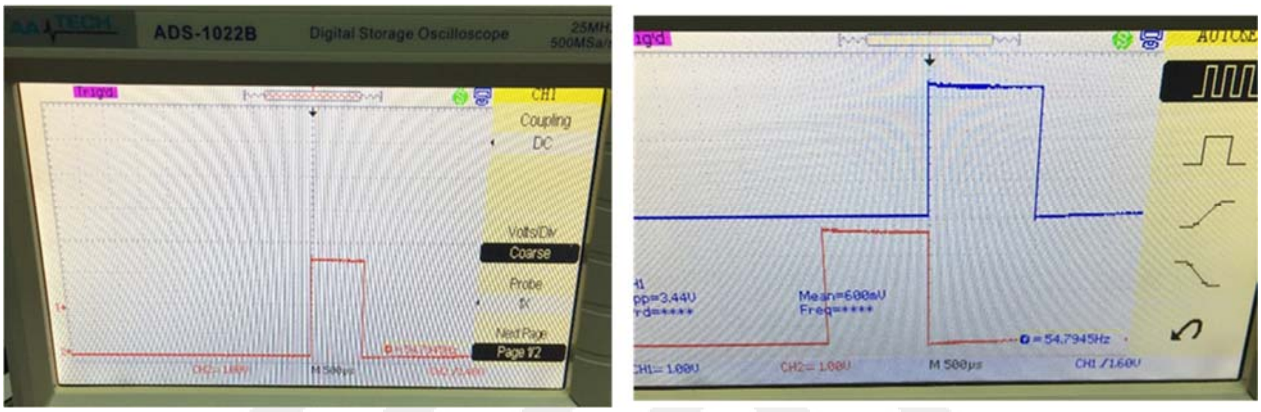


Figure 3.21 Oscilloscope signals from RF receiver.

Second type: we connect the RF receiver to the PWM pins in the Arduino, and we wrote a code for Arduino to read the signals from PWM pins. In Arduino to read the total pulse we need to read the high and the low duration of the pulse, and we get:

In High duration pulse: THR changes from 988 microsecond (down) to 1965 microsecond (up) (left bar in the remote), AIL (Roll) changes from (1250) microsecond (go to left side) to (1735) microsecond (go to right side) (right bar in remote) and the midpoint is (1495) microsecond. ELE (Pitch) changes from (1250) microsecond (go to forward) to (1735) microsecond (go to backward) (right bar in remote) and the midpoint is (1496) microsecond. RUD (Yaw) changes from (1250) microsecond (rotate to right) to (1739) microsecond (rotate to left) (left bar in remote) and the midpoint (1495) microsecond. AUX changes by the upper left tuning (VRB) from (983) microsecond (tune to the positive side) to (1907) microsecond (tune to the negative side) and the Midpoint (1373) microsecond (0 point), Fig (3.22) shows the diagram of high period pulses and Fig (3.23) shows the diagram of low period pulses.

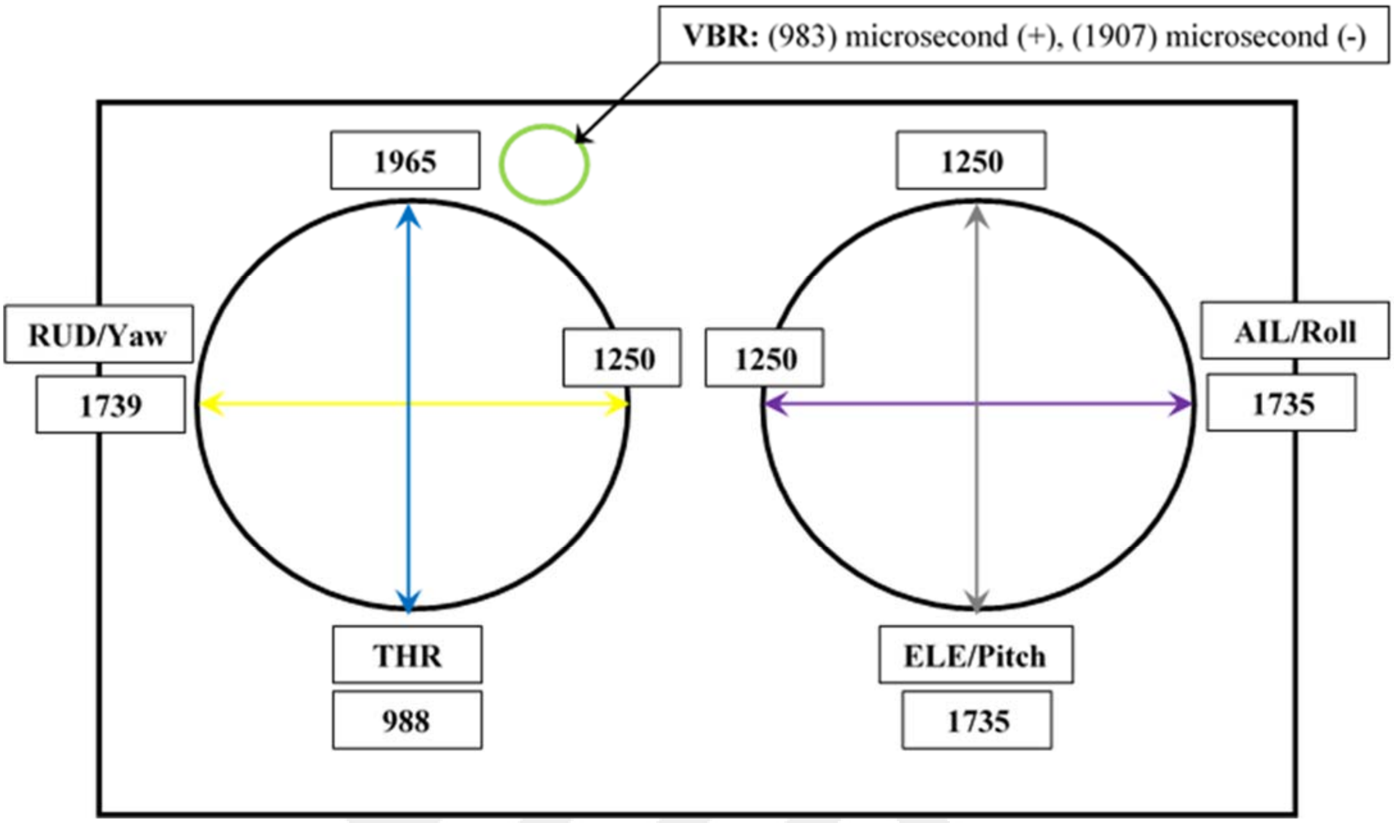


Figure 3.22 High period pulses in microsecond.

In Low duration pulse: THR changes from 17147 microsecond (down) to 16182 microsecond (up) (left bar in the remote), AIL (Roll) changes from (16894) microsecond (go to left side) to (16409) microsecond (go to right side) (right bar in remote) and the midpoint is (16693) microsecond. ELE (Pitch) changes from (16894) microsecond (go to forward) to (16412) microsecond (go to backward) (right bar in remote) and the midpoint is (16693) microsecond. RUD (Yaw) changes from (16893) microsecond (rotate to right) to (16406) microsecond (rotate to left) (left bar in remote) and the midpoint (16695) microsecond. AUX changes by the upper left tuning (VRB) from (17153) microsecond (tune to the positive side) to (16236) microsecond (tune to the negative side) and the Midpoint (16688) microsecond (0 point), Fig (3.22) shows the diagram of high period pulses and Fig (3.23) shows the diagram of low period pulses.

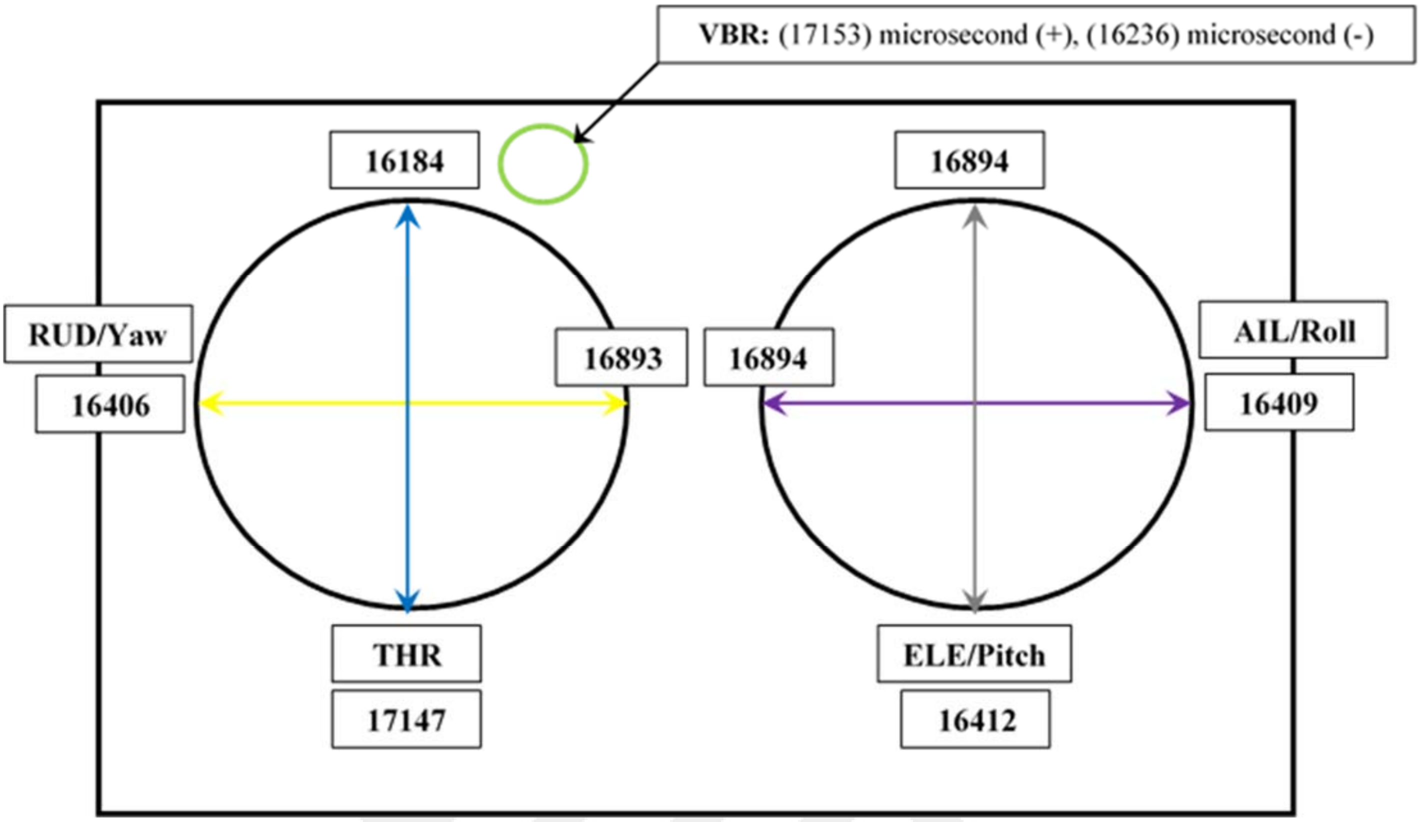


Figure 3.23 Low period pulses in microsecond.

We summarized the high and low durations of the pulses in the following table (3.1).

Table 3.1 High and low durations of pulses read from RF remote control.

		High duration of the pulse in µsec	**Low duration of the pulse in µsec**
THR	Min	988	17147
	Max	1965	16184
Yaw/RUD	Right	1250	16893
	Left	1739	16406
	Mid	1495	16695
Roll/AIL	Right	1735	16409
	Left	1250	16894
	Mid	1495	16693
Pitch/ELE	Forward	1250	16894
	Backward	1735	16412
	Mid	1496	16693

3.7 Generating PWM by using Arduino Uno and Raspberry Pi3

In order to achieve our contribution (IoT UAV) we replaced the RF remote control, by utilizing Arduino Uno and raspberry pi3 because RF remote control covers a limited distance and our contribution is to make our UAV infinite distance controlling by using the Internet.

We used Raspberry Pi3 for connection and designed a GUI (Graphical User Interface) to control the UAV. GUI is “a program interface that takes advantage of the computer's graphics capabilities to make the program easier to use. Well-designed graphical user interfaces can free the user from learning complex command languages. On the other hand, many users find that they work more effectively with a command-driven interface, especially if they already know the command language”.

Our GUI was written in python programming language, and control the UAV through it. Figure (3.24) shows our GUI.

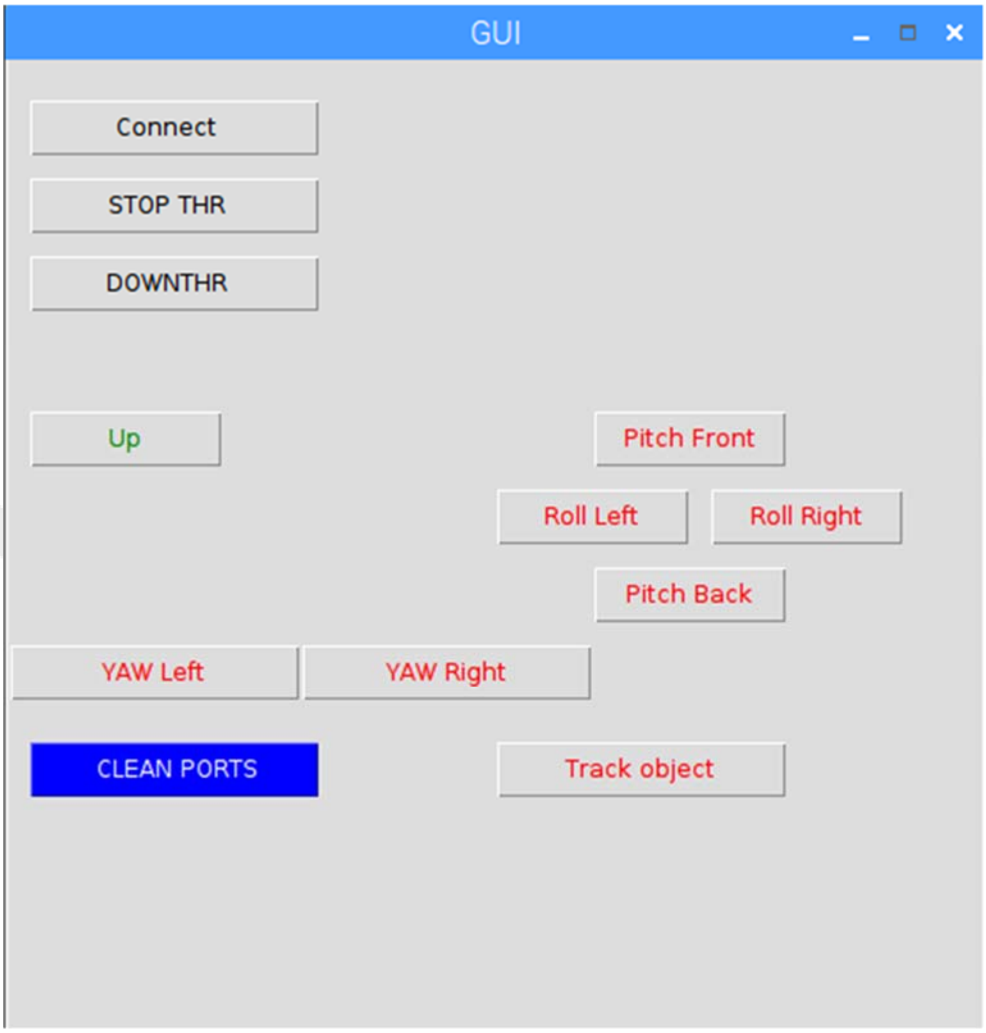


Figure 3.24 Designed GUI.

Because of our flight controller works in 5v and the Raspberry pi works in 3.3v therefore we need another electronic card that support 5v, this card is a microcontroller card Arduino Uno which gives us 5v with PWM ports.

Our procedure is to control the UAV from Raspberry Pi3 using GUI in python, python give orders to the microcontroller card (Arduino Uno) which the last generate the right pulses needed as we read it from the table (3.1) above, and send it to the flight controller.

3.8 Connection Establishment and object detection

In order to establishment a connection between UAV and the earth station (desktop or laptop) we used TeamViewer application. The TeamViewer have many advantages over using real IP. One of them it is free to use while the real IP providers requires fees per period. Also the TeamViewer is more easer in installation and use rather than real IP.

The earth station controlled the Raspberry Pi by running its operating system directly through TeamViewer and run the GUI of our UAV with all its instructions.

For more features, the UAV able to tracking objects by using Raspberry Pi camera and based on the BGR color moment calculations.

3.9 Color detection in OpenCV and Python

Color detection is the process of finding certain color and extracts it from its surroundings in order to detect objects having the same colors. The process in OpenCV is constructed using (cv2.inRange) function which is built in function and this function detects colors in a rage between lower and upper values that are provided for certain color range like blue for example in our work, in the following Figure (3.25) our detected object based on color.

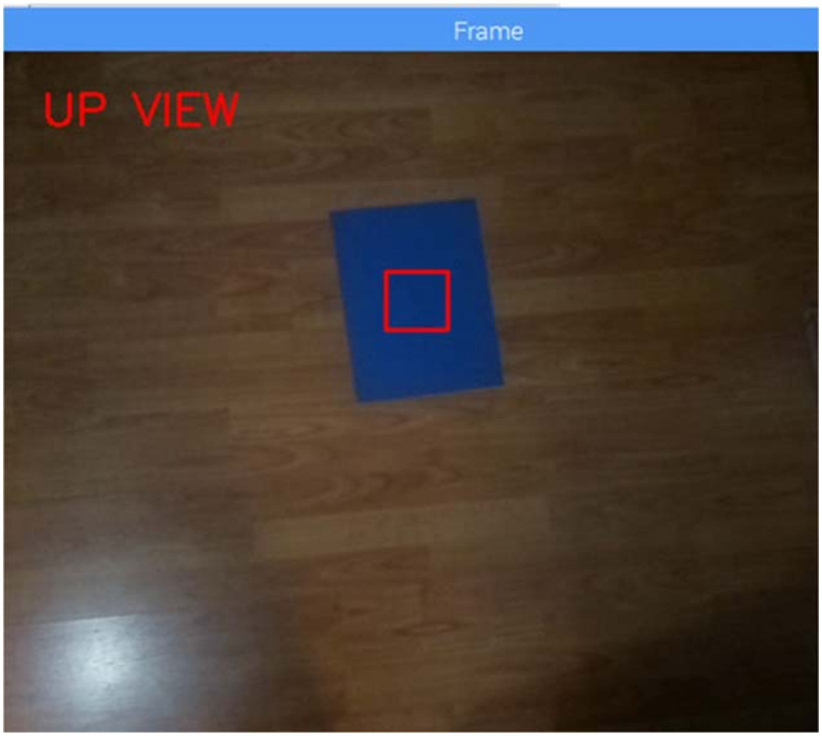


Figure 3.25 Detected object based on color.

In color detection it is recommended to convert the captured image from 'BGR' color range to HSV color range because it is more easer to isolate colors from each other's.

In numerical representation in our code:

`lower = np.array([76,31,4])` represents the lower value for the color that we intended to detect.

`upper = np.array([210,90,70])` represents the upper value for the color that we intended to detect.

`thresh = cv2.inRange(blur, lower, upper)` this part will detect the color in the image captured by camera that is limited between the lower and upper values of that is provided by us.

Then we will find the contours in the image or frame under test by the following line in our code:

```
“contours,hierarchy= cv2.findContours(thresh,cv2.RETR_LIST,
cv2.CHAIN_APPROX_SIMPLE)”
```

After that we will find the contour with the maximum area and we will consider it as our goal as shown in the following lines in code; finding contour with maximum area and store it as best_cnt:

```
“   max_area = 0
    best_cnt = 1
    for cnt in contours:
        area = cv2.contourArea(cnt)
        if area > max_area:
           max_area = area
           best_cnt = cnt “
```

After that we will find the centroid of the best contour and we will draw a box surrounding it as shown in previous Figure (3.25).

```
“M = cv2.moments(best_cnt)”
“cx,cy = int(M['m10']/M['m00']), int(M['m01']/M['m00'])”
```

CHAPTER 4

SIMULATION

4.1 Quad copter simulation in Simulink

The simulation process should be made before the construction of Quad copter, for that reason the simulation was made in Simulink to study the effect of weights, torques, forces, resulted from the frame weight, motors rotation speed, electronic card's weight, also the selecting of PID controller parameters and fine tuning these parameters.

The simulation process is made using Sims cape multi body, before that CAD model should be imported from one of the mechanical 3d construction design programs, like solid works or others.

Figure (4.1) shown below represents the frame 3D design for our quad copter and it will be used in our study.

Figure 4.1 drone final mechanical model imported from auto cad or 3D max

The quad copter consists of many collected items which will represent the full Drone body like shown in previous figure.

The importance of importing the drone as separated parts forming the full body is to control some of these parts and leaving others like rotating the motor shaft and keeping the motor body stable same as in reality.

The process of controlling the Quad copter or making flight controller for it consists of many stages starting with inputs which are coming from the sensors which represents the values of (Roll, Pitch, Yaw, Altitude) as shown in figure (4.2) below:

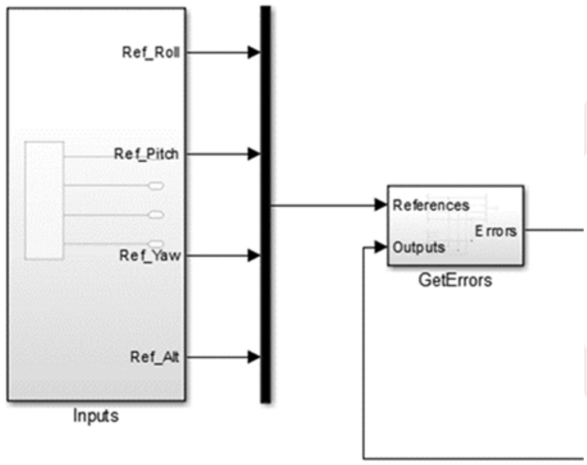


Figure 4.2 the sensor outputs

The main benefit of sensor values is to give feed back to the controller to check the level of the drone axis, for example if we have the value of Pitch is (-22) it means that the drone is turning to right side and if we will leave it turning for unwanted period of time that means it will fall down, so the controller is responsible of returning the drone to the horizontal plane to prevent it from falling down.

The same procedure will be applied for (Yaw, Roll and Altitude) to keep the drone stable in its flight.

The most important issue is that the input coming from the IMU which will represent the values of (Yaw, Roll, Altitude), these values should be accurate to provide the flight controller with suitable values to insure the best flight, the inputs are simulated by using the input block shown in figure (4.2) above.

The next step in the flight controller procedure is to calculate the error for (Yaw, Roll, Pitch and Altitude), error means the difference between the desired values and the measured ones, this step is made by the (GetErrors) block drown in figure(4.2) above.

After that the error values will be injected to flight controller in order to be evaluated and to be corrected as shown in figure (4.3) below.

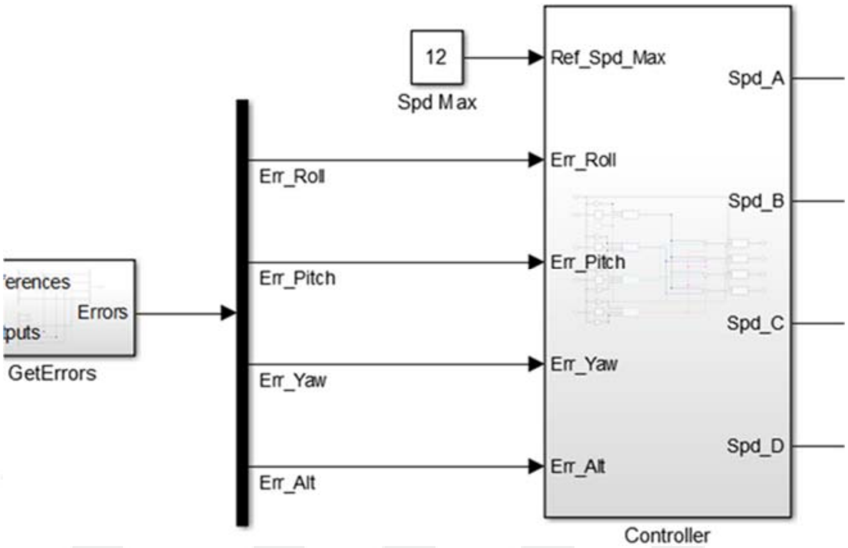


Figure 4.3 the injected values from error calculating block to the flight controller.

Inside the flight controller, a correcting process will be made to reduce the error for (Yaw, Roll, Pitch and Altitude) smoothly using PID algorithm for each value separately. The main purpose for applying simulation for drone is to estimate the correct values for (P gain, I gain, D gain) in a way that prevents the drone from doing extreme actions or to go unstable during flight.

To go further in our explanation we will show the inner circuits inside controller to show the hall procedure made by our system, as shown in Figure (4.4) below.

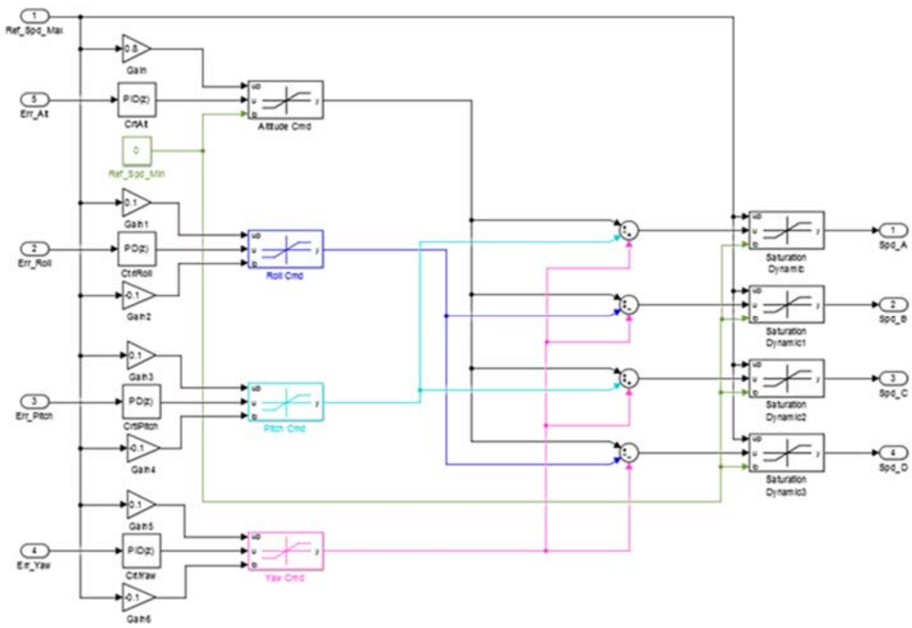


Figure 4.4 the PID controller for (Yaw, Roll, Pitch and Altitude).

To explain the operation of the circuit we will start with the first section as shown in figure (4.5) below:

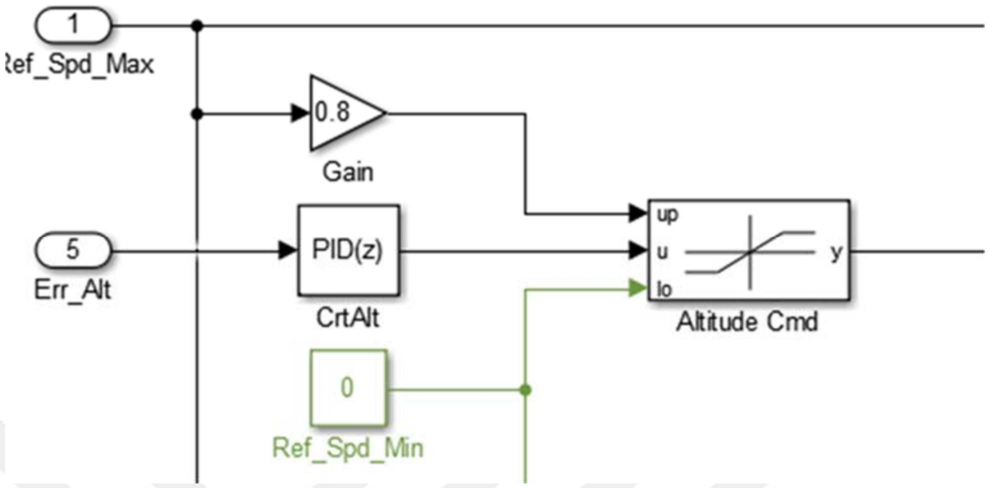


Figure 4.5 first section inside controller.

The PID controller for (z) means the PID for altitude; we can see that PID controller takes its input from the error signal (Err_Alt) which represents the difference between the actual altitude and the desired one, inside PID controller a PID control will be applied by selecting the values related to it as follows:

Proportional (P): 4.5

Integral (I):10

Derivative (D): 1

The output of PID controller is shown in figure (4.6) below and as we can see that both of its, stability and transient time is suitable (the most important goals of PID control), we will discuss the tuning effect on behavior later on.

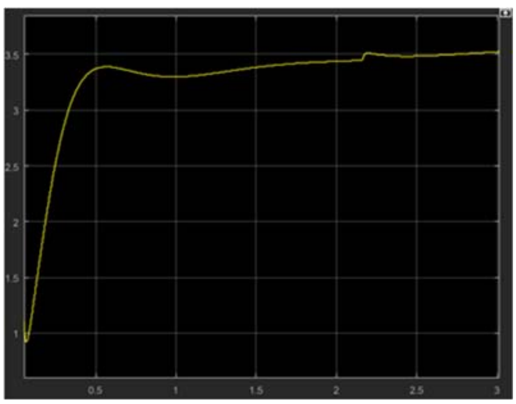


Figure 4.6 PID (z) output behavior.

The purpose of adding Altitude Cmd is to limit the boundaries of PID (z) control between the 0.8*(Ref_Spd_Max) as an upper limit and (Ref_Spd_Min) which represent the maximum reference speed and minimum reference speed, which means preventing the drone from reaching undesired speed. After that the (Saturation Dynamic) block also prevents the drone from exceeding the dynamic boundaries.

Finally the corrected values will be send out as a decision to Quad motors through terminals(Spd_A, Spd_B, Spd_C, Spd_D) to (Quadcotor 3D Model) block to reposition the drone to the correct level and frame.

We will discuss all the steps in the process to estimate the accurate values to our work.

After starting the simulation we can draw the trajectory of drone by drawing the XY graph as shown in figure (4.7) below:

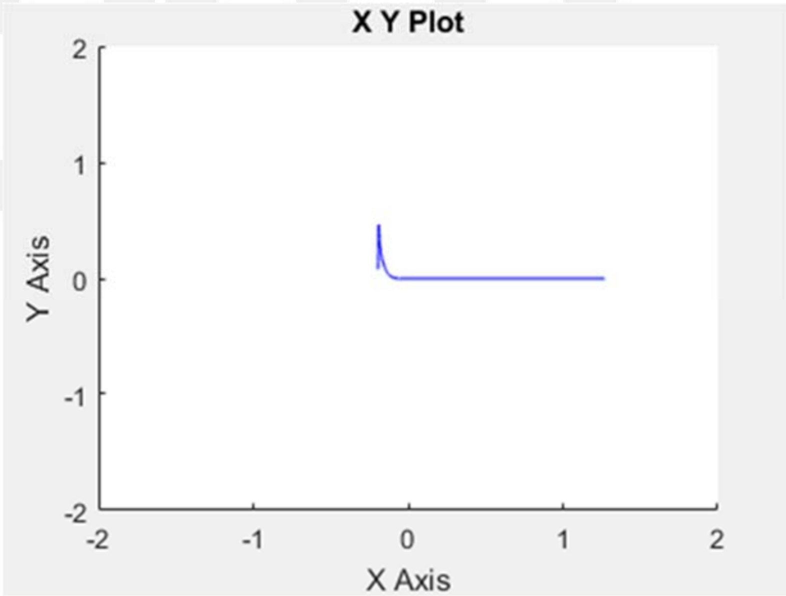


Figure 4.7 drone trajectory.

In the real time operation of quadcopter the commands operating it comes from remote control unit but in simulation in Simulink these commands can be applied from signal builder block in Simulink as shown figure(4.8) below:

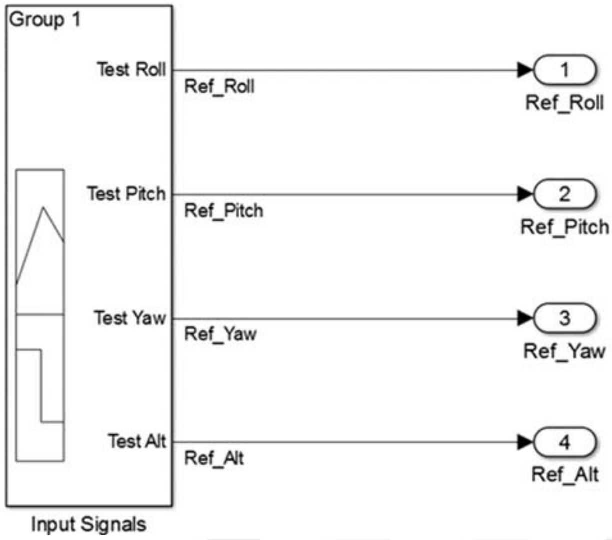


Figure 4.8 Signal builder as an input signals.

After starting simulation a 3D drone will be shown on mechanics explorer window as shown in figure (4.9) below:

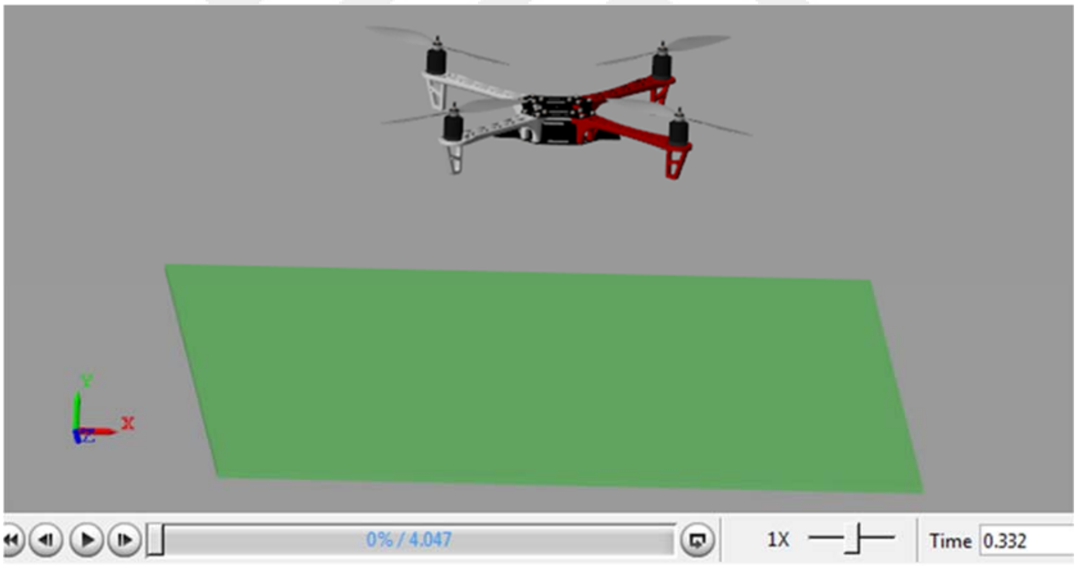


Figure 4.9 Mechanics explorer's window during simulation.

4.2 Drone mechanical parts

The mechanical parts forming the drone in Simulink is consist of the following drawing figure (4.10):

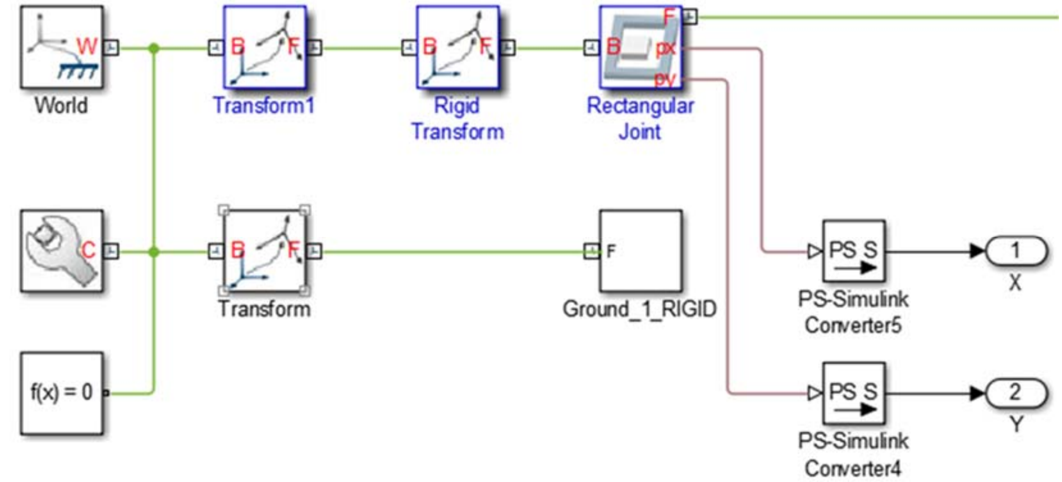


Figure 4.10 (a) Drone parts (first step).

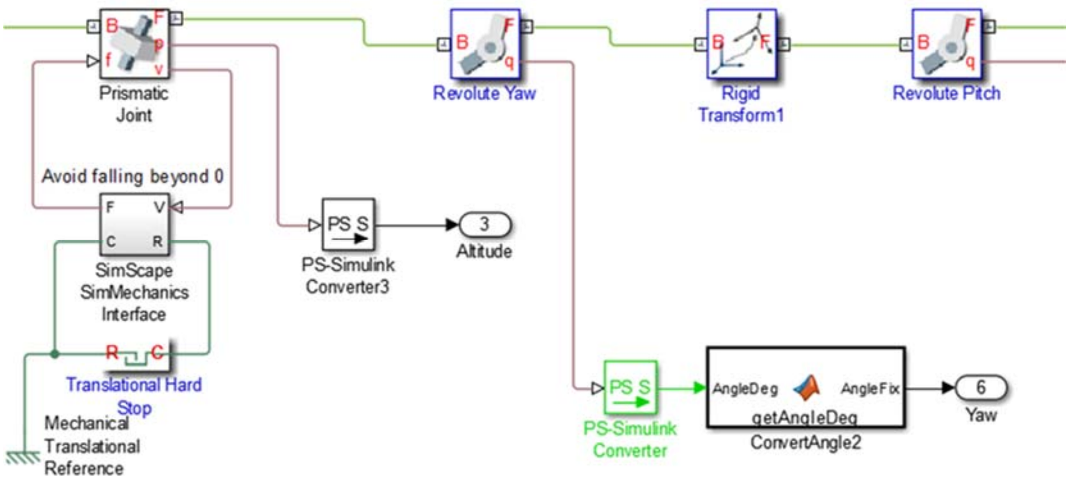


Figure 4.10 (b) Drone parts (second step).

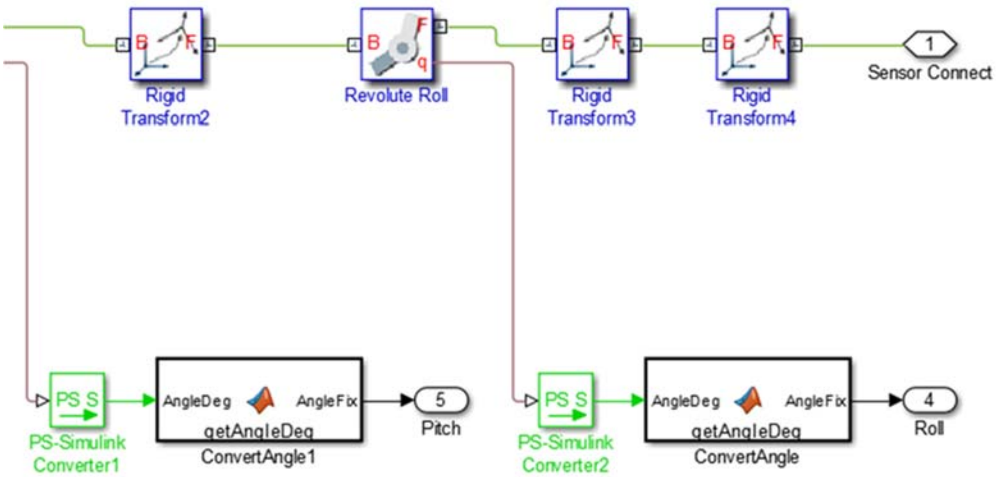
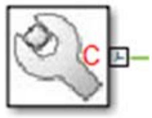


Figure 4.10 (c) Drone parts (third step).

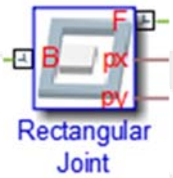
The first part consist of the following blocks:

World block: the world block represents the world frame which is motionless and represents ground.

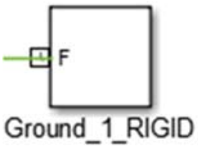


Mechanism configuration: it applies a gravity effect to the body and it can be added in any direction x, y or z direction.

The transform block represents connection between two frames.

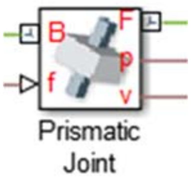


Represents a rectangular joint between two frames. This joint has two translational degrees of freedom represented by two prismatic primitives along a set of two mutually orthogonal axes.



The rigid body represents frame on ground (as a base).

This block converts the input Physical Signal to a unit less Simulink output signal.



This block represents a joint with one translational degree of freedom.

This block represents a joint with one rotational degree of freedom.

This block is a function block and if the angle value is more than 360 degree it will make it in range of 360.

4.3 PID different tuning values and its effect on flight

1. For the following values of PID (z) controller we will get the output signal as in figure (4.11).

Proportional (P): 4.5

Integral (I): 10

Derivative (D): 1

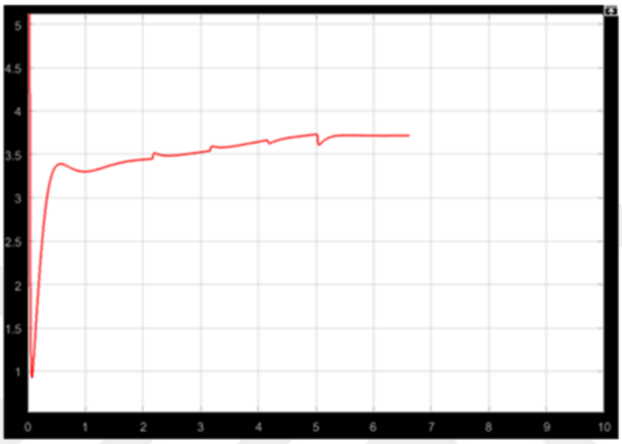


Figure 4.11 Output from PID (z) controller.

2. For the following values of PID (z) controller we will get the output signal as in figure (4.12).

Proportional (P): 3.5

Integral (I): 10

Derivative (D): 1

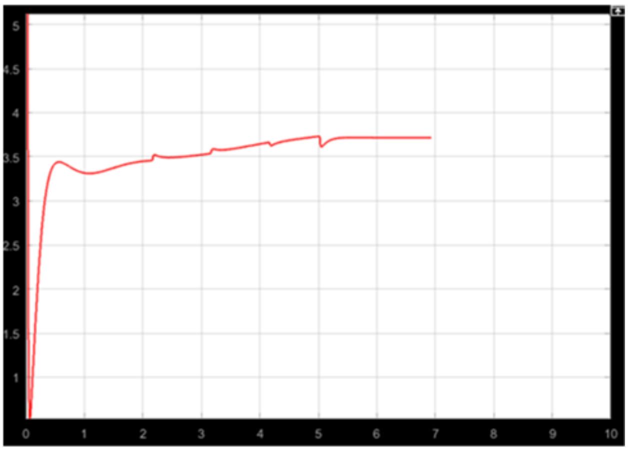


Figure 4.12 Output from PID (z) controller.

3. For the following values of PID (z) controller we will get the output signal as in figure (4.13).

Proportional (P): 2.5

Integral (I): 10

Derivative (D): 1

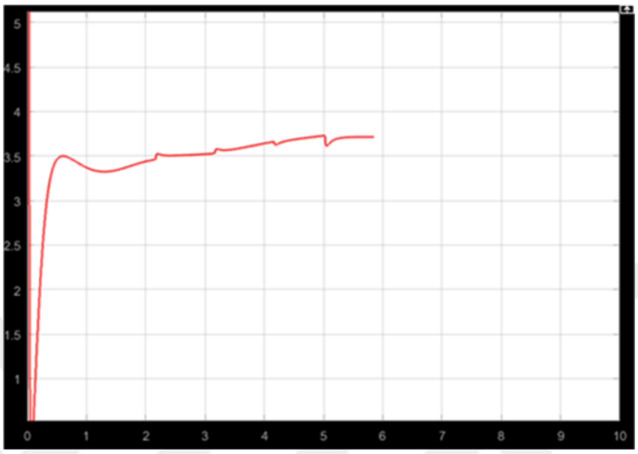


Figure 4.13 Output from PID (z) controller.

4. For the following values of PID (z) controller we will get the output signal as in figure (4.14).

Proportional (P): 0.5

Integral (I): 10

Derivative (D): 1

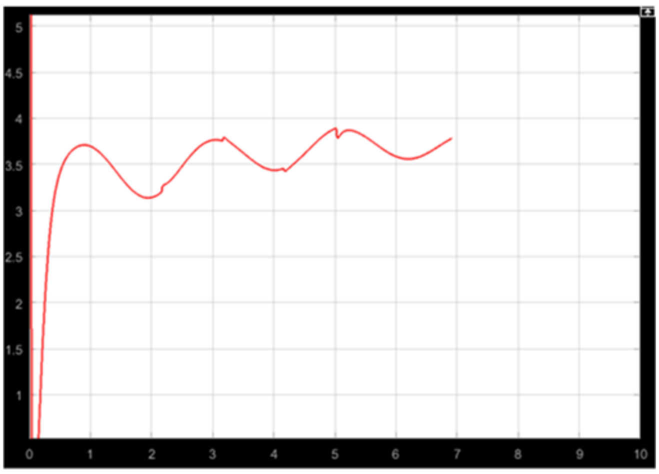


Figure 4.14 Output from PID (z) controller.

5. For the following values of PID (z) controller we will get the output signal as in figure (4.15).

Proportional (P): 4.5

Integral (I): 8

Derivative (D): 1

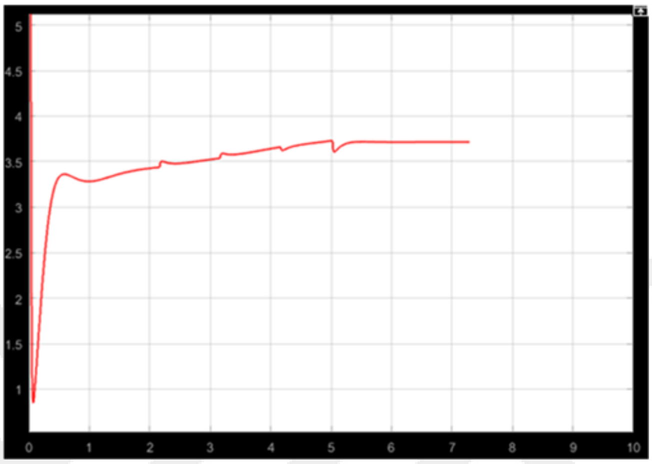


Figure 4.15 Output from PID (z) controller.

6. For the following values of PID (z) controller we will get the output signal as in figure (4.16).

Proportional (P): 4.5

Integral (I): 6

Derivative (D): 1

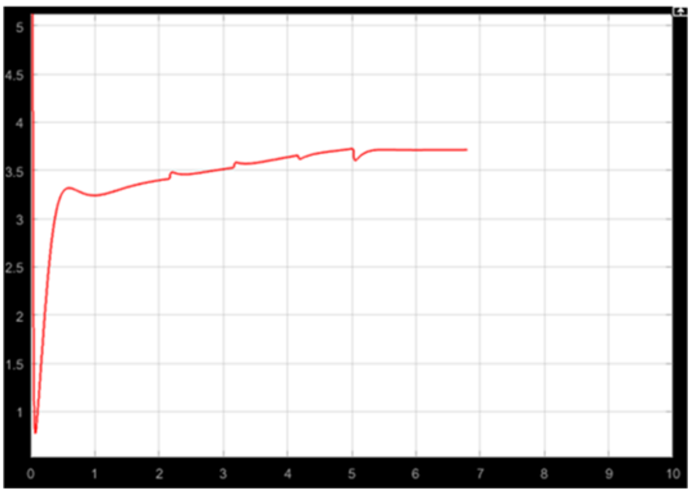


Figure 4.16 Output from PID (z) controller.

7. For the following values of PID (z) controller we will get the output signal as in figure (4.17).

Proportional (P): 4.5

Integral (I): 2

Derivative (D): 1

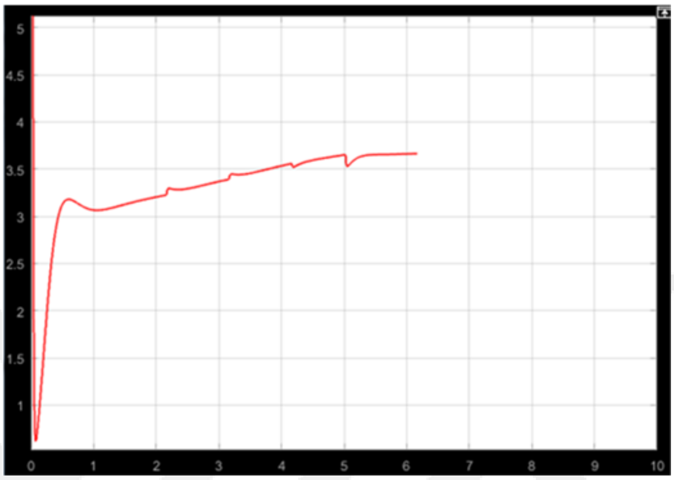


Figure 4.17 Output from PID (z) controller.

8. For the following values of PID (z) controller we will get the output signal as in figure (4.18).

Proportional (P): 4.5

Integral (I): 2

Derivative (D): 0.5

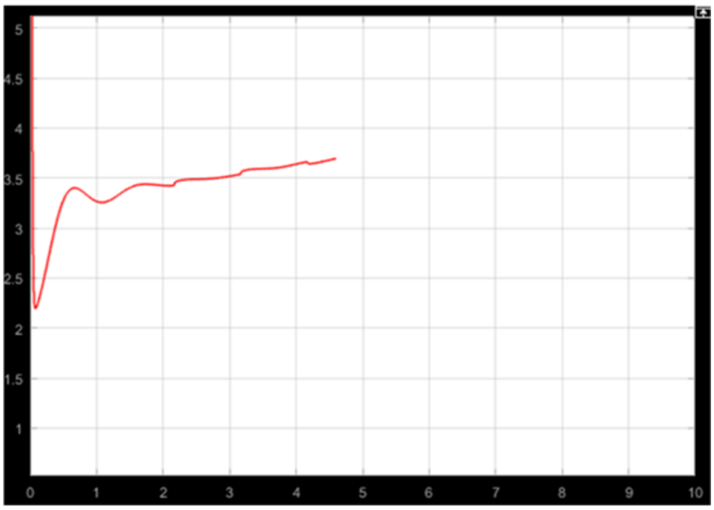


Figure 4.18 Output from PID (z) controller.

9. For the following values of PID (z) controller we will get the output signal as in figure (4.19).

Proportional (P): 4.5

Integral (I): 2

Derivative (D): 0.1

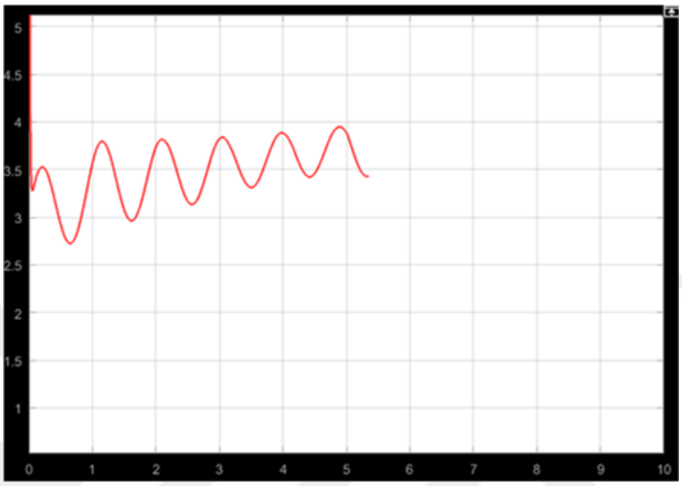


Figure 4.19 Output from PID (z) controller.

4.4 Conclusion of Simulation

As we can see from the previous drawings and values that There are infinite probability for changing PID gain values so the tuning of the right values need to be tried to get a small transient period and less oscillation but there will be limitations because the drone will response slowly to the process of control.

CHAPTER 5

RESULT AND DISCUSSION

The new generation of Wireless Sensor Networs, that is known as the the Internet of Things (IoT) enables the direct connection of physical objects to the Internet using microcontrollers.

The Internet of Things (IoT) is “a technology that allows objects to be connected to hte internet, enabling them with communication capabilities (with other objects and with people)”.

In our study we developed an UAV system that is controlled using internet connection which gives capability for infinite distance control, the UAV in our system can detect certain objects using their colors.

after construction of the system and after making test for flight control and object detection we discovered that the process of flight control based on internet connection demands a high bit rate for communication to ensure the stability of the UAV system and online video transmission and object detection , also such a system needs an autonomous flight algorithm in case of the connection lost or jamming procedure from enemy in case of military applications.

Also the process of using raspberry pi 3 for such an action is somehow not reasonable because of the limited capabilities for such an embedded systems and the heavy task needed in this operation.

In the futer work it is recommended to use Fuzzy Logic in after recognition step in order to control the movment of object toward target.

A detail study should be made to select the member ship functions for fuzzy logic, the selection also should cover the limit values for each member ship function and that would be made by experuments.

The movment should control the values of Yaw, Pitch and Roll to reach the target in a smooth motion and not to lose its trace during the tracking process.

The Defuzzification Methods should be made according to selected method from fuzzy known methods such as (Center of are, Center of sums, Center of maximum,...etc.).

REFERENCES

[1] Brien, O., Michael, H., (2016). “The Internet of Things”, Journal of Internet Law, 19 (12) : 1-20.

[2] Bughin, J., Chui, M., and Manyika, J., (2015). “An executive's guide to the Internet of Things”, Mckinsey Quarterly, 4 : 92-101.

[3] Velez, F., J., Nadziejko, A., Anders, L., Oliveira, S., Rodrigues, T., Costa, V., Duarte, M., Silva, F., and Gomes, J., (2015). “Wireless Sensor and Networking Technologies for Swarms of Aquatic Surface Drones”, IEEE 82nd Vehicular Technology Conference. 10(1109):739-793.

[4] Zahugi, E., Mohamed, H., Shanta, Mohamed, M., and Prasad, T.V., (2012). “Design of Multi-Robot System for Cleaning up Marine Oil Spill”, International Journal of Advanced Information Technology, 2(4):192-198.

[5] Burange, W., and Misalkar, H., D., (2015). “Review of Internet of Things in development of smart cities with data management & privacy”, International Conference on Advances in Computer Engineering and Applications, 10(1109):164-193.

[6] Spalazzi, L., Taccari, G., and Bernardini, A., (2014). “An Internet of Things ontology for earthquake emergency evaluation and response”, International Conference on Collaboration Technologies and Systems (CTS), 20(14):686-699.

[7] Jiuyan, C., Bin, L., Q. and Fang, Y., (2013). “A precision agriculture management system based on Internet of Things and WebGIS”, (2013). 21st International Conference on Geoinformatics, 66(2):61-73.

[8] D. Nield, “Boeing’s latest patent reveals a drone that can transform into a submarine”, ScienceAlert. http://www.sciencealert.com/boeings-latest-patent-reveals-a-drone-that-can-transform-into-a-submarine , 2016.

[9] Economic Research Office, Information and Communications in Japan, Ministry of Internal Affairs and Communications (MIC), Japan. http://www.soumu.go.jp/johotsusintokei/whitepaper/eng/WP2011/index.html, 2011.

[10] Qiantori, A., Sutiono, A., B., Hariyanto, H., Suwa, H., and Ohta, T., (2010). “An emergency medical communications system by low altitude platform at the early stages of a natural disaster in Indonesia” , J. Med. Syst., 36(1):41–52.

[11] Rogers, J., (2015). “How drones are helping the Nepal earthquake relief effort,” FoxNews.Com. http://www.foxnews.com/tech/2015/04/30/how-drones-are-helpingnepal-earthquake-relief-effort.html , April 2015.

[12] R. Debra and C. McCullough, Unmanned Aircraft Systems (UAS) Guidebook in Development. http://cops.usdoj.gov/html/dispatch/UAS_Guidebook_in_Development.asp , 7 Mar 2016

[13] S. W. Loke, “The Internet of Flying-Things: Opportunities and challenges with airborne fog computing and mobile cloud in the clouds,” Dept. Comput. Sci. Inf. Technol., La Trobe Univ., Melbourne VIC, Australia, Jul. 2015, https://arxiv.org/abs/1507.04492

[14] Brien, J., O., Baran, D., G., and Luu, B., B., (2006). “Ad hoc networking for unmanned ground vehicles: Design and evaluation at command, control, communications, intelligence, surveillance and reconnaissance on-the-move,” ARL, Adelphi, MD, USA, Army Res, 39:91-101.

[15] IAFC (International Association of Fire Chiefs). (Jan. 2014). Use of Unmanned Aerial Vehicles in Public Safety Emergency Response, Adopted by IAFC Board of Directories. http://www.iafc.org/IAFC-position-Use-of-Unmanned-Aerial-Vehicles-In-Emergency-Response

[16] Bolkcom, C. (2004, June). Homeland security: Unmanned aerial vehicles and border surveillance. Library of Congress Washington DC Congressional Research Service.

[17] M. T. McCaul, (2012). “Using unmanned aerial systems within the homeland: Security game changer?” House Committee Homeland Security, Washington, DC, USA, Tech. Rep, 11(2):10-17.

[18] W. Staff, “Fighting forest fires before they get big—With drones,” WIRED, Jun. 2015. http://www.wired.com/2015/06/fighting-forest-fires-get-big-drones

[19] J. K. Hart and K. Martinez, “Environmental sensor networks: A revolution in the earth system science?” Earth Sci. Rev., vol. 78, nos. 3–4, pp. 177–191, Oct. 2006.

[20] P. Dockrill, “Facebook is preparing its Internet-beaming drone for maiden launch,” ScienceAlert. Accessed on Mar. 7, 2016. http://www.sciencealert.com/facebook-is-preparing-its-internet-beaming-drone-for-maiden-launch.

[21] D. Sahota, “Internet.org building drones to connect remote communities,” Telecoms.com, Mar. 2014. http://telecoms.com/239252/internet-org-building-drones-to-connect-remotecommunities/ .

[22] T. Skinner, “Nokia and Du launch network testing drones in Dubai,” Telecoms.com, Jul. 2015. http://telecoms.com/430141/du-and-nokia-launch-network-testing-drones-in-dubai/ .

[23] G. Kimchi et al., “Unmanned aerial vehicle delivery system,” U.S. Patent 20 150 120 094, Apr. 2015.

[24] D. Lee, “Google plans drone delivery service for 2017,” BBC News. http://www.bbc.com/news/technology-34704868 .

[25] Amazon, “Amazon prime air” , http://www.amazon.com/b?node=8037720011

[26] Vermesan, O., & Friess, P. (Eds.). (2013). Internet of things: converging technologies for smart environments and integrated ecosystems. River Publishers.

[27] C. Snow, “Why drones are the future of the Internet of Things,” Dec. 2014, http://droneanalyst.com/2014/12/01/drones-are-the-future-of-iot/

[28] ITU Telecommunication Standardization Sector, ITU-T Recommendation database, http://handle.itu.int/11.1002/1000/11559-en?locatt=format:pdf&auth , 13 April 2015.

[29] Postscapes, Internet of Things Definition- Postscapes, Postscapes, http://postscapes.com/internet-of-things-definition , 13 April 2015.

[30] Cisco Systems, Internet of Things (IoT) - Cisco Systems, http://www.cisco.com/web/solutions/trends/iot/overview.html , 14 April 2015.

[31] IBM, IBM Social Media, https://www.youtube.com/watch?v=sfEbMV295Kk , 15 March 2010.

[32] J. Barrett, The Internet of Things Dr. John Barrett at TEDxCIT YouTube, TEDx Talks, https://www.youtube.com/watch?v=QaTIt1C5R-M , 5 October 2012.

[33] Upton, Eben (29 February 2016). "Raspberry Pi 3 on sale now at $35 - Raspberry Pi", Raspberry Pi.

[34] "How Much Power Does Raspberry Pi3B Use? How Fast Is It Compared To Pi2B?", http://www.raspi.tv.raspi.tv , 6 July 2016.

[35] "Raspberry Pi 2 on sale now at $35". Raspberry Pi Foundation, http://www.raspi.tv.raspi.tv , 5 August2015.

[36] "Raspberry Pi2 – Power and Performance Measurement". http://www.raspi.tv.raspi.tv , 6 July 2016.

[37] "I have a Raspberry Pi Beta Board AMA". http://www.reddit.com , 6 May 2012.

[38] Raspberry Pi boot configuration text file.

[39] "Nokia 701 has a similar Broadcom GPU". Raspberrypi.org. 2 February 2012.

[40] "introducing new firmware for the 512 MB Pi". 16 September 2014.

[41] "Raspberry Pi 3 specs". raspberrypi.com. Magpi. 1 October 2016.

[42] "Raspberry Pi 2 specs". raspberrypi.com. 1 October 2016.

[43] "Verified USB Peripherals and SDHC Cards;". Elinux.org. Retrieved 6 May 2012.

[44] "GPIO: Models A+, B+, Raspberry Pi 2 B and Raspberry Pi 3 B - Raspberry Pi Documentation". www.raspberrypi.org, Retrieved 2017-02-28.

[45] http://www.raspberrypi.org/documentation/hardware/raspberrypi/schematics/Raspberry-Pi-B-Plus-V1.2-Schematics.pdf

[46] https://www.raspberrypi.org/documentation/hardware/raspberrypi/schematics/RPI-3B-V1_2-SCHEMATIC-REDUCED.pdf

[47] http://www.raspberrypi.org/documentation/hardware/raspberrypi/schematics/Raspberry-Pi-Rev-2.1-Model-AB-Schematics.pdf

[48] "Using Atmel Studio for Arduino development". http://www.megunolink.com , 28 Jan 2016.

[49] "Using AVR Studio for Arduino development". Engblaze.com. 18 Jan 2016.

[50] "Ch Arduino". http://www.arduino.com , 7 OCT 2016.

[51] "Programming Arduino Getting Started with Sketches". McGraw-Hill. 8 Nov 2011.

[52] KK2.1.5 EVO and Mini Instruction Manual (Flight controller manual).

[53] Fly-Sky FS-CT6A Remote Control manual.

CURRICULUM VITAE

PERSONAL INFORMATION

Name Surname	: Fahad Yassir AL BAZZAZ
Date of birth and place	: Iraq – Mosul in 08/05/1983
Foreign Languages	: English and Turkish
E-mail	: fahad.albazzazapple@gmail.com

EDUCATION

Degree	**Department**	**University**	**Date of Graduation**
Master	Computer Eng.	Yildiz Technical University	2017
Undergraduate	Computer Eng.	Technical College of Mosul	2005
High School	AL Sharqya		2001

WORK EXPERIENCE

Year	**Corporation/Institute**	**Enrollment**
2014	The Ministry of Electricity	Control and Operating Department
2010	The Ministry of Electricity	
2005	The Ministry of Electricity	Computer Department

PUBLISHMENTS

Conference Papers

1. ICAS 2017, Istanbul, Turkey. Drone long distance remote control based on internet connection and embedded microcomputer