Sudan University of Science and Technology

Collage of Engineering

School of Electrical Engineering

Design and Implementation of Multifunction Quadcopter Intelligent System Using Machine Learning Techniques

تصميم و تنفيذ نظام طائرة رباعية بدون طيار ذكي متعدد الوظائف بإستخدام تقنيات تعلم الآلة

A Project Submitted in Partial Fulfillment for the Requirements of the Degree of B.Sc. (Honor) in Electrical Engineering

Prepared By:

1. Ali Mustafa Alnour Alhaj
2. Elebeid Khalid Elsayed Bakhit
3. Ahmed Mohammed Alsayed Ibrahim
4. Hashim Ahmed Mohammed Eltoum

Supervised By:

Ust. Omer Mohammed Salama Adam

November 2020

الآيـــــــــة

قال تعالي :

بسم الله الرحمن الرحيم

﴿ وَقُلِ اعْمَلُوا فَسَيَرَى اللَّهُ عَمَلَكُمْ وَرَسُولُهُ وَالْمُؤْمِنُونَ وَسَتُرَدُّونَ إِلَى
عَالِمِ الْغَيْبِ وَالشَّهَادَةِ فَيُنَبِّئُكُمْ بِمَا كُنْتُمْ تَعْمَلُونَ ﴾

صدق الله العظيم

سورة التوبة الآية (105)

DEDICATION

To:

The sake of Allah, our creator and our master. Alhamdulillah for everything, we can never thank Allah enough for the countless bounties he blessed us with.

To:

Our great teacher and messenger, Mohammed may Allah bless and grant him, who taught us the purpose of life.

To:

Our homeland Sudan, the warmest womb.

To:

The great martyrs and prisoners, the symbol of sacrifice.

To:

Sudan University of Science and Technology includes teachers, our second magnificent home and brothers.

To:

Our great parents, brothers, and sisters, who never stop giving of themselves in countless ways.

To:

Our friends who support us at every level.

ACKNOWLEDGEMENT

We would like to thank our supervisor **Ust. Omer Mohammed Salama Adam** for his dedicated support and guidance, he continuously provided encouragement and was always willing and enthusiastic to assist in any way he could throughout the project. The meetings and conversations were vital in inspiring us to think outside the box, from multiple perspectives to form a comprehensive and objective critique.

ABSTRACT

Natural disasters, Corona virus disease (COVID-19) and landmines all represent serious challenging to a large number of countries all over the world, and the classical systems used to face them have many limitations like higher cost, inefficient in many cases and others. Today quadcopters are the trend technologies used for these threatening to help in fight them, but stabilizing quadcopters under complex environment including system uncertainties, unknown noise and/or disturbance is so challenging and require a lot of experience with remote control to do tasks perfectly. This project aimed at designing a quadcopter system, stabilize it as good as possible using fuzzy and adaptive fuzzy control as examples of intelligent and adaptive control methods as well as to make the flight autonomous. Also, it aimed at implementation of a quadcopter vision system by using computer vision as new and future trends in quadcopters technology along with using mobile phone for live video streams, all these to make a system to face and deals with these threating effectively and with the lowest cost possible, and to illustrate examples of detecting and classifying fires/smokes, classify people as wearing face masks or not, detecting and recognition of landmines, and detecting objects such as people and animals in floods are chosen. The overall model is designed and implemented and the simulation is carried where results recorded and discussed, also it is observed from results that the system is achieving very good performance for both stabilization and vision.

مستخلص

الكوارث الطبيعية، مرض الكورونا (كوفيد–19)، و الألغام كلهم يشكلون تحدي حقيقي لعدد كبير من الدول في كل العالم، والأنظمة التقليدية المستخدمة لمواجهتهم لها العديد من القصور مثل التكلفة المرتفعة، عدم الكفاءة في العديد من الحالات و مشاكل أخرى. الآن الطائرات المسيرة الرباعية تمثل التكنولوجيا الرائدة المستخدمة لهذة المهددات للمساعدة في محاربتها، ولكن عملية استقرار الطائرة المسيرة الرباعية تحت البيئة المعقدة متضمنة حالات عدم التأكد في النظام، التشويش و/ أو الضوضاء غير المعروفان صعبة جدا و تتطلب الكثير من الخبرة مع ريموت التحكم لأداء المهام بكفاءة. هذا المشروع يهدف الى تصميم نظام طائرة مسيرة رباعية، جعلها مستقرة قدر الإمكان بإستخدام التحكم الغامض و التحكم الغامض التكيفي كأمثلة لطرق التحكم الحديثة والتكيفية و كذلك الطيران ذاتيا. أيضا يهدف الى تنفيذ نظام رؤية الطائرة بإستخدام الرؤية الحاسوبية كالتطورات الحالية و المستقبلية لتقنية الطائرات المسيرة الرباعية بالإستعانة بالهاتف المحمول للتصوير المباشر، كل هذا لعمل نظام يتعامل مع هذة المهددات بكفاءة و بأقل تكلفة ممكنة، و لتوضيح ذلك تم إختيار الأمثلة كشف و تصنيف الحريق/الدخان، تصنيف الأشخاص على أنهم يرتدون كمامات ام لا، الكشف و التعرف على الألغام، و الكشف عن الكائنات مثل الأشخاص و الحيوانات في الفيضانات. النظام الكلي تم تصميمة و تنفيذة و المحاكاة تم عملها حيث تم رصد النتائج و مناقشتها، ايضا تم ملاحظة من النتائج ان النظام يصل الى أداء جيد جدا لكل من الإستقرار و الرؤية.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

N	Negative
Z	Zero
P	Positive
GUM	Go Up Much
GU	Go Up
S	Stand
GDM	Go Down Much
GD	Go Down
NB	Negative Big
PB	Positive Big

LIST OF SYMBOLS

Symbol	Unit	Meaning
U or u	-	Control variable
U_1	N	Resultant force of lift force of the Quadcopter
U_2	N	Resultant force that affects the roll angle of the quadcopter
U_3	N	Resultant force that affects the pitch angle of the quadcopter
U_4	N	Resultant force that affects the yaw angle of the quadcopter
E or e	-	Error term
K_p	-	Proportional gain
K_i	-	Integral gain
K_d	-	Derivative gain
G_c	-	Controller transfer function
t	sec	Time
X	m	Position, x coordinate in body frame
Y	m	Position, y coordinate in body frame
Z	m	Position, z coordinate in body frame
x	m	Position, x coordinate in navigation frame
y	m	Position, y coordinate in navigation frame
z	m	Position, z coordinate in navigation frame
R	-	Rotation matrix

ϕ	rad	Roll angle
θ	rad	Pitch angle
ψ	rad	Yaw angle
p	rad/sec	Angular rate of change about X_b (Body Frame)
q	rad/sec	Angular rate of change about Y_b (Body Frame)
r	rad/sec	Angular rate of change about Z_b (Body Frame)
b	-	Body frame
n	-	Navigation frame
T	rad/sec	The Body fixed angular velocity to Euler rate matrix
F	N	Force
$\underline{r}$	m	Position vector $[X\ Y\ Z]^T$
$\underline{v}$	rad/s	Angular rate of change vector $[p\ q\ r]^T$
$\underline{\mathrm{v}}$	rad/s	Angular velocity vector in body frame $[\omega_x, \omega_y, \omega_z]^T$
m	kg	Total mass
M	N.m	Moment
I or J_r	$Kg.m^2$	inertia
l	m	Distance from the center of the frame to the center of the propeller
d	-	Constant which converts the thrust into moment
b	-	Lift Coefficient
w	N	Lift weight

g	m/s^2	Gravity acceleration
T_B	N	Thrust vector in body frame
τ	N.m	Torque
z^d	m	Desired altitude
z_{IR}	m	Altitude measured by the IR module
z_{SONAR}	m	Altitude measured by the SONAR module
ϕ^d, θ^d, ψ^d	rad	Desired roll, pitch and yaw angles
K_{Po}, K_{Io}, K_{Do}	-	Initial proportionality constants of PID controller

CHAPTER ONE

INTRODUCTION

CHAPTER ONE

INTRODUCTION

1.1 General Concepts

The rapid increase in technologies' development all over the world to adapt changes happen recently by trying to develop powerful software methods, algorithms, and networks that are capable of replacing classical hardware devices or reduce their costs, led to a great and extensive work on scientific research too in different fields to keep up the development of these technologies. One of the hottest research fields is the field of control engineering by developing intelligent control methods to replace the classical methods with many advantages like more flexibility, deals with complex and nonlinear plants with noise effectively, achieve optimal or near optimal stabilization and many other useful things.

On the other side the scientific research on machine learning specially computer vision algorithms and networks take a very huge concerns, because world now notice what is the powerful effect that computer vision can add when applied to critical and dangerous applications, recently like natural disasters, Corona virus disease (COVID-19) pandemic, military applications, delivery applications, and many others which serves humanity foremost. Also one of the reasons that makes world run toward uses of computer vision is that with computer vision you can build a powerful system with lowest possible cost, because all computer vision algorithms and networks are software and can be embedded in any device with little constraints even if an edge device.

Quadcopter, also known as quadrotor, is a helicopter with four rotors. The rotors are directed upwards and they are placed in a square formation with equal distance from the center of mass of the quadcopter. The quadcopter is controlled by adjusting the angular velocities of the rotors which are spun by electric motors. Quadcopter is a typical design for small unmanned aerial vehicles (UAV) because of the simple structure, and are used in surveillance, search and rescue, construction inspections and several other applications. Quadcopter currently being equipped with computer vision to form what is called a quadcopter vision system which is a new field of research these days and still under development with real time implementation in some countries for different important situations like detecting people and animals during flood period, ensure social distance between people and check whether people wearing masks or not during COVID-19 pandemic, landmines detection with the main aim of ensuring soldiers' safety and others.

Apply intelligent control methods to a complex non-linear system like a quadcopter with its 6 degrees of freedom (6 DOF) and uncertainties can prove actually the powerful effects of intelligent control over classical ones to obtain the best optimal control over trajectory control, position stabilization which leads to make quadcopter stable and be used in any sensitive application without any risks [30].

1.2 Problem Statements

Disasters that the world faced recently such as floods, landmines, fires, and lastly and till now COVID-19 pandemic represent a real huge challenge for humanity and environment, because of the resources and livings that are lost during these disasters. Sudan recently is affected by floods and COVID-19 pandemic very much and a huge loss are happened in environment as well as living things mainly because of the deteriorating economy and lack of capabilities. Classical systems for fighting these disasters have many limitations like non adaptation to situation, safety of workers, higher costs, and others, so recently quadcopters are used but they have many limitations too like stabilization and control them perfectly because of their 6 DOF and nonlinearity, also the problem of manual control which requires a large experience to make a good flight, and the problem that the distance to which quadcopters can be sent is not sufficient for real time dangerous situations like disasters, because of the small range of radio signal from the remote.

1.3 Objectives

- To be in accordance with new technologies and systems applied to fight disasters like quadcopter with lower cost than classical systems with much superior advantages, capabilities and results.
- To design and implement a complex, nonlinear system with 6 DOF and disturbance like a quadcopter.
- To stabilize a quadcopter using fuzzy and adaptive fuzzy control as examples of intelligent and adaptive control methods to investigate the power of these methods on such complex systems.
- To do autonomous flying so as to solve manual control issues and be in accordance with recent research of doing quadcopter missions by autonomous flying.
- To use internet of things technology to make a quadcopter system reach very long distances which is helpful in real time dangerous disasters.
- To use computer vision with a quadcopter to form a quadcopter vision system that is a hot topic recently in workshops and big conferences.
- To new trends of using edge devices such as mobile phones as a way to make a vision system in quadcopter which represents also an area of interest for current research in computer vision.
- To make the system applicable to be used in real world situations through using the vision system for detecting objects during floods, ensure that people are wearing masks during COVID-19 pandemic, detection of landmines which is used for military purposes, counting people and things for the purpose of security, detect and identify fires and smokes in forests for example during volcanos.

- To use some of the state of arts algorithms and networks for computer vision.

1.4 Methodology

Past and present literature reviews and researches were studied, then according to world trend technologies and the purpose of the project, the quadcopter system was chosen. The quadcopter system was first designed by calculating weights of components, thrust required, necessary flight time and other calculation according to the purpose, then the system was implemented and tested for flight. First it was observed that there was a problem in stabilization which was minimized by using fuzzy and adaptive fuzzy control methods, then also it was observed that to control the quadcopter perfectly need a lot of study with details for the parameters and other things, where here this problem was solved by making the flight autonomous. Also, for the purpose of increasing range or distance that the quadcopter can reach the internet of things (IoT) technology was implemented via using telemetry.

In order to implement a vision system certain algorithms and networks with their datasets and codes were prepared according to the project objectives, then these prepared codes were implemented through edge device which was mobile phone. The mobile was connected with the quadcopter via IoT technology and prepared for live video streams, after that the complete system was tested and results were recorded and discussed. Simulation diagrams were established for the quadcopter along with different controller after developing the mathematical models and the system was simulated for control the results were recorded and discussed, also an autonomous flying was simulated to test before apply. Computer vision algorithms and codes were implemented and the performance results also observed and be discussed.

1.5 Project Layout

This research consists of five chapters. Chapter one is an introduction which shows the general concepts, problem statements, objectives and methodology. Chapter two deals with literature review such as control systems and their design, artificial intelligence and some subfields of it, some common neural networks architectures used in computer vision, quadcopter, landmines, PID controller, fuzzy and adaptive fuzzy control and IoT. Chapter three deals with modelling process including different controllers. Chapter four deals with the design process, hardware and software implementations, operation of the system, simulation, the results and their discussions. Finally, chapter five includes the conclusion and recommendations.

CHAPTER TWO

LITERATURE REVIEW

CHAPTER TWO

LITERATURE REVIEW

2.1 Control Systems

When we use the word control in everyday life, we are referring to the act of producing a desired result. By this broad definition, control is seen to cover all artificial processes. A study of control involves developing a mathematical model for each component of the control system. We have twice used the word system without defining it. A system is a set of self-contained processes under study [1]. Figure 2.1 shows the general steps in designing a control system [2].

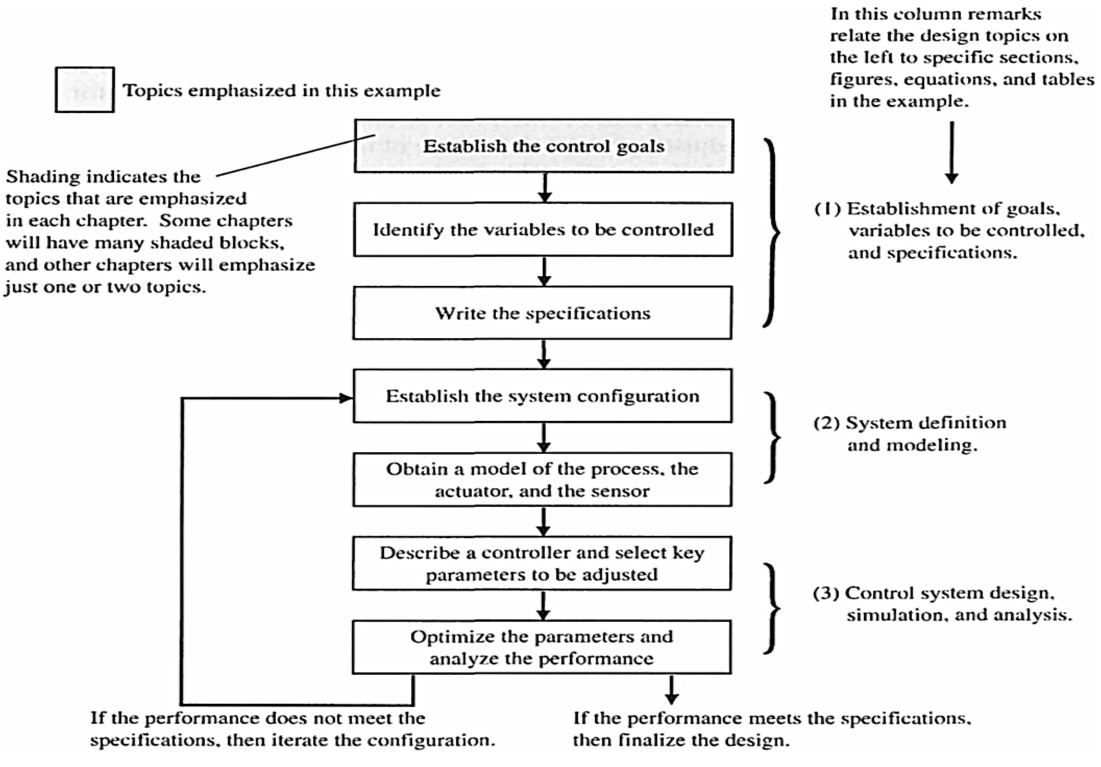


Figure 2.1: General steps for designing a control system

2.2 Artificial Intelligence

The definition of artificial intelligence (AI) actually not about one formal definition but it is a group of definitions according to different perspectives and fields, but the more common one is "AI is a characteristic of machines, computer programs and systems to perform intellectual and creative functions of humans, find ways to solve tasks, be able to make conclusions and decisions on their own" [3]. AI has research fields and research techniques. Research fields like intelligent machine, knowledge representation, machine earning, computer vision and others, while research techniques include optimization algorithms and soft computing techniques like neural networks, fuzzy logic control and expert systems.

2.3 Machine Learning

There are many definitions for machine learning (ML) over years according to different perspectives, the definition mention here is from engineering perspective: "a computer program is said to learn from experience (E) with respect to some task (T) and some performance measure (P), if its performance on (T), as measured by (P), improves with experience (E)" [3]. Figure 2.2 shows a schematic diagram about how a Machine learning solve problems [3].

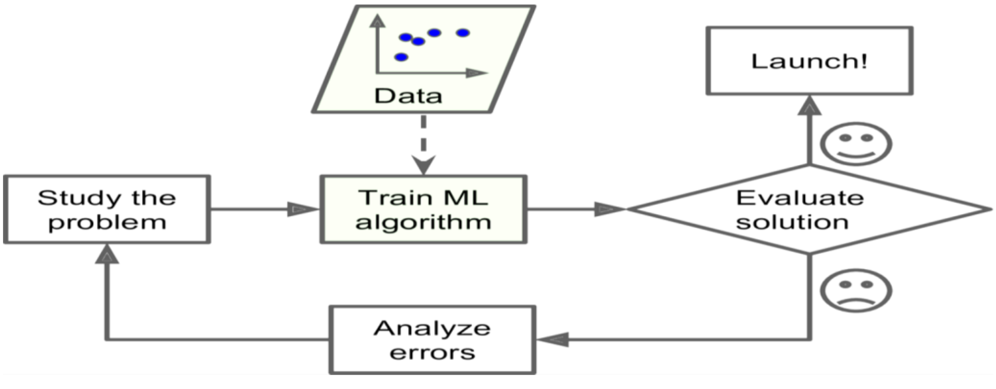


Figure 2.2: Machine learning approach for solving problems

Machine learning is great for many things like problems for which solutions require a lot of hand-tuning or long lists of rules, complex problems for which there is no good solution at all using a traditional approach, and others [3].

2.4 Deep Learning

Deep Learning (DL) is a specific subfield of machine learning: a new take on learning representations from data that puts an emphasis on learning successive layers of increasingly meaningful representations. The deep in deep learning isn't a reference to any kind of deeper understanding achieved by the approach; rather, it stands for this idea of successive layers of representations. How many layers contribute to a model of the data is called the depth of the model. Other appropriate names for the field could have been layered representations learning and hierarchical representations learning [4].

Modern deep learning often involves tens or even hundreds of successive layers of representations— and they're all learned automatically from exposure to training data. Meanwhile, other approaches to machine learning tend to focus on learning only one or two layers of representations of the data; hence, they're sometimes called shallow learning. In deep learning, these layered representations are (almost always) learned via models called neural networks, structured in literal layers stacked on top of each other. What do the representations learned by a deep-learning algorithm look like? Let's examine how a network several layers deep (see Figure 2.3) transforms an image of a digit in order to recognize what digit it is [4].

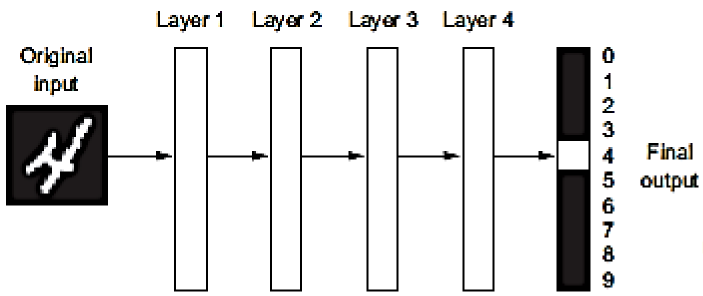


Figure 2.3: A deep neural network for digit classification

As seen in Figure 2.4, the network transforms the digit image into representations that are increasingly different from the original image and increasingly informative about the final result. You can think of a deep network as a multistage information-distillation operation, where information goes through successive filters and comes out increasingly purified (that is, useful with regard to some task) [4].

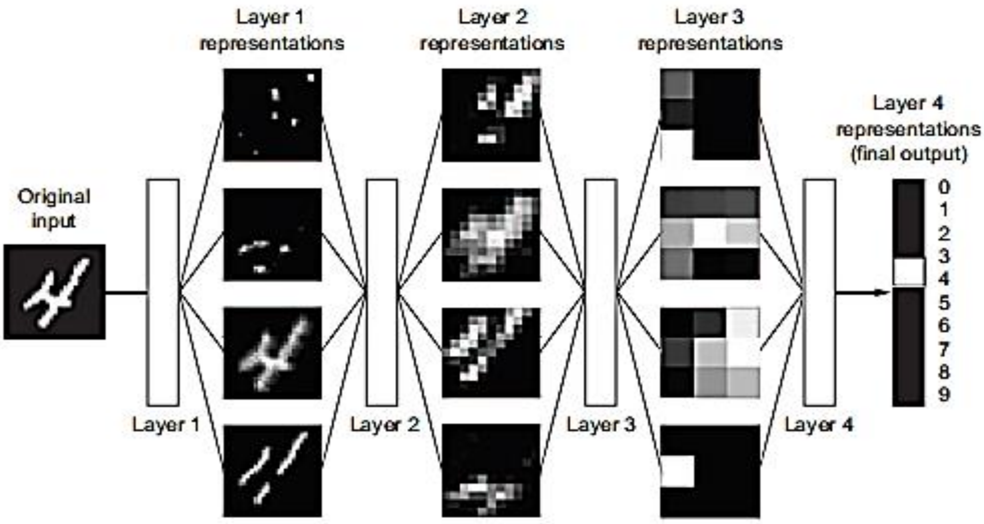


Figure 2.4: Deep representations learned by a digit-classification model

Convolutional neural network (CNN), is a class of artificial neural networks that has become dominant in various computer vision tasks, is attracting interest across a variety of domains, including radiology. CNN is designed to automatically and adaptively learn spatial hierarchies of features through backpropagation by using multiple building blocks, such as convolution layers, pooling layers, and fully connected layers. The CNN architecture includes several building blocks, such as convolution layers, pooling layers, and fully connected layers. A typical architecture consists of repetitions of a stack of several convolution layers and a pooling layer, followed by one or more fully connected layers. The step where input data are transformed into output through these layers is called forward propagation as shown in Figure 2.5. Although convolution and pooling operations described in this section are for 2D-CNN, similar operations can also be performed for three-dimensional (3D)-CNN [5].

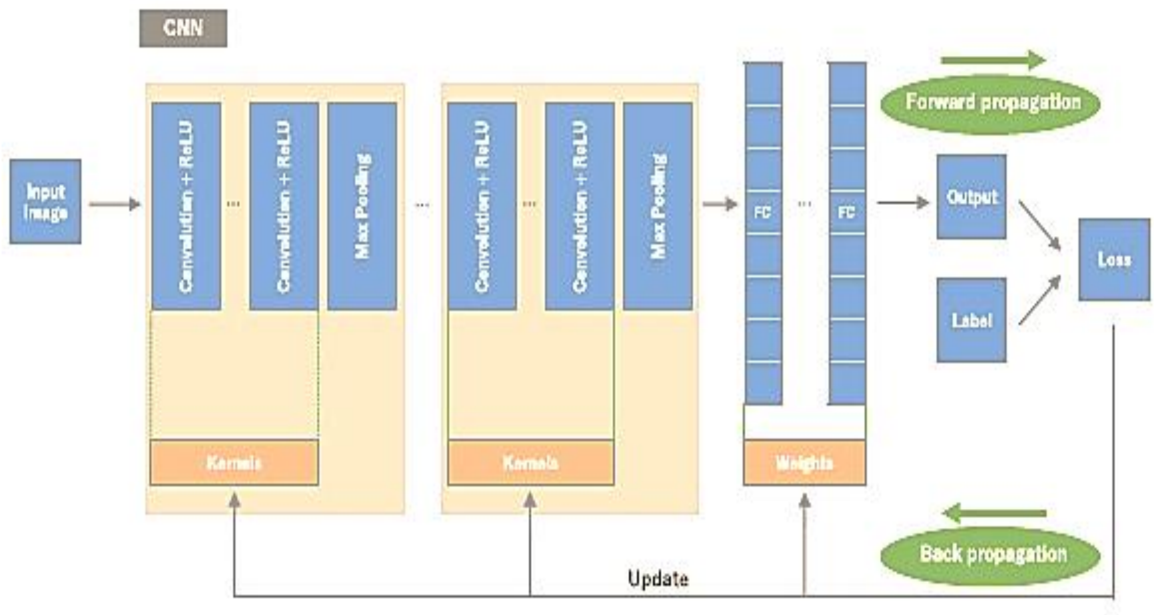


Figure 2.5: An overview of a convolutional neural network architecture and the training process

The functionalities of the said network are described as follows:

1) Convolutional layers: In this layer, the CNN uses numerous filters to convolve the entire image including intermediate feature maps & generating different feature maps.

2) Pooling Layers: This layer is similar to convolution layer but minimizes the measurements of feature maps and also the parameters of the network. Generally average and max pooling are used. Among all the three layers, the pooling layer happens to be the most profusely investigated. There exist mainly three approaches, with varied usage, that are related to the pooling layers.

3) Fully-connected layers: The final layer of CNN consists of 90% of the parameters. The feed forward network forms a vector of a particular length to follow up processing. Since these layers contain most of the parameters, there is a high computational burden while training the data [5]. Figure 2.6 shows these functionalities.

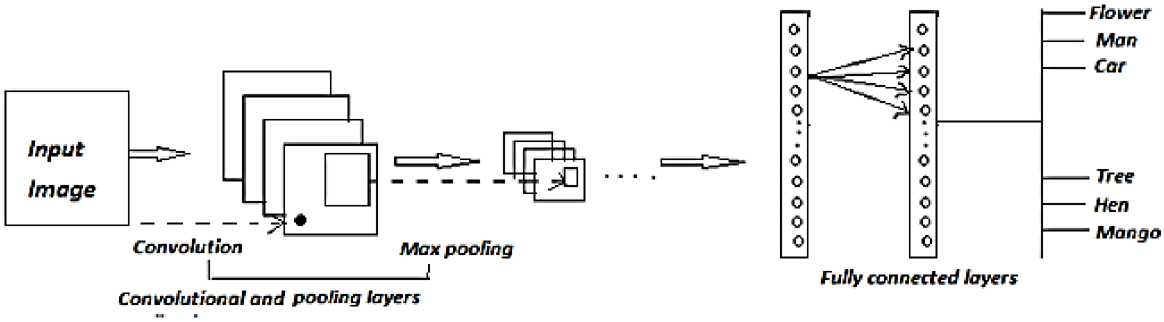


Figure 2.6: Pipelined architecture of convolutional neural network

2.5 MobileNetv2 Neural Network

The MobileNetv2 is a coevolutionary modulus that is particularly useful for mobile designs because it helps to completely materialize big intermediate tensors to greatly reduce the memory footprint needed during the inference. For several embedded hardware models with a minimal amount of cache operated by extremely fast software, this eliminates the need for principal memory access. The detailed structure of building block is a bottleneck depth-separable convolution with residuals is shown in Table 2.1 [6].

Table 2.1: The bottleneck residual block of MobileNetv2

Input	Operator	Output
$h \times w \times k$	1×1 conv2d, ReLU6	$h \times w \times (tk)$
$h \times w \times tk$	3×3 dwise s=s, ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	Linear 1×1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k'$

The original full-convolute layer of 32 filters and 19 bottleneck residual layers are included in the MobileNetv2 architecture. It uses ReLU6 as a non-linearity for low-computation due to its robustness. This often uses kernel size 3x3, and normalizes batch and dropout during processing, as is typical for modern networks. Designers have used a continuous network expansion limit but for the first layer. The MobileNetv2 convolution block is shown in Figure

2.7 [6]. Implementing the face mask detector to embedded products will reduce this facemask detection systems' production cost, which is why we chose to use this method.

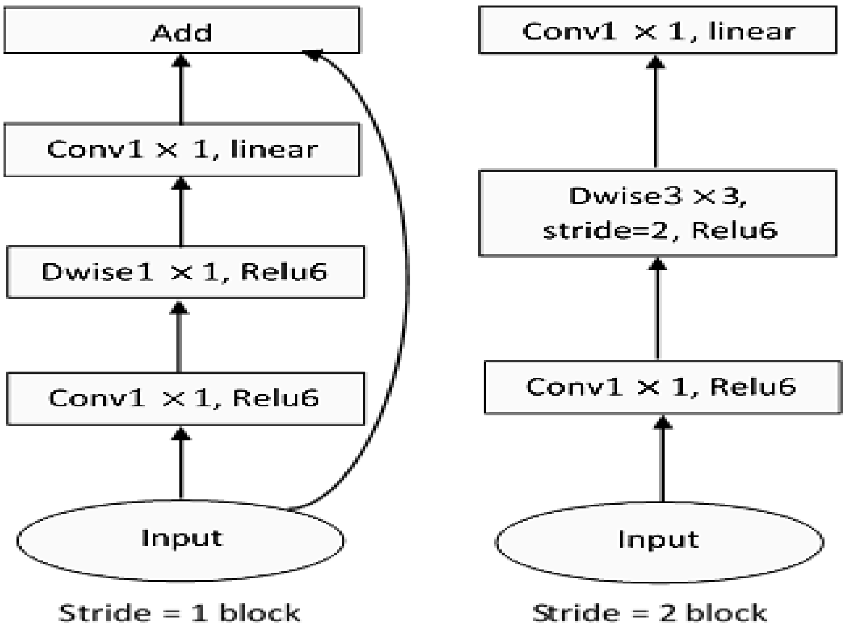


Figure 2.7: Convolution block for MobileNetv2

2.6 You Can Only Look Once Network

Object detection is one of the classical problems in computer vision where you work to recognize what and where, specifically what objects are inside a given image and also where they are in the image. The problem of object detection is more complex than classification, which also can recognize objects but doesn't indicate where the object is located in the image. In addition, classification doesn't work on images containing more than one object.

You Can Only Look Once (YOLO) uses a totally different approach. YOLO is a clever CNN for doing object detection in real-time. The algorithm applies a single neural network to the full image, and then divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities [7].

YOLO is popular because it achieves high accuracy while also being able to run in real-time. The algorithm "only looks once" at the image in the sense that it requires only one forward propagation pass through the neural network to make predictions. After non-max suppression (which makes sure the object detection algorithm only detects each object once), it then outputs recognized objects together with the bounding boxes. With YOLO, a single CNN simultaneously predicts multiple bounding boxes and class probabilities for those boxes. YOLO trains on full images and directly optimizes detection performance. This model has a number of benefits over other object detection methods [7]:

- YOLO is extremely fast
- YOLO sees the entire image during training and test time so it implicitly encodes contextual information about classes as well as their appearance.
- YOLO learns generalizable representations of objects so that when trained on natural images and tested on artwork, the algorithm outperforms other top detection methods.

Figure 2.8 shows the network architecture for YOLO [7].

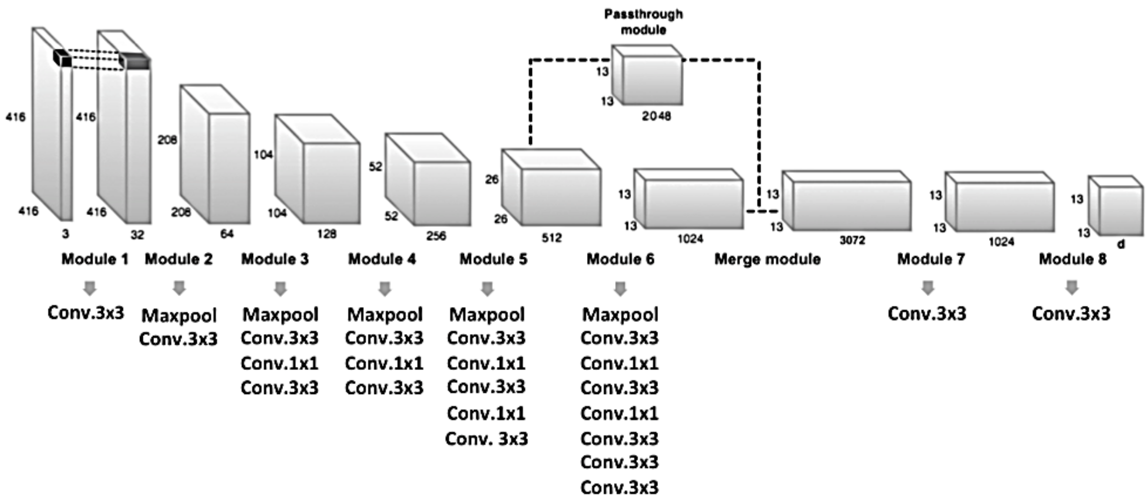


Figure 2.8: YOLO network architecture

2.7 Computer Vision

Computer Vision has a dual goal. From the biological science point of view, computer vision aims to come up with computational models of the human visual system. From the engineering point of view, computer vision aims to build autonomous systems which could perform some of the tasks which the human visual system can perform (and even surpass it in many cases). Many vision tasks are related to the extraction of 3D and temporal information from time-varying 2D data such as obtained by one or more television cameras, and more generally the understanding of such dynamic scenes. Of course, the two goals are intimately related. The properties and characteristics of the human visual system often give inspiration to engineers who are designing computer vision systems. Conversely, computer vision algorithms can offer insights into how the human visual system works. In this paper we shall adopt the engineering point of view [8].

Open source computer vision (OpenCV) is an open source computer vision library. The library is written in C and C++ and runs under Linux, Windows and Mac OS X. There is active development on interfaces for Python, Ruby, MATLAB, and other languages. OpenCV was designed for computational efficiency and with a strong focus on real-time applications [8].

The most applications of OpenCV are:

(1) Object detection

Object detection is a computer vision technique that works to identify and locate objects within an image or video. Specifically, object detection draws bounding boxes around these detected objects, which allow us to locate where said objects are in (or how they move through) a given scene [9]. Figure 2.9 shows the block diagram of this operation.

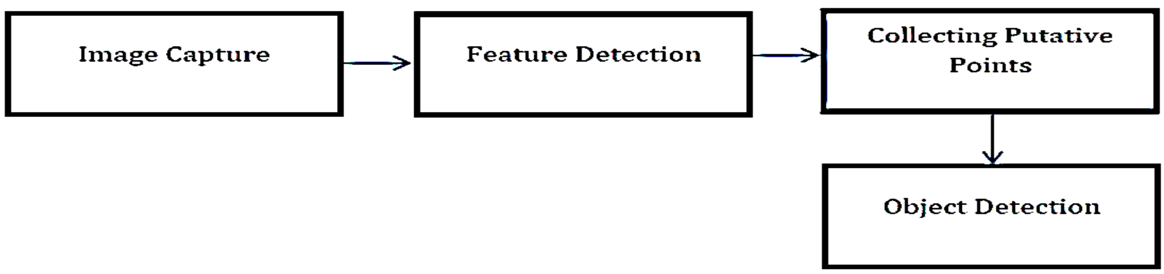


Figure 2.9: Object detection process

(2) Object recognition

Object recognition is concerned with determining the identity of an object being observed in the image from a set of known labels. Oftentimes, it is assumed that the object being observed has been detected or there is a single object in the image [10]. Figure 2.10 shows a general block diagram of object recognition operation.

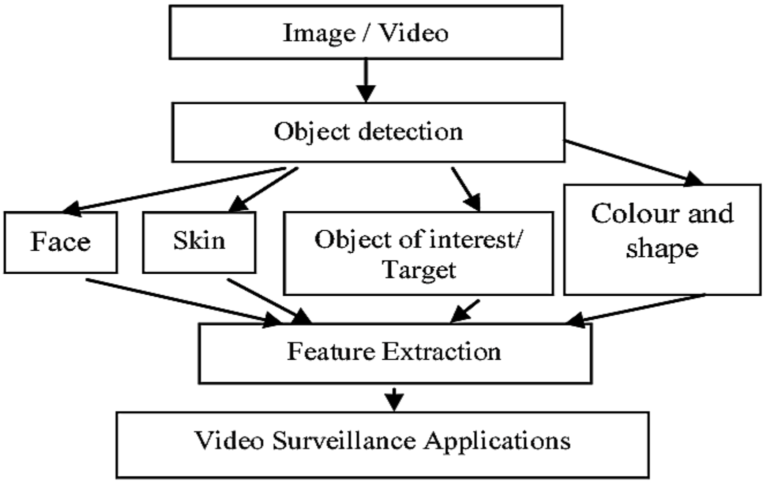


Figure 2.10: Object recognition process

(3) Object classification

A classifier is an algorithm that takes a set of features that characterize objects and uses them to determine the class of each object. The classic example in astronomy is distinguishing stars from galaxies. For each object, one measures a number of properties (speed, size, compactness, boundary box etc.); the classifier then uses these properties to determine whether each object is a star or a galaxy. There are of two types of classification supervised and unsupervised. In supervised classification, meaning that a human expert both has determined into what classes an object may be categorized and also has provided a set of sample objects with known classes. This set of known objects is called the training set because it is used by the classification programs to learn how to classify objects. In unsupervised classification methods in which the induction engine works directly from the data, and there is neither training sets nor pre-determined classes [11]. Figure 2.11 shows a diagram of steps of building object classifier.

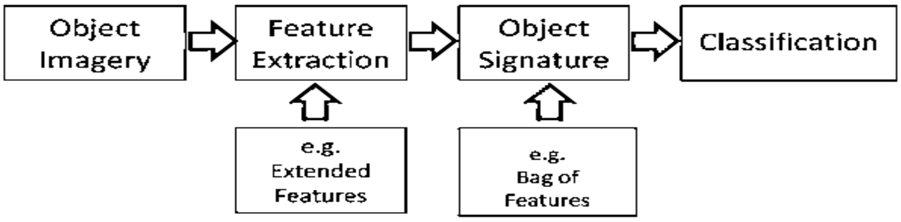


Figure 2.11: Steps for building an object classifier

(4) Haar cascade classifier

Haar Cascade is a machine learning object detection algorithm proposed by Paul Viola and Michael Jones in their paper "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade classifier is trained from a lot of positive and negative images (where positive images are those where the object to be detected is present, negative is those where it is not). It is then used to detect objects in other images. Luckily, OpenCV offers pre-trained Haar cascade algorithms, organized into categories (faces, eyes and so forth), depending on the images they have been trained on [8]. Figure 2.12 illustrates operation of this algorithm schematically.

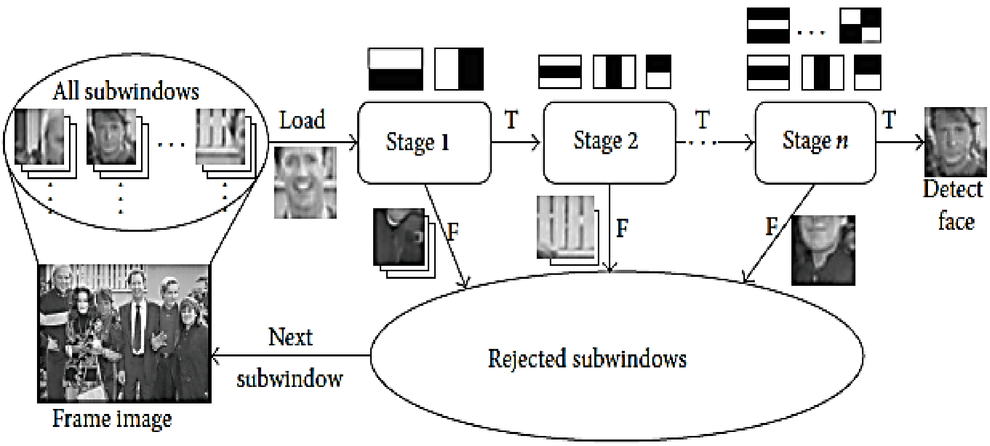


Figure 2.12: Illustration of Haar algorithm operation

(5) Aggregated channel features detector

Aggregated Channel Features (ACF) detector is an object detector network that uses a combined feature which consisting of three channels of LUV color space, a normalized gradient channel and a six-channel histogram of oriented gradient (HoG) and then arranged in a boosted tree. The general overview is shown in Figure 2.13. ACF detector will extract the proposal region consisting of the positive area and negative region. Positive proposal regions are obtained from training data containing the bounding box of the ground truth area. Meanwhile, the negative proposal region is extracted automatically by the sliding window in all image area except the bounding box of ground truth area [12].

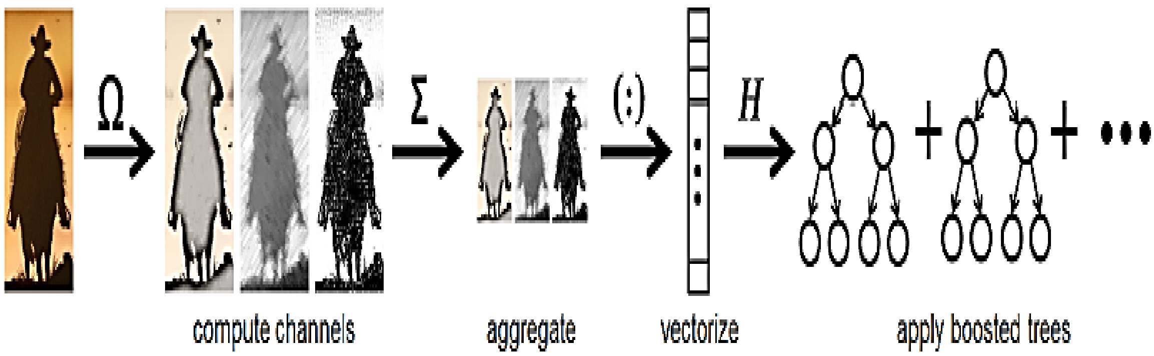


Figure 2.13: Overview of ACF detector

YOLO network represents one of the state of arts networks for object detection applications, while Mobile Net network represents a very common network used recently in object classification tasks. ACF object detector which implemented on MATLAB program represents new area of research for efficient object detection missions. Haar cascade classifier also used extensively by OpenCV for classification applications. Specially designed CNNS are developed recently for special tasks like FireDetectionNet that are developed mainly to build fires and smoke classifiers.

2.8 Quadcopter

Quadcopter is a type of multi-rotors helicopter. The word "quad" originated from a Latin word "quattro" which means four to indicate the number of rotors that give it the thrust it needs to move. Quadcopter also known as Quad rotor Helicopter, Quad rotor is a multi -rotor helicopter that is lifted and propelled by four rotors. Quadcopter are classified as rotorcraft, as opposed to fixed-wing aircraft, because their lift is generated by a set of rotors (vertically oriented propellers). Unlike most helicopters, Quadcopter uses two set of identical fixed pitched propellers: two clock wise and two counters- clockwise. These use variation of RPM to control loft and torque. Control of the Quadcopter is achieved by altering the rotation rate of one or more rotor discs, thereby changing it torque load and thrust/lift characteristics. A number of manned designs appeared in the 1920s. These vehicles were among the first successful heavier- than- air Vertical Take-Off and Landing (VTOL) vehicles. However, early prototypes suffered from poor performance, and latter prototypes required too much pilot work load, due to poor stability augmentation and limited control authority. More recently Quadcopter designs have become popular in Unmanned Aerial Vehicle (UAV) research. These vehicles use an electronic control system and electronic sensors to stabilize the aircraft. With their small size and agile maneuverability, these Quadcopter can be flown indoors as well as outdoors.

A typical Quadcopter is equipped with an inertial navigation unit (3 accelerometers, 3 gyroscopes and 3 magnetometers) for attitude determination, a barometer (outdoor) or an ultrasonic proximity sensor (indoor) for altitude measurements and optionally they come with a camera or GPS receiver [13]. Quadcopter has become very popular among researchers due to the advantages it offers. Like conventional helicopters it can take off and land vertically. In addition to this, it can fly closer to obstacles than the conventional helicopters, and it can make maneuvers that the conventional helicopters cannot make. But from the aerodynamic point of view quadcopter is inherently unstable. To control the quadcopter, many control schemes have been proposed. Robust feedback controllers based on H_∞ techniques, fuzzy control, PD controllers, back-stepping controllers and Neural-Network Adaptive flight control are just few examples of the control schemes proposed for controlling quadcopter [13]. Figure 2.14 shows quadcopter block diagram.

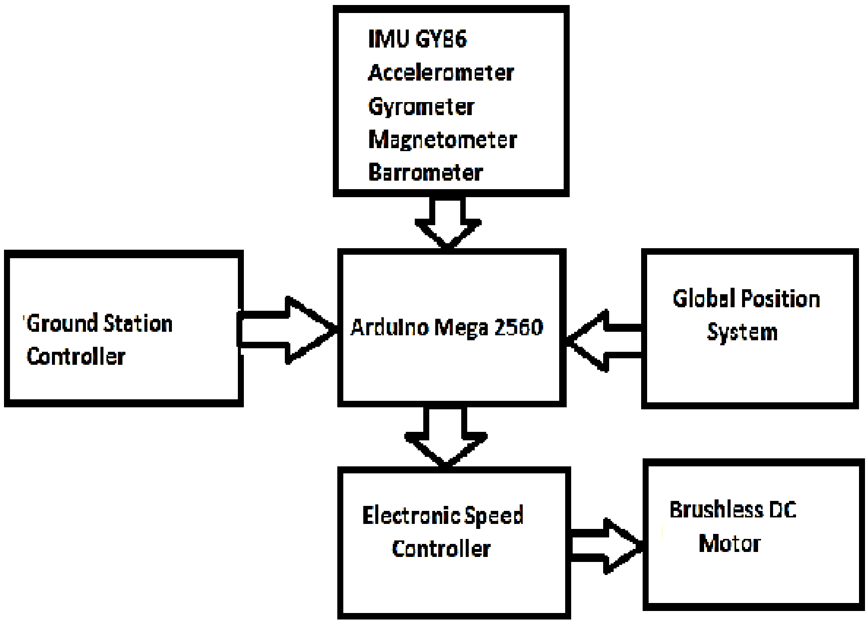


Figure 2.14: Quadcopter block diagram

The "controller" is being soft controller written in MATLAB for example will control the block "quadcopter" actually it controls the position controllers (yaw, pitch, roll and altitude or height) where the state of these controllers are measured through the different types of sensors like GPS, compass, camera, quadcopter's internal sensors [14].

There are two classifications of quadcopter depending on direction of Motor movement, plus (+) and X configuration. In plus (+) configuration every two-motor working in same direction and the other working in opposite direction as shown in Figure (2.15) to the left. In X configuration every two diagonal motor working in same direction and other two motor work in opposite direction as shown in Figure 2.15 to the right [15].

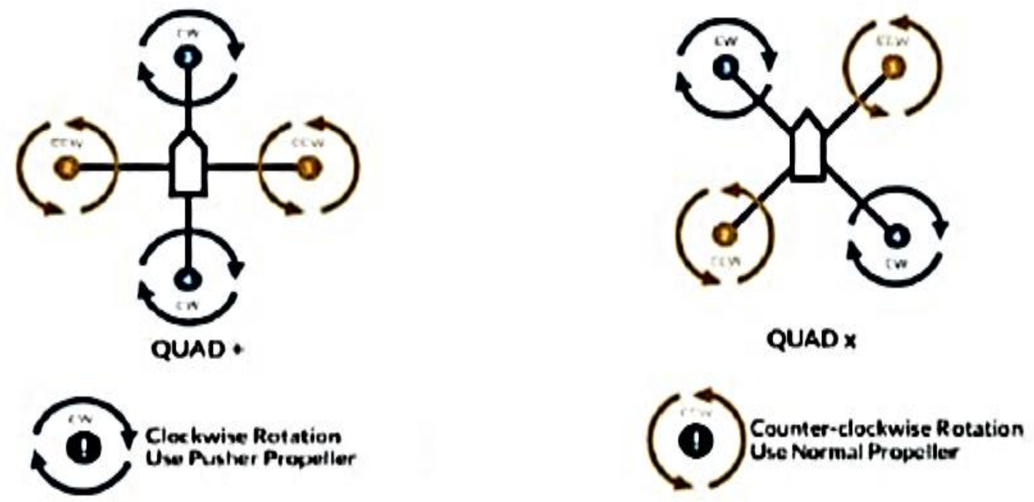


Figure 2.15: X and + configurations

There are a wide variety of advantages when using a quadcopter like simplicity of the control system, improved stability, safe to use for civilians because of small rotor, less complex mechanical structure, and others. Quadcopters are used for many important roles like in research platform which act as a tool for researchers to test and evaluate a new idea in a number of different fields, also a quadcopter can be used for commercial purposes as it is used these days, and many other important uses.

2.8.1 Quadcopter maneuverability

From the very first day in helicopter research, the quadcopter configuration was considered as an alternative. In a conventional helicopter layout, the torque produced by the main rotor is compensated by the tail rotor. When using two rotor configuration the torques created by the rotors are counteracted by each other. But two rotor configurations still have issues in control as the rotational and translational motions are highly dependent to each other. In four rotor configuration the control becomes simpler and the rotational motions are decoupled for the gyroscopic effects. Translational motions can be achieved by tilting the vehicle.

In the most common design for quadcopter, the two sets of rotors (1,3) and (2, 4) rotate in opposite directions as shown in Figure 2.16 by changing the rotor speeds, lift forces can be changed and motion can be created in desired direction. Changing the speeds of all four rotors generate vertical motion. Increasing or decreasing the speeds of rotor (2, 4) conversely produces roll rotation coupled with lateral motion. Pitch rotation and the corresponding lateral motion result from changing the speed of rotors (1, 3). Yaw rotation results from the difference in the counter-torque between the pairs of rotors [16].

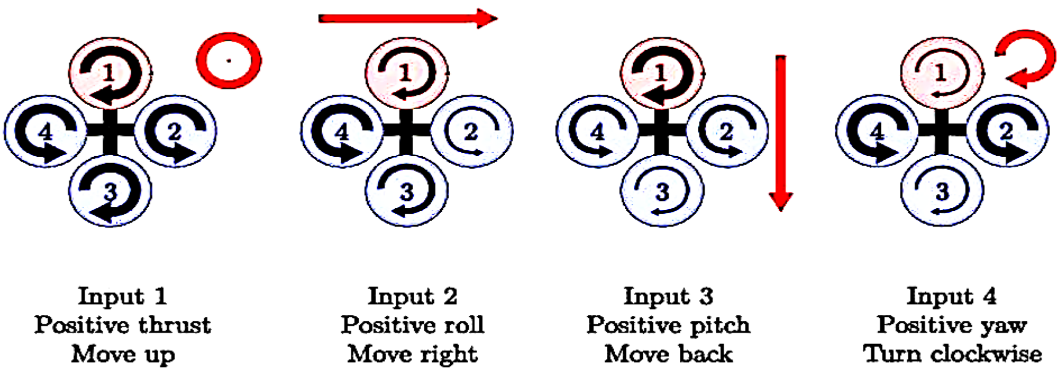


Figure 2.16: Quadcopter maneuverability

Quadrotors are three dimensional robots that need six degrees of freedom. The movement in the three-dimensional space is generated by coupling the linear movement (translation) with the angular movement (rotation) resulting in a robot that can linearly move through and rotate about the three-principal axis (x, y and z) [15].

(1) Altitude motion

The throttle movement is provided by increasing (or decreasing) the speed of all the rotors by the same amount. It leads a vertical force with respect to body-fixed frame which raises or lowers the quad rotor. When all actuators are at equal thrust, the craft will either hold in steady hover (assuming no disturbance) or increase/decrease altitude depending on actual thrust value.

(2) Roll motion

The roll movement is provided by increasing (or decreasing) the left rotor's speed and at the same time decreasing (or increasing) the right rotor's speed. It leads to a torque with respect to the central axis which makes the quad rotor to roll. The overall vertical thrust is the same as in hovering. If one of the actuators is decreased or increased on the roll axis as compared to the other actuator on the same axis, a roll motion will occur. In this instance, the craft would roll towards the right.

(3) Pitch motion

The pitch movement is provided by increasing (or decreasing) the front rotor's speed and at the same time decreasing (or increasing) the back-rotor's speed. It leads to a torque with respect to the central axis. The overall vertical thrust is the same as in hovering. Similar to the roll axis, if either actuator is changed on the pitch axis, the axis will rotate in the direction of the smaller thrust. In this instance, the craft nose would pitch up due to the differential on the pitch axis.

(4) Yaw motion

The yaw movement is provided by increasing (or decreasing) the front-rear rotor's speed and at the same time decreasing (or increasing) the left-right couple. It leads to a torque which makes the quad rotor turn in horizon level. The overall vertical thrust is the same as in hovering. If the clockwise spinning actuators are decreased (or the counter clockwise actuators increased), a net torque will be induced on the craft resulting in a yaw angle change. In this instance, a clockwise torque is induced.

2.8.2 Autonomous flying

Autonomous flying means to set up a mission on specific control station and asks the quadcopter to perform this mission autonomously by itself without human intervention. Autonomous quadcopters become so popular recently and represent an area of research to use autonomous quadcopters in many applications in industry and sensitive situations.

2.9 Landmines

A land mine is a type of self-contained explosive device which is placed onto or into the ground, exploding when triggered by a vehicle, a person, or an animal. The name originates from the practice of sapping, where tunnels were dug under opposing forces or fortifications and filled with explosives. Land mines generally refer to devices specifically manufactured for this purpose, as distinguished from improvised explosive devices ("IEDs"). It can also be defined as explosive charge buried just below the surface of the earth, used in military operations against troops and vehicles. It may be fired by the weight of vehicles or troops on it, the passage of time, or remote control. Though improvised land mines (buried artillery shells) were used in World War I, they only became important in warfare during World War II and have been widely used since. Most early mines had metal cases; later models were sometimes made of other materials to prevent magnetic detection [17]. The components of a typical landmine include the firing mechanism (including anti handling devices), the detonator (to set off the booster charge), the booster charge (may be attached to the fuse, or the ignitor, or to be part of the main charge), the main charge (in a container, usually forms the body of the mine) and the casing which contains all of the above parts [17]. Figure 2.17 shows these components.

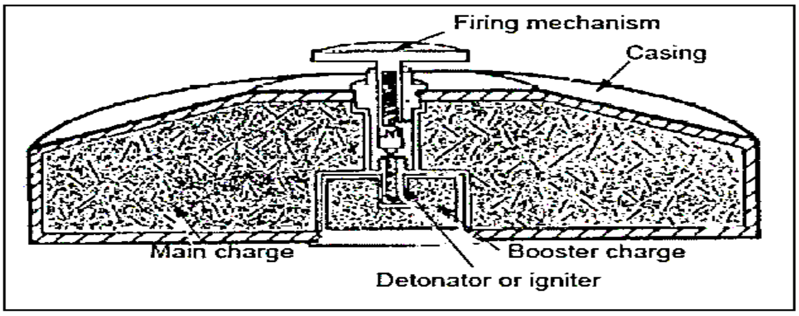


Figure 2.17: Landmine components

2.9.1 Types of landmines

Land mines are of two basic types; antitank and antipersonnel. Antitank mines are larger and more powerful than antipersonnel mines. However, antipersonnel mines are the most common type of mine, yet the most difficult to find because they are small and often made of plastic. Antitank mines generally contain more metal than do antipersonnel mines and are thus more easily detectable by simple metal detectors. Both types are buried as close to the surface as possible and are found in a variety of soils and terrain--rocky or sandy soil, open fields, forested areas, steep terrain, jungle. For both types of mines, detonation is typically caused by pressure, although some are activated by a trip-wire or other mechanisms. Thus, a land-mine detector must do its job without having direct contact with a mine. It also must be able to locate all types of mines individually in a variety of environments [17].

(a) Anti-tank mines

Anti-tank mines are designed to immobilize or destroy vehicles and their occupants. Anti-tank mines can achieve either a mobility kill (m-kill) or a catastrophic kill (k-kill). A mobility kill destroys one or more of the vehicle's vital drive components (for example, breaking a track on a tank) thus immobilizing the target. A mobility kill does not always destroy the weapon system or injure the crew. In a catastrophic kill, the weapon system and/or the crew are disabled. Figure 2.18 shows an example of this type.

Figure 2.18: Anti-tank mine

Anti-tank mines are typically larger than anti-personnel mines and require more pressure to detonate. The high trigger pressure (normally 100 kg (220 lb.)) prevents them from being set off by infantry. More modern anti-tank mines use shaped charges to cut through armor. These were first deployed in large numbers in World War II.

(b) Anti-personnel mines

Anti-personal mines are normally designed to kill or injure as many enemy combatants as possible. Smaller anti-personnel mines are sometimes designed to maim rather than kill in order to increase the logistical (mostly medical) support required by such an enemy force. Some types of anti-personnel mines can also damage the tracks on armored vehicles or the tires of wheeled vehicles. Figure 2.19 shows an example of this type.

Figure 2.19: Anti-personnel mine

2.9.2 Modern (intelligent) detection systems

These systems use improvements happened in the field of Control Engineering (like intelligent systems) accordingly with artificial intelligence techniques (especially machine learning techniques and software) to make processes on images captured by different ways to tell if this image is an image of a mine or not depending on many features and other things which are extracted from images. These intelligent systems for detection are still new and recently there are many scientific papers related to landmines appears which use these systems basically like:

- Vision system for detecting landmines using artificial neural networks (ANNs).
- Detection landmines using computer vision.
- Colpitts oscillator sensor detection method.
- Signal processing techniques for landmine detection using impulse ground penetrating radar.

2.10 Proportional- Integral- Derivative Controller

A Proportional-Integral-Derivative (PID) controller is basically a compensator connected in cascade with the closed-loop system. It controls the signal that governs the system by taking into account the error signal. Its name stems from proportional, integral, and derivative, which are the actions that each of its three parts performs. Depending on the properties of the system, several combinations of these actions can be used. In general, every closed-loop system with cascade compensation has the structure shown in Figure 2.20 [18].

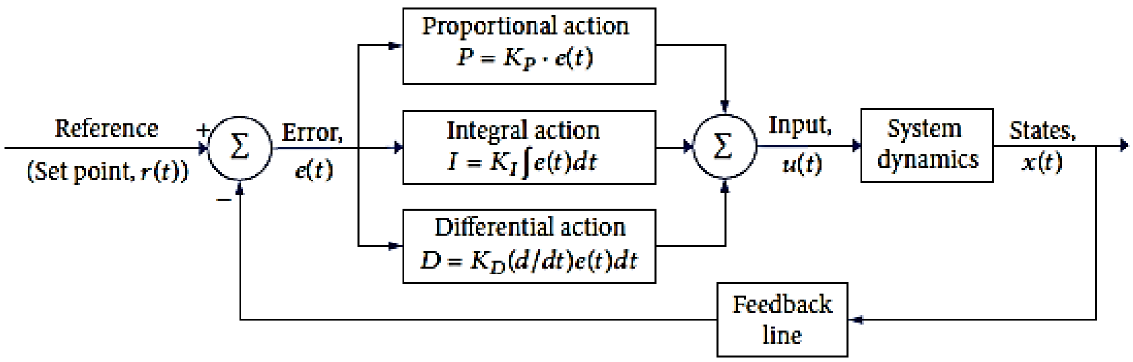


Figure 2.20: Control system using PID controller block diagram

A PID controller is sometimes called three-term controller and is described by the following equations:

$$u(t) = K_p\left(e(t) + \frac{1}{T_i}\int_0^t e(t)\,dt + T_d\frac{de(t)}{dt}\right) \tag{2.1}$$

And:

$$G_c(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s \tag{2.2}$$

The results of the action of every P, I, and D controllers in a closed-loop system are summarized in Table 2.2.

Table 2.2: Effects of P, I, and D controllers on a closed-loop system

Controller type	Rise time	Maximum overshoot	Settling time	Steady-state error
P	Decrease	Increase	Small change	Decrease
I	Decrease	Increase	Increase	Elimination
D	Small change	Decrease	Decrease	Small change

2.10.1 The proportional controller

With a proper choice of the proportional gain K_p of the proportional (P) controller, the steady-state oscillations are eliminated and the output signal of the closed-loop system is stabilized. However, the system tends to present steady-state error, which can be decreased but not eliminated. The equations that describe the P controller are:

$$u(t) = K_p e(t) \tag{2.3}$$

And:

$$G_c(s) = \frac{U(s)}{E(s)} = K_p \tag{2.4}$$

2.10.2 The derivative controller

With the addition of the integral term of the derivative (D) controller, the stability of the system increases and the behavior of the transient response improves. The mathematical expressions are:

$$u(t) = K_d \frac{de(t)}{dt} \tag{2.5}$$

And:

$$G_c(s) = \frac{U(s)}{E(s)} = K_d s \tag{2.6}$$

The most common used structure of PID controller for quadcopter's position is PD controller.

2.10.3 The Proportional-Derivative controller

The Proportional-Derivative (PD) controller allows the operation of the plant with larger (compared to a P controller) gain values. It diminishes the oscillations and it decreases steady-state error. The mathematical expression is:

$$u(t) = K_p e(t) + K_d \frac{de(t)}{dt} \tag{2.7}$$

And:

$$G_c(s) = \frac{U(s)}{E(s)} = K_p + K_d s \tag{2.8}$$

2.10.4 PID controller tuning

The process of selecting the controller parameters to meet given performance specifications is known as controller tuning. A change in the proportionality constants of these terms P, I and D changes the type of response of the system, that is why PID tuning, which is the variation of the PID proportionality constants, is of utmost importance. There have been various types of techniques applied for PID tuning, one of the earliest being the Ziegler Nichols technique. These techniques can be broadly classified as classical and computational or optimization techniques.

Classical techniques make certain assumptions about the plant and the desired output and try to obtain analytically, or graphically some feature of the process that is then used to decide the controller settings. These techniques are computationally very fast and simple to implement, and are good as a first iteration. But due to the assumptions made, the controller settings usually do not give the desired results directly and further tuning is required. A few classical techniques are Ziegler-Nichols methods and Cohen-Coon method.

Optimization techniques are usually used for data modeling and optimization of a cost function, and recently they have been used in PID tuning. Few examples are neural networks (computational models to simulate complex systems), Fuzzy logic control, genetic algorithm and differential evolution. The optimization techniques require a cost function they try to minimize. There are four types of cost functions used commonly which are Integral of Absolute Error (IAE), Integral of Square Error (ISE), Integral of Time Multiplied Absolute Error (ITAE) and Integral of Time Multiplied Squared Error (ITSE).

Computational models are used for self-tuning or auto tuning of PID controllers. Self-tuning of PID controllers essentially sets the PID parameters and also models the process by using some computational model and compares the outputs to see if there are any process variations, in which case the PID parameters are reset to give the desired response. There are various intelligent optimization techniques such as ant-colony optimization, particle swarm optimization, immune algorithm, genetic algorithms and many other optimization methods and algorithms [19].

2.11 Fuzzy Logic and Adaptive Fuzzy Logic Control

Fuzzy logic represents one type of intelligent control systems (or soft computing) techniques. Advances in Artificial Intelligence (AI), soft computing, and related scientific fields have brought new opportunities and challenges for researchers to deal with complex and uncertain problems and systems, which could not be solved by traditional methods. Many traditional approaches that have been developed for mathematically well-defined problems with accurate models may lack in autonomy and decision-making ability and hence cannot provide adequate solutions under uncertain and fuzzy environments. Intelligent systems represent a new approach to addressing those complex problems with uncertainties.

Intelligent systems are defined with such attributes as high degree of autonomy, reasoning under uncertainty, higher performance in a goal seeking manner, high level of abstraction, data fusion from a multitude of sensors, learning and adaptation in a heterogeneous environment, etc. [20]. Table 2.3 shows a comparison between classical and intelligent control.

Table 2.3: Comparison between classical and intelligent control

Comparison item	Classical Control	Intelligent Control
Basic Concept	Mathematical modeling-Designer designed the system which includes system dynamics	Abstract modeling-Designer input the behavior to the system and then system attempt to abstractly define the system
Characteristics	Need to have prior information about the system dynamics	Doesn't need to know all the system dynamics and conditions
Examples of methods	Open loop system	Fuzzy logic
	Closed loop system	Artificial Neural Networks

Fuzzy logic is a misnomer. Developed by Lotfi.A. Zadeh, it is firmly grounded in mathematical theory. Fuzzy logic control (FLC) is a digital control methodology that emulates human decision making by allowing partial truths i.e. multivalued logic. It allows for the imprecision inherent in real world scenarios. The output of a fuzzy system is smooth and continuous thereby lending itself to the control of continuously variable systems such as electric motors. In general, FLC is used when the system to be controlled is nonlinear, time variant and ill-defined [21].

2.11.1 Fuzzy logic control process

Fuzzy logic maps crisp inputs to crisp outputs according to the process shown in Figure 2.21 below.

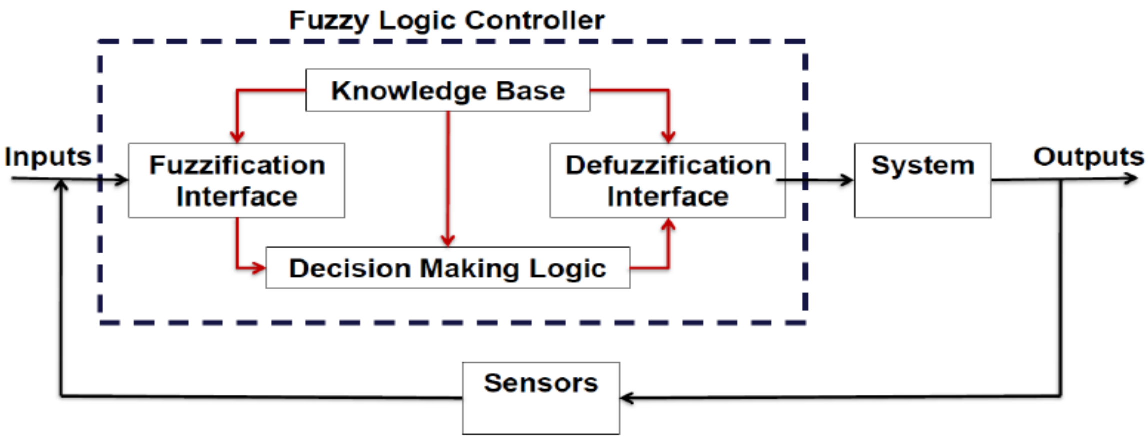


Figure 2.21: Block diagram of a fuzzy logic process for a system

The crisp inputs are first fuzzified i.e. converted into fuzzy membership functions, these input and output membership functions are then related by the inference system based on the rule base developed and then the fuzzy output is defuzzified to produce crisp outputs. Fuzzy logic offers a form of multivalued logic that is in sharp contrast to the binary logic in place today in computational world. It allows for the partial truths that are inherent in real world scenarios. The fuzzy logic control process is described below in more detail. A membership function is a function that assigns the degree of membership of a particular value says X of its entire domain. The value of MF always lies between the interval [0, 1] where 0 implies no membership and 1 implies complete membership. This is in contrast to typical, binary values that can either be only 0 or 1 i.e. a particular value X must belong completely or not at all to a particular classification. Furthermore, the domain of a value is typically broken into several membership functions that correspond to the ranges over which the input can be divided. These membership functions typically take the form of triangles, trapezoids, bell curves, Gaussian functions and/or sigmoid.

The assignment of the variable to membership functions is based on the designer's knowledge, reasoning, intuition etc. MF's can also be computer generated using artificial neural networks, genetic algorithms etc. [21]. It is worth mentioning here that designing FLC's is an extremely recursive effort where multiple simulations are done to observe performance and tweak the system by adjusting membership function shapes and relative placement on the Universe of discourse. Once a set of rules is created, they can be evaluated based on the crisp inputs given. If the value of the crisp input falls within the domain of the membership function, then the rules involving those membership functions "fire" and are evaluated. The degree to which a rule fires corresponds to the degree of membership of the crisp value in the membership function. Finally, all the outputs from the evaluated rules are totaled and defuzzified [21].

Defuzzification can be done in various ways. Two commonly used ones are: the mean of maximum method (MoM) where this method calculates the average of all variable values with max membership degrees as shown in Figure 2.22.

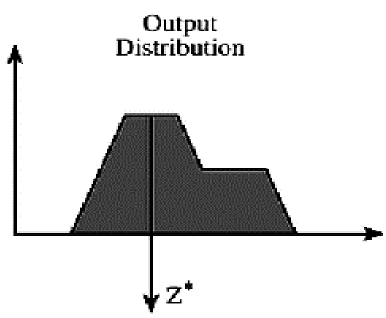


Figure 2.22: Defuzzify by Mean of Maximum (MoM) method

The other method is called the center of area or centroid method (CoA) where in this method the weighted average of the fuzzy set is calculated as shown in Figure 2.23.

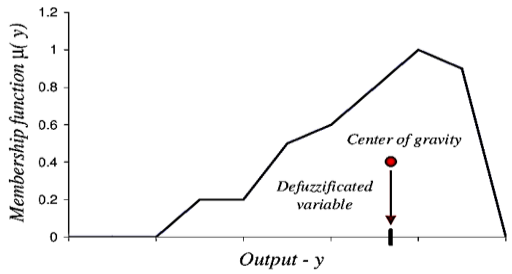


Figure 2.23: Defuzzify by Center of Area (CoA) method

2.11.2 Adaptive fuzzy PID control

Adaptive control scheme is used in order to let the real time system change its value to any external error by itself. As the name suggests this control technique is a mixture of two control techniques which are fuzzy and PID control as shown in block diagram of Figure 2.24.

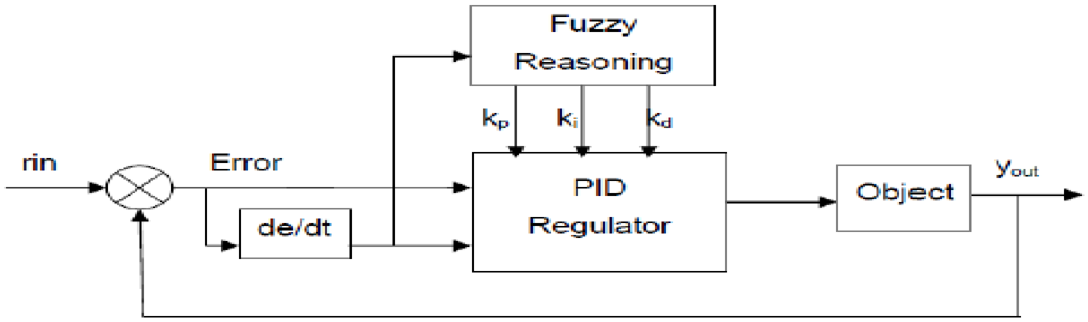


Figure 2.24: Adaptive fuzzy PID control system

As known adaptive control is basically a method which can tune and operate a system in a real time environment. Conventional PD controllers are the most used controller in industrial applications and position controllers but in robotic applications the biggest drawbacks of these controllers are overshoots and oscillation near settling points so this can be improved using a fuzzy logic control. When we design an adaptive controller, basic principle is to determine the relationship of fuzzy logic with the parameters of classical PD controller and also with the error and change of error. The error and change of error are supplied to the fuzzy logic controller and we get change in K_p and K_d as an output from the fuzzy logic controller [21].

2.12 Internet of Things

Internet of Things is referred to the general idea of things, especially everyday objects, that are readable, recognizable, locatable, addressable through information sensing device and/or controllable via the Internet, irrespective of the communication means (whether via RFID, wireless LAN, wide area networks, or other means). Everyday objects include not only the electronic devices we encounter or the products of higher technological development such as vehicles and equipment but things that we do not ordinarily think of as electronic at all such as food, clothing, chair, animal, tree, water, etc. [22]

2.13 Some Natural Disasters Face World Currently

The world today faces serious natural disasters commonly like floods, corona virus disease 2019 (COVID-19), military issues. Floods recently specially in Sudan kills thousands of people and a lot of resources are used to fight them and save people and things. On the other side COVID-19 and until now kills millions of people during first period or wave and it is expected that during it is second period which is start actually it will be more powerful and dangerous than the first period. Military issues either with other countries or inside the same country -like landmines- causes a lot of deaths and a lot of losses in natural resources specially for environment.

CHAPTER THREE

DESIGN, MODELING AND IMPLEMENTATION

CHAPTER THREE

DESIGN, MODELING AND IMPLEMENTATION

3.1 Design Process

The design process was begin by determine the objective of the project (here is doing machine learning -especially computer vision- tasks), and control goals (here it is the stabilization of quadcopter's position), the next step was to calculate weights for the necessary components -so as to choose the right frame with appropriate weight to fly properly- and to identify the angles to be controlled (here they are attitude angles and altitude angle). After that the configuration for the complete model was established -to do tasks perfectly- and the different controllers were designed and cascaded with the model to study the control problem under study-which was stabilization problem-. Finally, the controllers must be tuned perfectly and the tasks must be tested to see performance of the model. Figure 3.1 illustrates the design by shown electrical model.

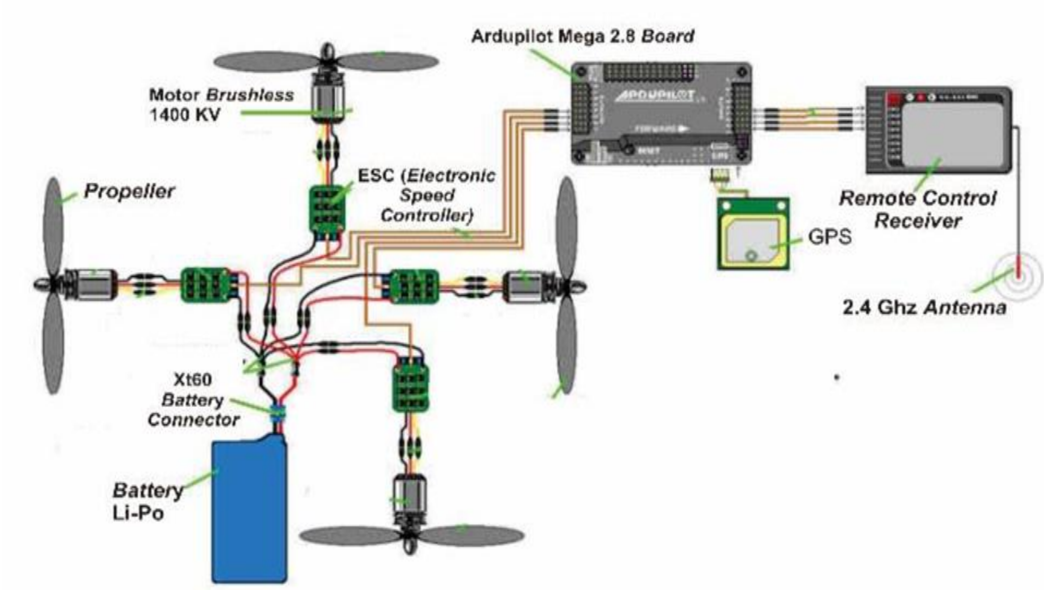


Figure 3.1: Electrical model for a quadcopter

3.2 Multirotor Mathematical Model Concepts

Mathematical model describes multirotor movement and behavior with the respect to the input values of the model and external influences on multirotor. Mathematical model can be considered as a function that is mapping inputs on outputs. By using mathematical model, it is possible to predict position and attitude of multirotor by knowing the angular velocities of propellers, i.e. it enables computer simulation of multirotor behavior in different conditions [23]. Figure 3.2 shows inputs and outputs of multirotor mathematical model.

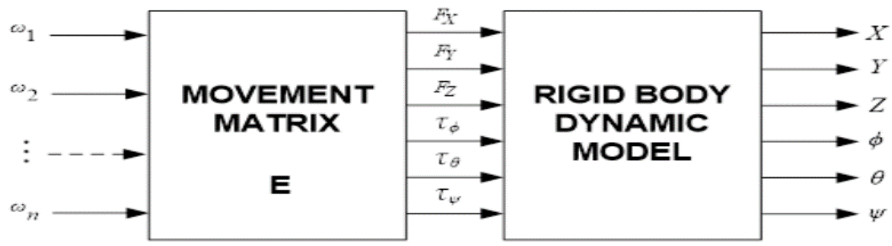


Figure 3.2: Inputs and outputs of a multirotor model

3.2.1 Coordinate system

When the quadcopter is navigating in three-dimensional space there are two different coordinate systems present. One is the body coordinate system, which is affected by the motors. The other is the navigation frame, were forces like gravitation has influence. The body coordinate system will move along with the quadcopter, while the navigation coordinate system is the reference point for the quadcopter. The reference coordinate system can be placed anywhere, but has to be fixed once the quadcopter starts moving [24]. Figure 3.3 shows these systems where the blue coordinates are the navigation frame, while the black is the body coordinates.

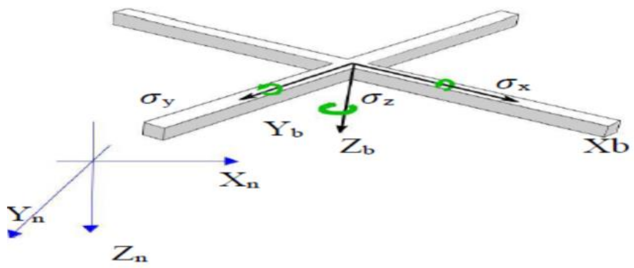


Figure 3.3: Body frame and navigation frame

In order to get the orientation of the quadcopter in terms of angles, the quadcopters coordinate system has to be linked to the navigation coordinates. The way this is done is by the implementation of Euler angles. There are three Euler angles, known as roll, pitch and yaw. This notation is often used in avionics. The transformation relationship between the body coordinate system and the ground coordinate system. The transformation relation of the coordinate system is related to the attitude angle: Φ, θ, ψ as shown in Figure 3.4.

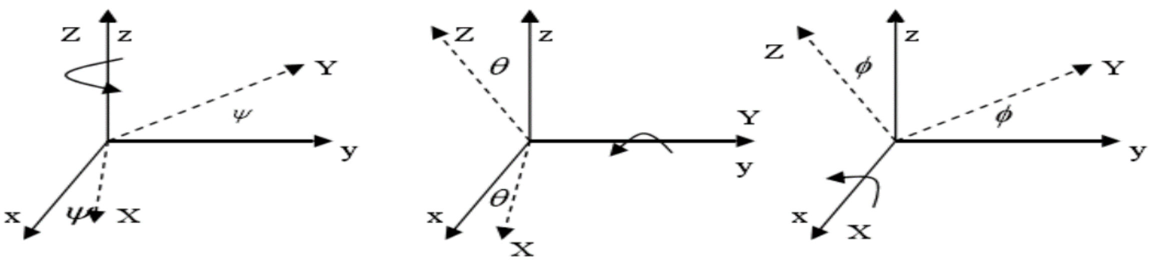


Figure 3.4: Transformation relationships between coordinate system and attitude angles

From Figure 3.4, the transformation from the body coordinate system to the ground coordinate system is described as follows: Using the right-hand rule, a new coordinate system E_1 is obtained by rotating the body coordinate system B (oxyz) around its z axis. The new coordinate system E_2 is obtained by rotating the angle θ on the y axis. Finally, E_2 is rotated around the x axis to obtain the final ground coordinate system E (OXYZ). From B (oxyz) to E_1, from E_1 to E_2, and from E_2 to E (OXYZ), the corresponding transformation matrix is:

$$R = \begin{bmatrix} cos\psi cos\theta & cos\psi sin\theta sin\phi - sin\psi \cos\phi & cos\psi sin\theta cos\phi + sin\psi \sin\phi \\ sin\psi cos\theta & sin\psi sin\theta sin\phi + cos\psi \cos\phi & sin\psi sin\theta cos\phi + cos\psi \sin\phi \\ -sin\theta & cos\theta sin\phi & cos\theta cos\phi \end{bmatrix} \quad (3.1)$$

Euler angular velocity and the body angular velocity conversion: Before establishing the kinetic model, the relationship between the angular velocity and the body angular velocity is:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -sin\theta \\ 0 & \cos\phi & cos\theta sin\phi \\ 0 & -\sin\phi & cos\theta cos\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \dot{\phi} - \dot{\psi} sin\theta \\ \dot{\theta} cos\phi + \dot{\psi} cos\theta sin\phi \\ -\dot{\theta} sin\phi + \dot{\psi} cos\theta cos\phi \end{bmatrix} \quad (3.2)$$

Which can also be written as:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & tan\theta sin\phi & tan\theta cos\phi \\ 0 & \cos\phi & -sin\phi \\ 0 & sec\theta sin\phi & sec\theta cos\phi \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} (p\ cos\theta + q\ sin\theta sin\phi + r\ sin\theta cos\phi)/cos\theta \\ (q\ cos\phi - r\ sin\phi) \\ (q\ sin\phi + r\ cos\phi)/cos\theta \end{bmatrix} \quad (3.3)$$

Equation (3.4) can be reduced to a standard unit matrix when the four rotors are stabilized in flight, the roll angle and the pitch angle are small, and the corresponding angular rate is also small [24]:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (3.4)$$

3.2.2 Motor reference

The motors rotation and position are also of significant. Motor one and two are placed along the X-axis and rotate the in the clockwise direction. Motor three and four are placed on the Y-axis and rotate in the counterclockwise direction as in Figure 3.5 [24].

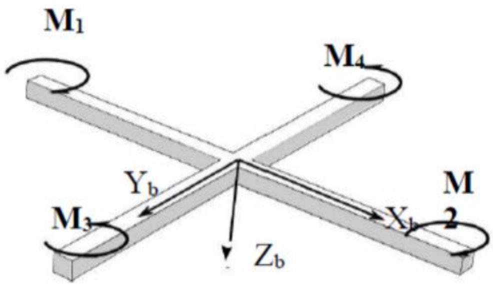


Figure 3.5: Motor reference

3.3 Quadcopter Dynamics

There are two different methods to achieve dynamics of a quadcopter, either using Lagrangian equations or Newton’s law, but it is much easier to understand using Newton’s law. Since the Quadcopter is controlled by varying the rotational speed of the four rotors attached to it, the altitude, change in position and velocity is caused by the increase or decrease of the speed of the four propellers together. By varying the speed of the first and third of propeller, the aircraft will tilt about the ‘y’ axis, which represents the pitch angle theta "θ". By varying the speed of the second and fourth propellers, the aircraft will tilt about the ‘x’ axis, which represents the roll angle phi "ϕ" [25].

3.3.1 Rotation matrix

The rotational matrix is:

$$R = \begin{bmatrix} cos\psi cos\theta & cos\psi sin\theta sin\phi - sin\psi cos\phi & cos\psi sin\theta cos\phi + sin\psi sin\phi \\ sin\psi cos\theta & sin\psi sin\theta sin\phi + cos\psi cos\phi & sin\psi sin\theta cos\phi + cos\psi sin\phi \\ -sin\theta & cos\theta sin\phi & cos\theta cos\phi \end{bmatrix} \quad (3.5)$$

The rotational matrix, is used to make a total conversion from the body frame to the navigational frame and the transpose of the rotational matrix, is used to make a total conversion from the navigational frame to the body frame [25].

3.3.2 Body fixed angular velocity to Euler rate matrix

The body fixed angular velocity to Euler rate matrix is:

$$T = \begin{bmatrix} 1 & tan\theta sin\phi & tan\theta cos\phi \\ 0 & \cos\phi & -sin\phi \\ 0 & sec\theta sin\phi & sec\theta cos\phi \end{bmatrix} \tag{3.6}$$

3.3.3 Forces

The forces acting on the system include the motor thrust, gravity, drag, airflow and disturbance forces. Using Newton's first law:

$$F_{tot_b} = m\underline{\ddot{r}_b} + m\left(\underline{v}\ x\ \underline{\dot{r}_b}\right) = m\begin{bmatrix} \ddot{X}_b \\ \ddot{Y}_b \\ \ddot{Z}_b \end{bmatrix} + m\left(\begin{bmatrix} p \\ q \\ r \end{bmatrix}\ x\ \begin{bmatrix} \dot{X}_b \\ \dot{Y}_b \\ \dot{Z}_b \end{bmatrix}\right) \tag{3.7}$$

$$F_{tot_b} = -F_{thrust_b} + F_{gravity_b} + F_{disturbance_b} + F_{drag} + F_{airframe_b} \tag{3.8}$$

3.3.4 Force equation

Using Newton's first law, the sum of the forces acting on the quadcopter was equated to the product of the mass of the system [25].

$$m\underline{\ddot{r}_b} = -F_{thrust_b} + F_{gravity_b} + F_{disturbance_b} + F_{drag} + F_{airfeame_b} \tag{3.9}$$

$$m\underline{\ddot{r}_b} = -\begin{bmatrix} 0 \\ 0 \\ F_{rotor1} + F_{rotor2} + F_{rotor3} + F_{rotor4} \end{bmatrix} + R^T.m.\begin{bmatrix} 0 \\ 0 \\ 9.81 \end{bmatrix} + F_{disturbance_b} + m.\left(\underline{v}\ x\ \underline{\dot{r}_b}\right) \tag{3.10}$$

3.3.5 Acceleration

By rearranging Equation (3.10) and making the acceleration vector the subject of the formula, the acceleration of the quadcopter in the body frame can be determined and by the use of integration, the velocity in body frame can be determined [25].

$$\begin{bmatrix}\ddot{X}_b\\ \ddot{Y}_b\\ \ddot{Z}_b\end{bmatrix} = \left(\frac{1}{m}\right)\begin{Bmatrix} -\begin{bmatrix}0\\0\\F_{rotor1}+F_{rotor2}+F_{rotor3}+F_{rotor4}\end{bmatrix} + \\ R^T.m.\begin{bmatrix}0\\0\\9.81\end{bmatrix} + F_{disturbance_b}\end{Bmatrix} + \left(\begin{bmatrix}p\\q\\r\end{bmatrix} x \begin{bmatrix}\dot{X}_b\\ \dot{Y}_b\\ \dot{Z}_b\end{bmatrix}\right) \quad (3.11)$$

In order to obtain the acceleration of the quadcopter in the navigation or earth frame, the rotational matrix needs to be applied to the acceleration vector in body frame, which can be seen in Equation (3.12) [25].

$$\begin{bmatrix}\ddot{X}_n\\ \ddot{Y}_n\\ \ddot{Z}_n\end{bmatrix} = R.\begin{bmatrix}\ddot{X}_b\\ \ddot{Y}_b\\ \ddot{Z}_b\end{bmatrix} \quad (3.12)$$

The forces acting on the system determine the moments about the various axis. The moments of the system can be described by the following equations [25]:

$$M_{total} = \underline{I}.\underline{\dot{v}} + \underline{v}\ x\ (\underline{I}.\underline{v}) \quad (3.13)$$

$$M_{total} = M_{thrust} + M_{motor\ inertia} \quad (3.14)$$

The thrust moments about the x, y and z axis is influenced by the thrust generated by the four rotors. The moment about the x and y axis is found by subtracting the forces generated by the rotors on the same axis and multiplying the result by the distance *l*, which is the distance from the center of the propeller to the center of the quadcopter. The z moment is found by subtracting the total force generated by the counter clockwise spinning rotors from the force generated by the clockwise spinning rotors and multiplying the result by the constant (d) [25].

$$M_{thrust} = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} \tag{3.15}$$

$$M_x = (F_{rotor3} - F_{rotor4}).\, l \tag{3.16}$$

$$M_y = (F_{rotor1} - F_{rotor2}).\, l \tag{3.17}$$

$$M_z = (F_{rotor1} + F_{rotor2} - F_{rotor3} - F_{rotor4}).\, d \tag{3.18}$$

3.3.6 Moment equations

The total moment of the system can be calculated and equated as [25]:

$$M_{thrust} + M_{motor_{inertia}} = \underline{I}.\, \underline{\dot{v}} + \underline{v}\ x\ (\underline{I}.\, \underline{v}) \tag{3.20}$$

By making the angular acceleration the subject, we get:

$$\underline{I}.\, \underline{\dot{v}} = M_{thrust} + M_{motor_{inertia}} - \underline{v}\ x\ (\underline{I}.\, \underline{v}) \tag{3.21}$$

$$\underline{\dot{v}} = \underline{I}^{-1}.\, [M_{thrust} + M_{motor_{inertia}} - \underline{v}\ x\ (\underline{I}.\, \underline{v})] \tag{3.22}$$

Therefore:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \underline{I}^{-1}.\, [M_{thrust} + M_{motor_{inertia}} - \underline{v}\ x\ (\underline{I}.\, \underline{v})] \tag{3.23}$$

To obtain the Euler rates of the quadcopter, the angular velocity of the quadcopter in the body frame is multiplied by the body fixed angular velocity to Euler rate matrix which is shown below [25]:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = T.\begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.24}$$

The Euler angles can thereafter be obtained by integration.

3.4 Quadcopter Kinematics

Before delving into the physics of quadcopter motion, let formalize the kinematics in the body and inertial frames. The position and velocity of the quadcopter in the inertial frame defined as $x = (x, y, z)^{T}$ and $\dot{x} = (\dot{x}, \dot{y}, \dot{z})^{T}$, respectively. Similarly, the roll, pitch, and yaw angles in the body frame are defined as $\theta = (\theta, \phi, \psi)^{T}$, with corresponding angular velocities equal to $\dot{\theta} = (\dot{\theta}, \dot{\phi}, \dot{\psi})^{T}$. However, it is noted that the angular velocity vector $\underline{v} = \dot{\theta}$. The angular velocity is a vector pointing along the axis of rotation, while$\dot{\theta}$ is just the time derivative of yaw, pitch and roll. To convert these angular velocities into the angular velocity vector, the following relation is used [24]:

$$\underline{v} = \begin{bmatrix} 1 & 0 & -sin\theta \\ 0 & \cos\phi & cos\theta sin\phi \\ 0 & -\sin\phi & cos\theta cos\phi \end{bmatrix} \tag{3.25}$$

The body and inertial frame are related by a rotation matrix (R) which goes from the body frame to the inertial frame. This matrix is derived by using the XYZ Euler angle conventions and successively “undoing” the yaw, pitch, and roll [24].

3.4.1 Kinetic model

Figure 3.6 is the force analysis chart, quadcopter during the movement by the force and torque are: the aircraft's own gravity, along the three-axis translation of the air resistance, rotation around the three-axis torque, four rotor Rotation of the lift, three axial rotation torque, the rotor rotation generated by Coriolis torque, vibration and external disturbance generated torque.

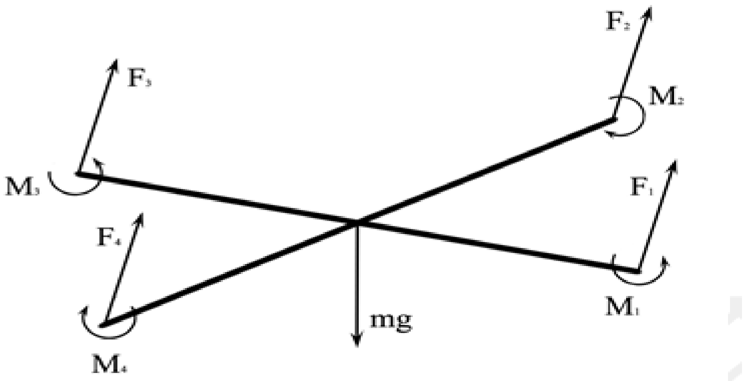


Figure 3.6: Force analysis chart

After completion of the force analysis, per Newton's law of motion and Euler equation, based on the force analysis is easy to obtain quadcopter dynamics model equation [24]:

$$m\dot{v} = F_f + F_d + F_g \tag{3.26}$$

$$J\underline{\dot{v}} + \underline{v}x\, J\underline{v} = M_f - M_d + M_c \tag{3.27}$$

In order to study the convenience and effectiveness of this model, and according to the research, the mathematical model to increase the constraints: (1) in the absence of indoor or outdoor circumstances, the aircraft hover and (2) As the rotor mass and volume are very small, its moment of inertia [J_r] is also very small, the rotor rotation generated when the Coriolis torque:

$$\text{Mc} = \sum_{i=1}^{4} \underline{v} \times J_r[0\ 0\ (-1)i + 1w_i]^T \tag{3.28}$$

Which is not obvious and neglects its influence on the angular motion of the aircraft. Therefore, the simplified model applied as:

$$m\dot{v} = F_f + F_g \tag{3.29}$$

$$J\underline{\dot{v}} + \underline{v}\, x\, J\underline{v} = M_f \tag{3.30}$$

Explain the specific meaning of the variables in this model: The translational motion of a four-rotor vehicle is achieved by the lift provided by four propellers, and since the propeller shaft of the aircraft is fixed, its lift direction is constant in the body coordinate system, i.e. perpendicular to the body. In the ground coordinate system, the horizontal force along the three-axis, can be expanded as follows [24]:

$$F_f = R\begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^{4} F_i \end{bmatrix} = \begin{bmatrix} cos\psi sin\theta cos\phi + sin\psi \sin\phi \\ sin\psi sin\theta cos\phi + cos\psi \sin\phi \\ cos\theta cos\phi \end{bmatrix} \sum_{i=1}^{4} F_i \tag{3.31}$$

The lift provided by the rotor is proportional to the product of the lift coefficient and the square of the rotor speed, Therefore, F_i is defined as the lift produced when the i^{th} rotor rotates, specifically expressed as:

$$F_i = bw_i^2 \quad i = 1,2,3, \dots \dots \dots \tag{3.32}$$

Quadcopter by the gravity of the body (select the positive direction up) can be written as:

$$F_g = [0 \quad 0 \quad -mg]^T \tag{3.33}$$

By substituting previous equations, we can get:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} cos\psi sin\theta cos\phi + sin\psi \sin\phi \\ sin\psi sin\theta cos\phi + cos\psi \sin\phi \\ cos\theta cos\phi \end{bmatrix} \left[\frac{b}{m}\sum_{i=1}^{4} w_i^2\right] - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{3.34}$$

When the rotor around the axis of symmetry in space, high-speed rotation, the axis of symmetry changes in resistance will appear when the torque, known as the gyro moment. During the flight of a four-rotor aircraft, the rotor rotates at high speed and its rotation axis changes with the position and attitude of the aircraft in space. At this time, the aircraft will be affected by its own gyro moment, which can be expressed as [24]:

$$\underline{v} x\, J\underline{v} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} x \begin{bmatrix} I_x p \\ I_y q \\ I_z r \end{bmatrix} = \begin{bmatrix} qr(I_z - I_y) \\ pr(I_x - I_z) \\ pq(I_y - I_x) \end{bmatrix} \tag{3.35}$$

Quadcopter in the air to do roll, pitch, yaw movement, will be driven by the rotational torque, resulting in changes in attitude. Specific roll, pitch, and yaw torques are expressed as:

$$M_f = \begin{bmatrix} l(F_4 - F_2) \\ l(F_3 - F_1) \\ d({w_1}^2 + {w_3}^2 - {w_2}^2 - {w_4}^2) \end{bmatrix} = \begin{bmatrix} lb({w_4}^2 - {w_2}^2) \\ lb({w_3}^2 - {w_1}^2) \\ d({w_1}^2 + {w_3}^2 - {w_2}^2 - {w_4}^2) \end{bmatrix} \quad (3.36)$$

Substituting Equations above yields the following equation:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} qr\left(\frac{I_y - I_z}{I_x}\right) \\ pr\left(\frac{I_z - I_x}{I_y}\right) \\ pq\left(\frac{I_x - I_y}{I_z}\right) \end{bmatrix} = \begin{bmatrix} \frac{l}{I_x} b({w_4}^2 - {w_2}^2) \\ \frac{l}{I_y} b({w_3}^2 - {w_1}^2) \\ \frac{d}{I_z}({w_1}^2 + {w_3}^2 - {w_2}^2 - {w_4}^2) \end{bmatrix} \quad (3.37)$$

Assumed that the control input of the four rotorcraft is U = (U1, U2, U3, U4)^T. The control input U is:

$$U_1 = b({w_1}^2 + {w_2}^2 + {w_3}^2 + {w_4}^2) \quad (3.38)$$

$$U_2 = b({w_4}^2 - {w_2}^2) \quad (3.39)$$

$$U_3 = b({w_3}^2 - {w_1}^2) \quad (3.40)$$

$$U_4 = d({w_1}^2 + {w_3}^2 - {w_2}^2 - {w_4}^2) \quad (3.41)$$

The dynamic model of a four-rotor vehicle can be expressed as follows:

$$\ddot{x} = (\, cos\psi sin\theta cos\phi + sin\psi \sin\phi)\frac{U_1}{m} \quad (3.42)$$

$$\ddot{y} = (sin\psi sin\theta cos\phi - cos\psi \sin\phi)\frac{U_1}{m} \quad (3.43)$$

$$\ddot{z} = cos\theta cos\phi \frac{U_1}{m} - g \quad (3.44)$$

$$\ddot{\phi} = \dot{\theta}\psi\left(\frac{I_y - I_z}{I_x}\right) + \frac{1}{I_x}U_2 \quad (3.45)$$

$$\ddot{\theta} = \dot{\phi}\dot{\psi}\left(\frac{I_z - I_x}{I_y}\right) + \frac{1}{I_y}U_3 \tag{3.46}$$

$$\ddot{\psi} = \dot{\phi}\dot{\theta}\left(\frac{I_x - I_y}{I_z}\right) + \frac{1}{I_z}U_4 \tag{3.47}$$

3.4.2 Kinematic analysis

The state variables of the quadrotor are the following twelve quantities in Table 3.1 [24].

Table 3.1: Different state variables and their functions

State variables	Function
$\{x, y, z\}$	Position of the center of moment of quadrotor in inertial frame (R_i)
$\{\theta, \phi, \psi\}$	Rotation Euler angles from inertial to body
$\{v_{qx}, v_{qy}, v_{qz}\}$	Linear velocities measured along each axis in body frame (R_q)
$\{\omega_{qx}, \omega_{qy}, \omega_{qz}\}$	Angular velocities measured along each axis in body frame

Assuming the rotation applied from inertial to body follow sequence 1-2-3, there is a relation between position and velocities:

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ -z \end{bmatrix} = R_q^i v_q =$$
$$\begin{bmatrix} cos\psi cos\theta & cos\psi sin\theta sin\phi - sin\psi cos\phi & cos\psi sin\theta cos\phi + sin\psi sin\phi \\ sin\psi cos\theta & sin\psi sin\theta sin\phi + cos\psi cos\phi & sin\psi sin\theta cos\phi - cos\psi sin\phi \\ -sin\theta & sin\theta & cos\theta cos\phi \end{bmatrix} \tag{3.48}$$

Now, it is needed to relate Euler angles to angular velocities, considering the angular velocities are body frame quantities, each Euler angles along different direction as following Figure 3.7.

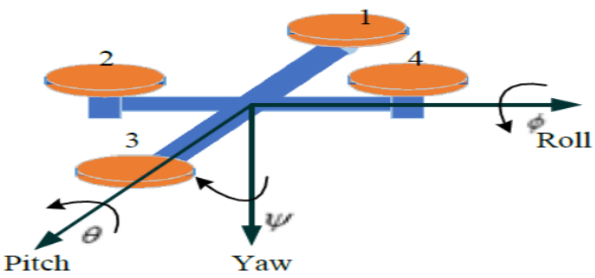


Figure 3.7: Euler angles

Set following identity matrix for relating Euler angles to angular velocities:

$$R^b_{Roll}(\phi) = R^b_{Pitch}(\theta) = R^b_{Yaw}(\psi) = \mathrm{I} \tag{3.49}$$

Then:

$$\begin{bmatrix}\omega_{qx}\\ \omega_{qy}\\ \omega_{qz}\end{bmatrix} = R^b_{Roll}(\phi)\begin{bmatrix}\phi\\0\\0\end{bmatrix} + R^b_{Roll}(\phi)R^b_{Pitch}(\theta)\begin{bmatrix}0\\ \theta\\0\end{bmatrix} + R^b_{Roll}(\phi)R^b_{Pitch}(\theta)R^b_{Yaw}(\psi)\begin{bmatrix}0\\0\\ \psi\end{bmatrix} \tag{3.50}$$

wrote the above equation in more compact form:

$$\begin{bmatrix}\omega_{qx}\\ \omega_{qy}\\ \omega_{qz}\end{bmatrix} = \begin{bmatrix}\phi\\0\\0\end{bmatrix} + R^b_{Roll}(\phi)\begin{bmatrix}0\\ \theta\\0\end{bmatrix} + R^b_{Roll}(\phi)R^b_{Pitch}(\theta)\begin{bmatrix}0\\0\\ \psi\end{bmatrix}$$

$$= \begin{bmatrix}\phi\\0\\0\end{bmatrix} + \begin{bmatrix}1 & 0 & 0\\ 0 & cos\phi & sin\phi\\ 0 & -sin\phi & cos\phi\end{bmatrix}\begin{bmatrix}0\\ \theta\\0\end{bmatrix}$$

$$+ \begin{bmatrix}1 & 0 & 0\\ 0 & cos\phi & sin\phi\\ 0 & -sin\phi & cos\phi\end{bmatrix}\begin{bmatrix}cos\theta & 0 & sin\theta\\ 0 & 1 & 0\\ sin\theta & 0 & cos\theta\end{bmatrix}\begin{bmatrix}0\\0\\ \psi\end{bmatrix}$$

$$= \begin{bmatrix}1 & 0 & -sin\theta\\ 0 & cos\phi & sin\phi cos\theta\\ 0 & -sin\theta & cos\phi cos\theta\end{bmatrix}\begin{bmatrix}\phi\\ \theta\\ \psi\end{bmatrix} \tag{3.51}$$

Inverted the coefficient transformation matrix, we get:

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} 1 & sin\phi tan\theta & cos\phi tan\theta \\ 0 & cos\phi & -sin\theta \\ 0 & sin\phi sec\theta & cos\phi sec\theta \end{bmatrix} \begin{bmatrix} \omega_{qx} \\ \omega_{qy} \\ \omega_{qz} \end{bmatrix} \tag{3.52}$$

3.4.3 Commands

There are four inputs to the quadcopter system, which are the inputs to the four rotors. In order to cause the quadcopter to roll, pitch and yaw the current system inputs need to be mapped to a roll, pitch and yaw as well a height input. To obtain the desired input to the four rotors, the following set of equations were created [25]:

$$\begin{bmatrix} Throttle1_{desired} \\ Throttle2_{desired} \\ Throttle3_{desired} \\ Throttle4_{desired} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & -1 & 1 \\ 1 & 1 & 0 & -1 \\ 1 & -1 & 0 & -1 \end{bmatrix} \begin{bmatrix} height_{input} \\ roll_{input} \\ pitch_{input} \\ yaw_{input} \end{bmatrix} \tag{3.53}$$

The four inputs of the system will now become roll, pitch, yaw and height. These four inputs will control the desired throttle for rotor 1 to 4.

3.5 Equations of Motion

In the inertial frame, the acceleration of the quadcopter is due to thrust, gravity, and linear friction. The thrust vector obtained in the inertial frame by using our rotation matrix *R* to map the thrust vector from the body frame to the inertial frame. Thus, the linear motion can be summarized as: [24]

$$m\ddot{x} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + RTB + F_D \tag{3.54}$$

While it is convenient to have the linear equations of motion in the inertial frame, the rotational equations of motion are useful to us in the body frame, so that the express rotations about the center of the quadcopter instead of about our inertial center. The rotational equations of motion derived from Euler’s equations for rigid body dynamics. Expressed in vector form, Euler’s equations are written as:

$$M = I\dot{\underline{\mathrm{v}}} + I\underline{\mathrm{v}}^2 \tag{3.55}$$

this was writing as:

$$\dot{\underline{\mathrm{v}}} = \frac{\tau - I\underline{\mathrm{v}}^2}{M} \tag{3.56}$$

The quadcopter can be modeled as two thin uniform rods crossed at the origin with a point mass (motor) at the end of each. With this in mind, it's clear that the symmetries result in a diagonal inertia matrix of the form [24]:

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{3.57}$$

Therefore, the final result for the body frame rotational equations of motion is obtained as:

$$\dot{\underline{\mathrm{v}}} = \begin{bmatrix} M\varphi I_{xx}^{-1} \\ M\varphi I_{yy}^{-1} \\ M\varphi I_{zz}^{-1} \end{bmatrix} - \begin{bmatrix} \frac{I_{yy} - I_{zz}}{I_{xx}} \underline{\mathrm{v}}_y \underline{\mathrm{v}}_z \\ \frac{I_{zz} - I_{xx}}{I_{yy}} \underline{\mathrm{v}}_z \underline{\mathrm{v}}_x \\ \frac{I_{xx} - I_{yy}}{I_{zz}} \underline{\mathrm{v}}_x \underline{\mathrm{v}}_y \end{bmatrix} \tag{3.58}$$

3.6 Quadcopter Control

The control algorithm receives, as inputs, the data from the sensors and from the task. During the computation it uses a lot of constants and variables which describe the dynamics and the quadrotor states. The output of the algorithm is the code which determine the PWM signal of the four motors. The controller can be divided in four components according to Figure 3.8 [26].

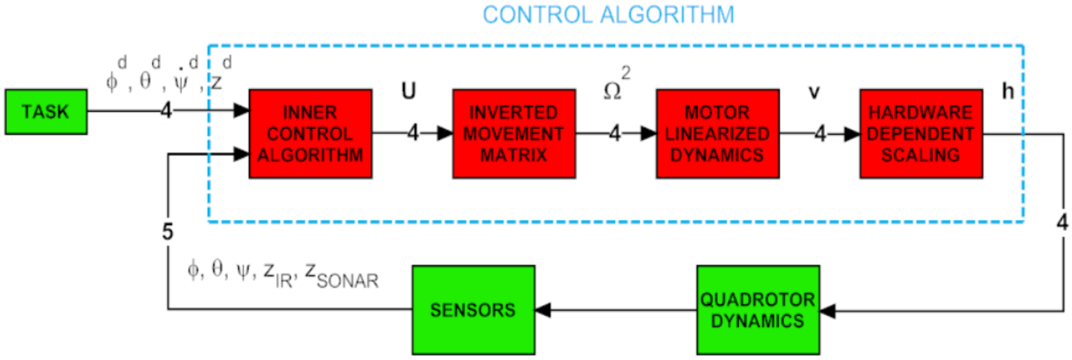


Figure 3.8: Control modeling schematic of a quadcopter

- Inner control algorithms: represents the core of the control algorithms. It processes the task and the sensors data and provides a signal for each basic movement which balances the position error.

- Inverted movement matrix: is the second block in the control chain. It is used to compute the propellers' squared speed from the four basic movement signals.

- Motor linearized dynamics: is the third block in the control chain. Several approaches can be followed in this block. The first one is simply to solve numerically the equation to obtain the right voltage values. The second one is to linearize the equation to reduce the computation (in comparison with the previous option) and the third method is to characterize experimentally the motor behavior and use the deduced relation.

- Hardware dependent scaling: is the last block in the control chain. It processes the inputs voltage vector (v) and provide the PWM code vector (h) to the hardware. The control algorithm before this block doesn't take into account the battery voltage. This variable is instead very important to determine the right commands. Therefore, a scaling must be added to consider the voltage. Furthermore, this block is used to avoid distortion in case of saturation. This means that if the required voltage is above the battery one, the algorithms gives priority to the attitude over the height. Without this operation an undesired behavior can occur [27].

3.6.1 PID control

Stabilization is very important for an under-actuated system like a quadrotor, as it is inherently unstable due to its six degrees of freedom and four actuators. A control system is modeled for the quadcopter using four PID controllers to control the Attitude (roll, pitch and yaw) and the Altitude (Z height) to introduce stability these four controllers form the inner loop of control for the quadcopter system. And then two more PID controllers are used to control the position of the quadcopter (X and Y axes) and the output of these two controllers will be input to the roll and pitch controllers these two PID's will form the outer control loop [26]. The traditional PID structure is composed of the addition of three contributes, as shown in Figure 3.9.

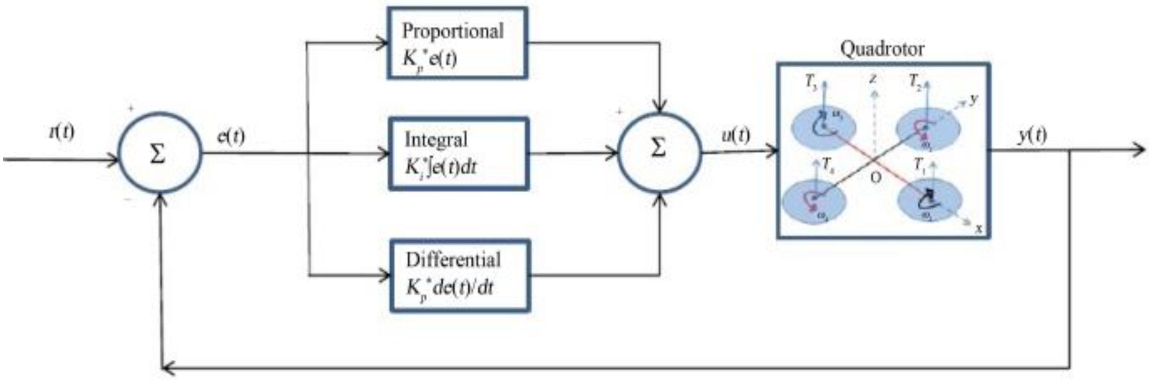


Figure 3.9: Schematic of PID controller application to a quadcopter system

The expression for the control variable [u] may be written as follows: -

$$u(t) = K_P e(t) + K_I \int_0^t e(t)\, dt + K_D \frac{de(t)}{dt} \tag{3.59}$$

Inner control loop controls quadcopter altitude and attitude. Input variables for inner loop can be divided in two parts, desired and sensor signals. Desired signals are obtained from the control signals coming directly form the pilot or the autopilot program. These signals are the Height (altitude) and pointing (yaw) of the quadcopter the other two signals desired roll and pitch comes for the output of the outer loop control since they are translated form the desired x and y position in the outer PID's [27].

(1) Altitude control

Figure 3.10 shows the block diagram of altitude control.

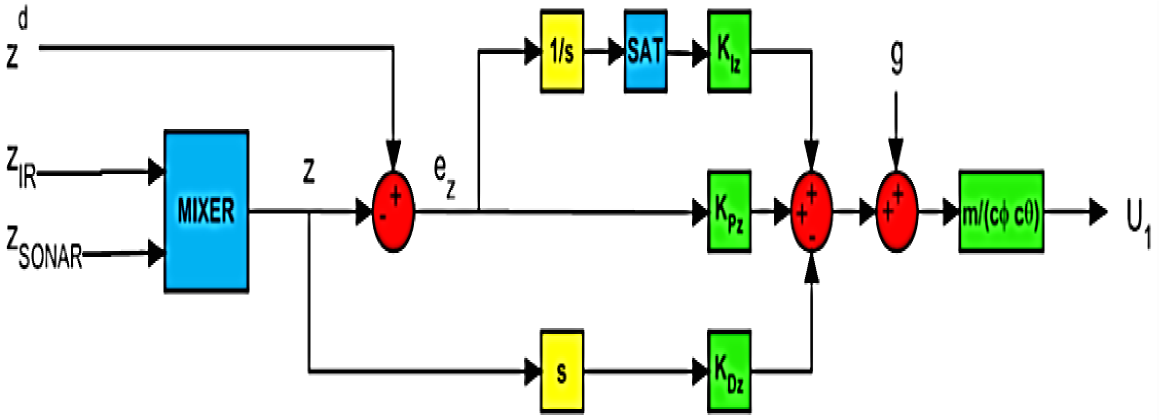


Figure 3.10: Block diagram of altitude control

Equation for the thrust force control variable is:

$$U_1 = K_{Pz}e_z + K_{Iz}\int e_z - K_{Dz}\frac{d}{dt}e_z \tag{3.60}$$

(2) Roll control

Figure 3.11 shows the block diagram of roll control.

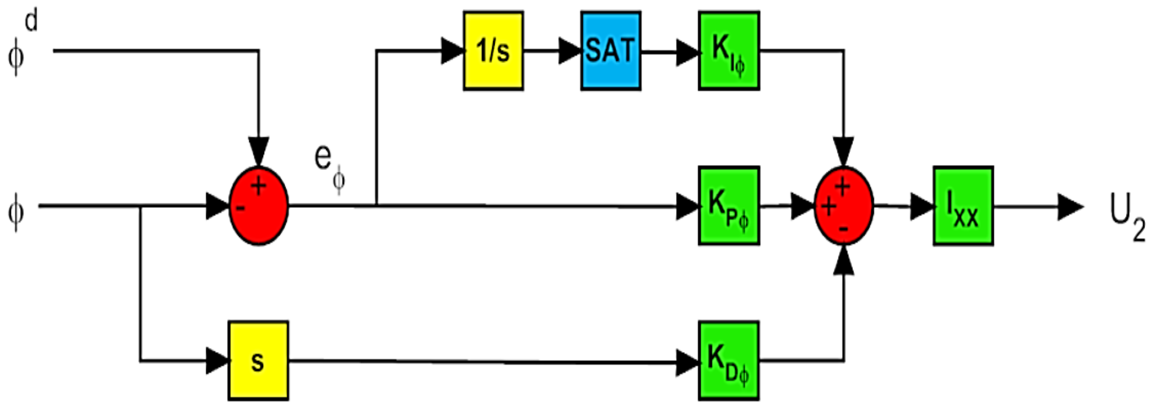


Figure 3.11: Block diagram of roll control

Equation for the roll moment control variable is:

$$U_2 = K_{P\phi} e_\phi + K_{I\phi} \int e_\phi - K_{D\phi} \frac{d}{dt} e_\phi \tag{3.61}$$

(3) Pitch control

Figure 3.12 shows the block diagram of pitch control.

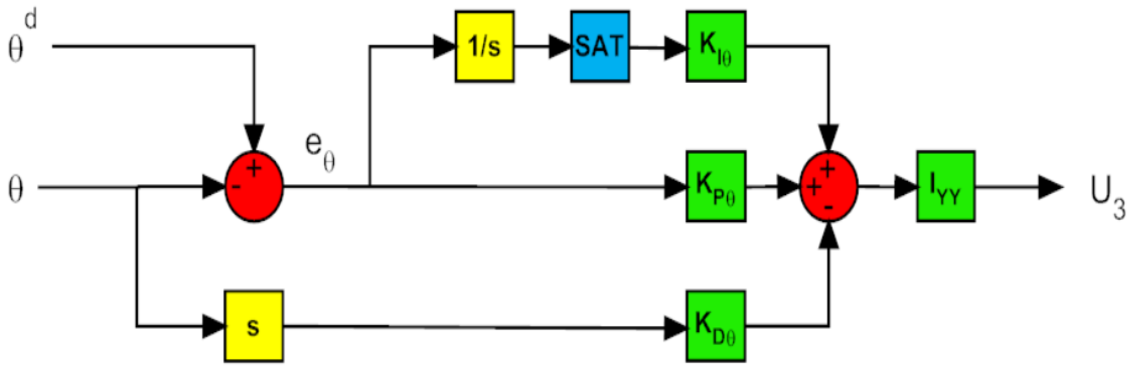


Figure 3.12: Block diagram of pitch control

Equation for the pitch moment control variable is:

$$U_3 = K_{P\theta} e_\theta + K_{I\theta} \int e_\theta - K_{D\theta} \frac{d}{dt} e_\theta \tag{3.62}$$

(4) Yaw control

Figure 3.13 shows the block diagram of yaw control.

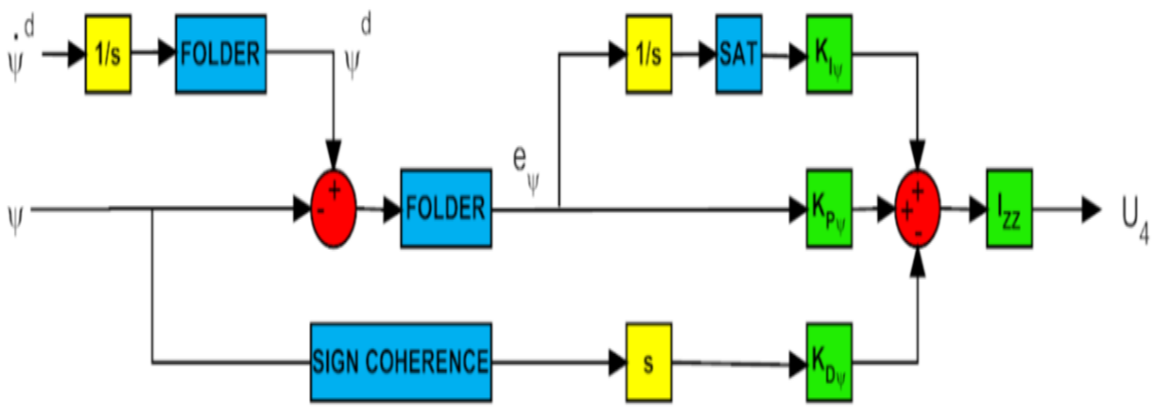


Figure 3.13: Block diagram of yaw control

Equation for the yaw moment control variable is:

$$U_4 = K_{P\psi} e_\psi + K_{I\psi} \int e_\psi - K_{D\psi} \frac{d}{dt} e_\psi \tag{3.63}$$

Outer control loop is applied since the quadcopter is under-actuated system and it is not applicable to control all of the quadcopter 6-DOF straightly. As mentioned earlier, the inner loop directly controls 4-DOF (three angles and altitude). To be able to control X and Y position, outer loop is implemented. The outer control loop outputs are desired roll and pitch angles, which they are the inputs to the inner loop for the desired X and Y position [27]. Equations for the quadcopter X and Y linear accelerations:

$$\ddot{X} = (\ cos\psi sin\theta cos\phi + sin\psi \sin\phi) \frac{U_1}{m} \tag{3.64}$$

$$\ddot{Y} = (sin\psi sin\theta cos\phi - cos\psi \sin\phi) \frac{U_1}{m} \tag{3.65}$$

Quadcopter dynamics of the X and Y linear accelerations can be simplified in to:

$$\ddot{X} = (\ \theta cos\psi + \phi sin\psi) \frac{U_1}{m} \tag{3.66}$$

$$\ddot{Y} = (\theta sin\psi - \phi cos\psi) \frac{U_1}{m} \tag{3.67}$$

3.6.2 Fuzzy logic control

The conventional PID controller has several limitations. First, their fixed gains limit system performance over a wider operational range. When the required range of operation is large, the conventional PID controller is prone instability, because the nonlinearities in the system cannot be properly dealt with. Second, as conventional PIDs are based on a linear model, their performance may suffer in a nonlinear system like a quadcopter. Several studies have attempted to overcome these shortcomings like fuzzy logic control, adaptive fuzzy logic control, evolutionary algorithm and many others.

The fuzzy logic control of a quadcopter can be represented by the block diagram in Figure 3.14 [28].

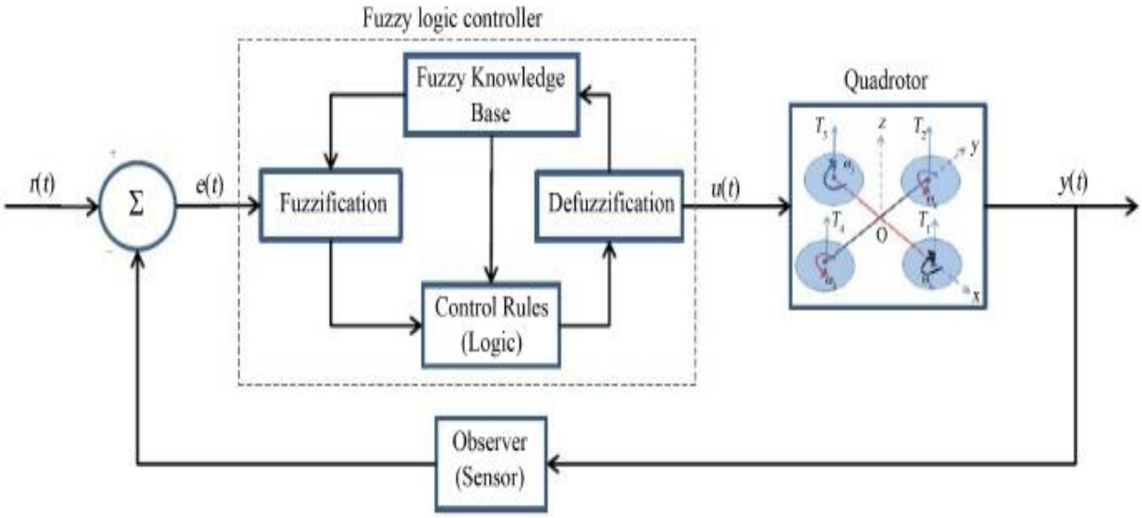


Figure 3.14: Schematic of fuzzy logic application to a quadcopter system

For quadrotor control four different controllers for roll, pitch, yaw and height are developed. Main objective is to design a controller which is least complex.

(1) Fuzzy altitude control

The equation for control input is same as described in the Equation (3.60) with an additional saturation blocks are used in the design of the controller to avoid the any situation that can take value out of the defined range.

(2) Fuzzy roll control

The equation for control input is same as described in the Equation (3.61) with an additional saturation blocks are used in the design of the controller to avoid the any situation that can take value out of the defined range.

(3) Fuzzy pitch control

The equation for control input is same as described in the Equation (3.62) with an additional saturation blocks are used in the design of the controller to avoid the any situation that can take value out of the defined range.

(4) Fuzzy yaw Control

The equation for control input is same as described in the Equation (3.63) with an additional saturation blocks are used in the design of the controller to avoid the any situation that can take value out of the defined range.

3.6.3 Adaptive fuzzy logic control

Adaptive PID controller based on parameter optimization with fuzzy inference means that the three gains K_p, K_i and K_d of PID controller are adjusted online by using fuzzy logic control. Online tuning gains of PID controller lead to enhance the adaptive performance of PID controller [29]. The structure of the proposed adaptive PID controller is as in Figure 3.15.

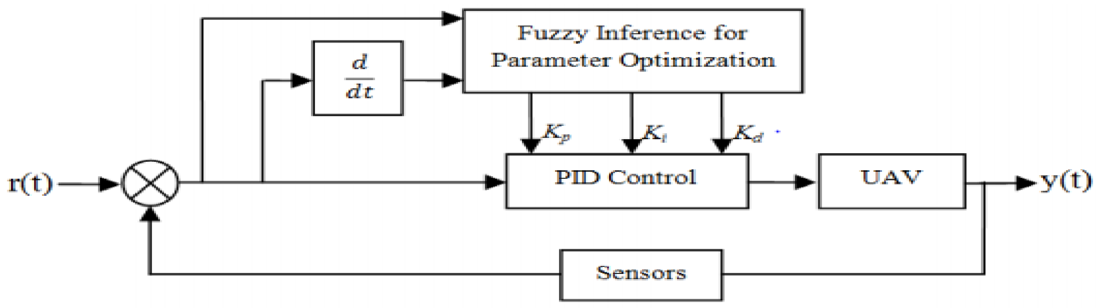


Figure 3.15: Schematic of adaptive fuzzy PID controller application to a quadcopter system

The improved PID controller is designed based on the traditional PID algorithm. The initial setting values of PID gains are set and the added values are obtained online using fuzzy control. The final gains of the adaptive PID controller can be calculated from:

$$\begin{aligned} K_P &= K_{Po} + \Delta K_P \\ K_I &= K_{Io} + \Delta K_I \\ K_D &= K_{Do} + \Delta K_D \end{aligned} \tag{3.68}$$

(1) Adaptive fuzzy altitude control

The control input is same as described in the Equation (3.60) with the values of proportionality constants for PID controller as described in Equation (3.68).

(2) Adaptive fuzzy roll control

The roll's control input is same as described in the equation 3.61 with the values of proportionality constants for PID controller as described in equation 3.68.

(3) Adaptive fuzzy pitch control

The control input is same as described in the equation 3.62 with the values of proportionality constants for PID controller as described in equation 3.68.

(4) Adaptive fuzzy yaw control

The control input is same as described in the equation 3.63 with the values of proportionality constants for PID controller as described in equation 3.68.

3.7 Practical Model of a Quadcopter System

Divides into two parts mechanical and electrical parts each of them involves a number of components as follows.

3.7.1 Mechanical parts of a quadcopter

(i) Frame

The airframe is the main body of the aircraft where all other components such as propellers, batteries computer, etc. are mounted to the airframe. Airframes can vary greatly in size and complexity. An airframe is like a suit of armor for all of the sensitive electrical components that constitute a quadcopter. It is essential that a frame is as durable and rugged as possible, while still accommodating to the needs of the pilot without hindering the flying experience and the inevitable maintenance that will ensue. Many different types of materials are used to build the modern-day drone, some of the more popular options include carbon fiber, fiberglass, and various types of plastic or metal. The frame that was chosen for doing the project was "F450 Quadcopter Frame" which is a composite material frame (built from quality glass fiber and polyamide nylon) as shown in Figure 3.16.

Figure 3.16: F450 quadcopter frame

The specifications of this frame are as follows:

- Width: 450mm.
- Height: 55mm.
- Weight: 280g (w/out electronics).
- Net weight: 272g.

(ii) Propellers

A quadcopter propeller is the key component that keeps your multirotor in the air, they have the most direct impact on how your machine flies, and over the lifetime of your multirotor they are likely to add up to be the biggest investment that you make to keep flying. Propellers are commonly described using a series of numbers separated by an "x" that relate to size, pitch, and blade configuration. For instance, you may see a propeller being described as a 5×4.3×3, the first number corresponds to the Size, 5 inches in this case. The second to the pitch, 4.3 inches in this case and the third to the number of blades, which in this case would be a 3 bladed propeller. The used in this project was"1045" propellers" because they are recommended by manufacturers of the frame as shown in Figure 3.17.

Figure 3.17: 1045" propellers

3.7.2 Electrical parts of a quadcopter

(i) Brushless DC motor (BLDC)

The selection of an electric Drone Motor has an enormous influence on the flight characteristics of the multi-copter. Tiny variations in the construction of a motor result in significant impacts regarding the weight, responsiveness and total power of the multi-copter. A brushless DC motor (known as BLDC) is a permanent magnet synchronous electric motor which is driven by direct current (DC) electricity and it accomplishes electronically controlled commutation system (commutation is the process of producing rotational torque in the motor by changing phase currents through it at appropriate times) instead of a mechanically commutation system. In this work the used was "1000 KV BLDC" which is shown in Figure 3.18.

Figure 3.18: 1000KV BLDC motor

The specifications of this motor are as follows:

- Kv: 1000 RPM/V.
- Max Efficiency: 80%.
- No Load Current: 0.5A at 10V.
- Max Watts: 150W.

(ii) Electronic speed controller (ESC)

The speed controller acts as the interface between the power circuit (connected to the battery) and the control circuit (signals coming from the ardupilot mega (APM) board). It transforms the pulse width modulation (PWM) low voltage signal from the onboard computer into a signal powered by the battery that the brushless motor can understand and be powered by. Here the used was "30A ESC" as shown in Figure 3.19.

Figure 3.19: 30A Electronic speed controller

The specifications of this ESC are as follows:

- Cont. Current:30A.
- Burst Current: 55A.
- BEC Mode: Linear.

(iii) Battery

A Drone battery is the foundational component of a quadcopter and must be considerately selected to achieve an ideal balance between performance and flight time. Lithium batteries are the most common battery chemistry used to power quadcopters due to their high energy densities and high discharge capabilities. The lithium battery packs used to power quadcopters have two common chemistries: Lithium polymer (LiPO) and lithium polymer high voltage (LiHV). The primary difference between the two is that a LiPO cell has a fully charged voltage of 4.2V compared to a LiHV cell which has a voltage of 4.35V at full charge. LiPO packs are the most commonly used across all sizes of quadcopters. Battery voltage is the potential energy difference between the positive and negative terminals. A higher Drone battery voltage allows the pack to provide more power to the quadcopter without increasing the current or amp draw. LiPO packs are commonly sold in 1S, 2S, 3S, 4S, 5S or 6S configurations where the digit followed by the 'S' stands for number of cells in that specific pack. The used battery was "4000mAh battery" as shown below in Figure 3.20.

Figure 3.20: 4000mAh LiPO battery

The specification of this battery are as follows:

- Minimum Capacity: 4000mAh.
- Configuration: 3S1P / 11.1v / 3Cell.
- Constant Discharge: 30C.
- Peak Discharge (10sec): 40C.

(iv) Transmitter and receiver

One of the crucial pieces of equipment necessary for flying a multirotor is a Drone Radio Transmitter (Tx) and Radio Receiver (Rx).

(A) Quadcopter radio transmitters

An FPV quadcopter radio transmitter is an electronic device that uses radio signals to transmit commands wirelessly via a set radio frequency over to the Radio Receiver, which is connected to an aircraft or multirotor being remotely controlled.

(B) Quadcopter radio receivers

A radio receiver is the device capable of receiving commands from the radio transmitter, interpreting the signal via the flight controller where those commands are converted into specific actions controlling the aircraft. The used was "Fly sky (FS)transmitter and receiver" as shown in Figure 3.21.

Figure 3.21: Fly sky (FS) transmitter and receiver

The specifications are:

- Item: FS-i6X RC Transmitter.
- Channel: 6-10 (Default 6).
- RF Range: 2.408-2.475GHz.
- RF Power: < 20dBm.
- Bandwidth: 500KHz.

(v) Ardupilot mega flight controller

Ardupilot mega (APM) is a professional quality IMU autopilot that is based on the Arduino Mega platform. This autopilot can control fixed-wing aircraft, multi-rotor helicopters, as well as traditional helicopters. The used was "Ardupilot mega v2.8" as shown in Figure 3.22.

Figure 3.22: Ardupilot mega v2.8

3.8 Other Essential Components

There are many other components that are needed to do functions like:

3.8.1 Raspberry Pi

Raspberry Pi is the name of a series of single-board computers made by the Raspberry Pi foundation, a UK charity that aims to educate people in computing and create easier access to computing education. The Raspberry Pi is a very cheap computer that runs Linux, but it also provides a set of GPIO (general purpose input/output) pins that allow you to control electronic components for physical computing and explore the Internet of Things (IoT). The Raspberry Pi operates in the open source ecosystem: it runs Linux (a variety of distributions), and its main supported operating system, Raspbian, is open source and runs a suite of open source software. The Raspberry Pi Foundation contributes to the Linux kernel and various other open source projects as well as releasing much of its own software as open source. The raspberry pi used was " Raspberry Pi 4 Model B – 2GB RAM" as shown in Figure 3.23.

Figure 3.23: Raspberry Pi 4 model B

The specifications for this raspberry pi are as follows:

- Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC at 1.5GHz, 1GB RAM (LPDDR2 SDRAM).
- 5V DC via USB-C connector (minimum 3A).
- 2 GB memory

3.8.2 Raspberry Pi camera

The Raspberry Pi High Definition (HD) Camera Board connects to any Raspberry Pi or Compute Module, allowing you to create HD video and still photographs. It utilizes the IMX219PQ image sensor from Sony which offers high-speed video imaging and high sensitivity. The Raspberry Pi Camera Module v2 is a high quality 8-megapixel Sony IMX219 image sensor custom designed add-on board for Raspberry Pi, featuring a fixed focus lens. The used here was " V2.1 Original 8MP Raspberry Pi 3 Camera module 8-megapixel IMX219PQ sensor 8MP " as shown in Figure 3.24.

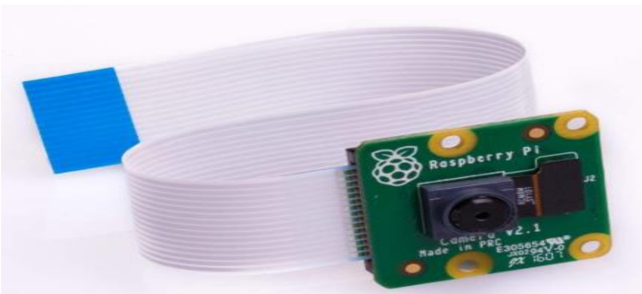


Figure 3.24: 8MP raspberry pi camera

The specifications of this camera are as follows:

- High-quality imagery.
- High data capability.
- 8 megapixels fixed focus.
- Supports 1080p, 720p60 & VGA90.

3.9 Methodology

Includes hardware and software implementations as well as operation of the system.

3.9.1 Hardware implementation

The hardware part consists of assemble the mechanical and electrical parts together to get the overall structure of the system as well as attaching raspberry pi and its camera or a mobile phone as shown below in Figure 3.25.

Figure 3.25: Overall model

3.9.2 Software implementation

This section contains the calibration for the quadcopter's components, prepare mobile phone and raspberry pi for live video streams to detect or classify objects and things, preparing used datasets and machine learning - computer vision-algorithms and networks.

(1) Calibration

Different components for the quadcopter were calibrated and tested using Mission Planner (MP) program, which is the officially program to program ardupilot-represents the neural system for the quadcopter that command all components to function- and calibrate and test different components as Figure 3.26 below illustrates motors calibration.

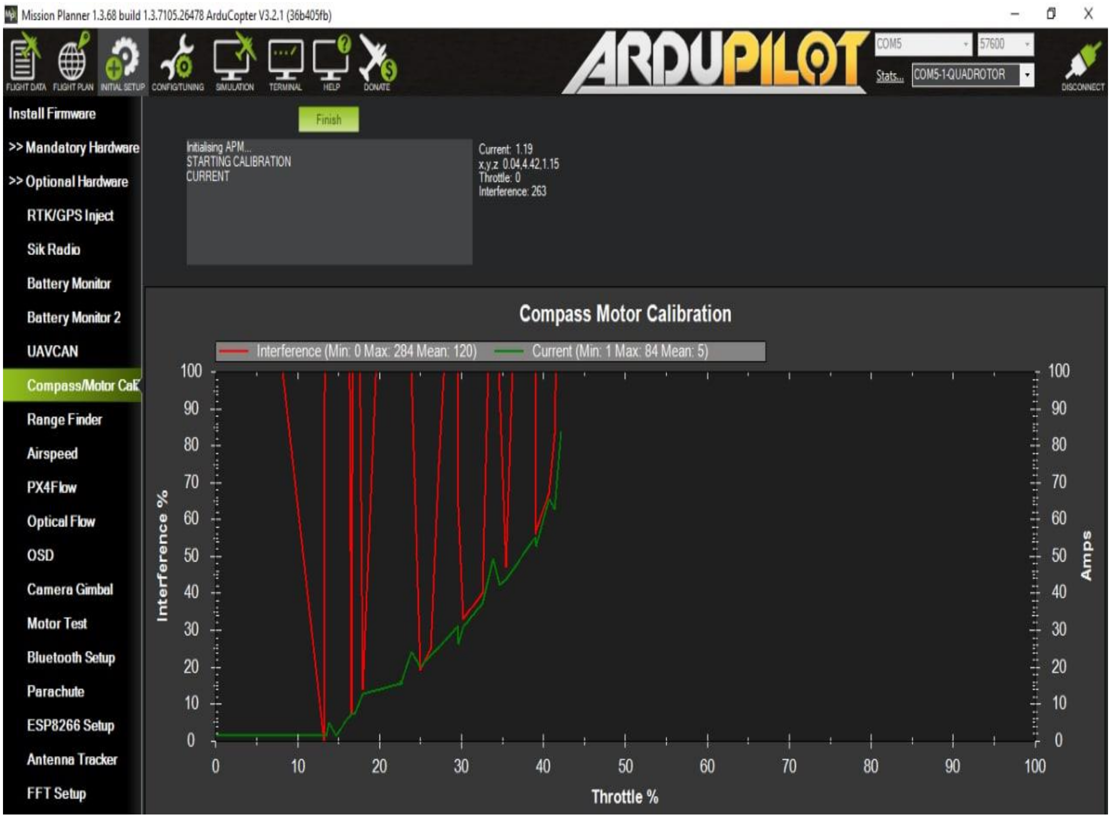


Figure 3.26: Motors calibration

The main problem of quadcopters is the stabilization in position problem - i.e. controlling or tuning yaw, pitch, roll and altitude via tuning PD controller associated with them- because it has 6 degrees of freedom (6DOF), nonlinear behavior, and noises and disturbances, which form a real challenge to tune and control it perfectly but we try to control it as best as possible to achieve superior results. Figure 3.27 shows the flow chart for the operation of a quadcopter after being assembled and calibrated.

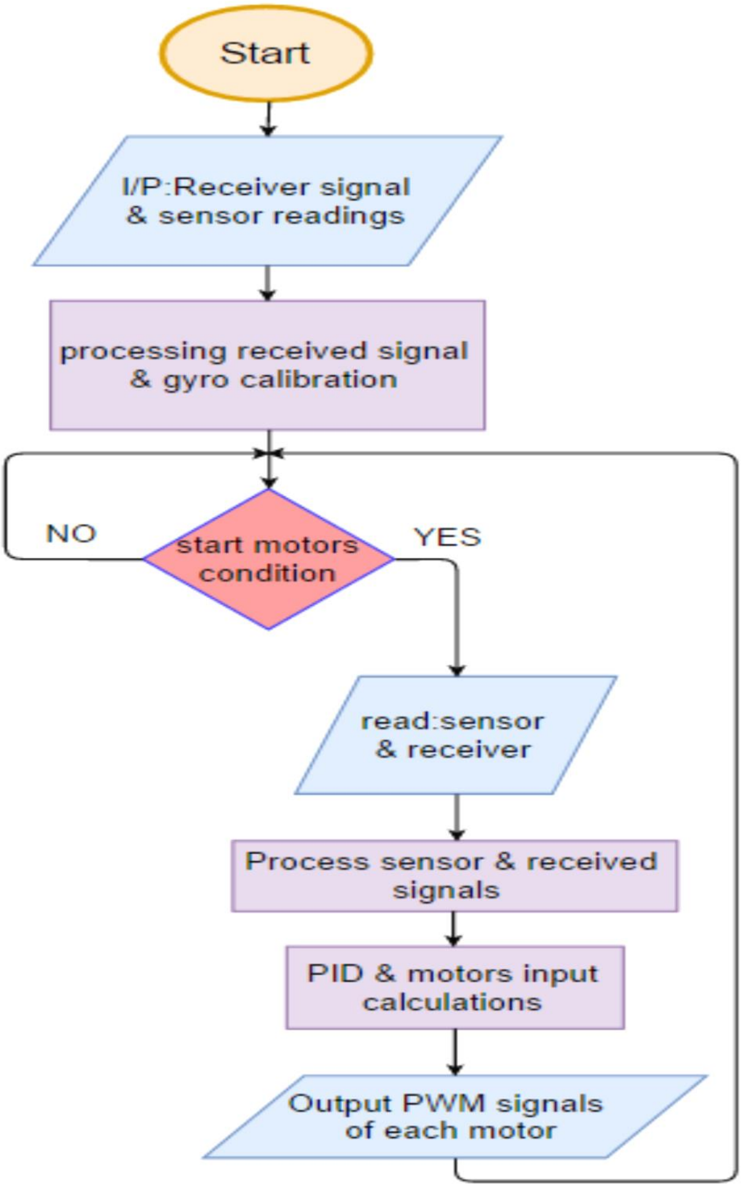


Figure 3.27: Quadcopter operation flowchart

(2) Prepare mobile phone and raspberry pi for vision tasks

The mobile phone was prepared by using IoT technology via connected it to web and Laptop -and hence the quadcopter- by using an application called IP webcam which is an application being used recently -for quadcopters field- to integrate quadcopters for vision tasks and achieve excellent results as shown in Figure 3.28.

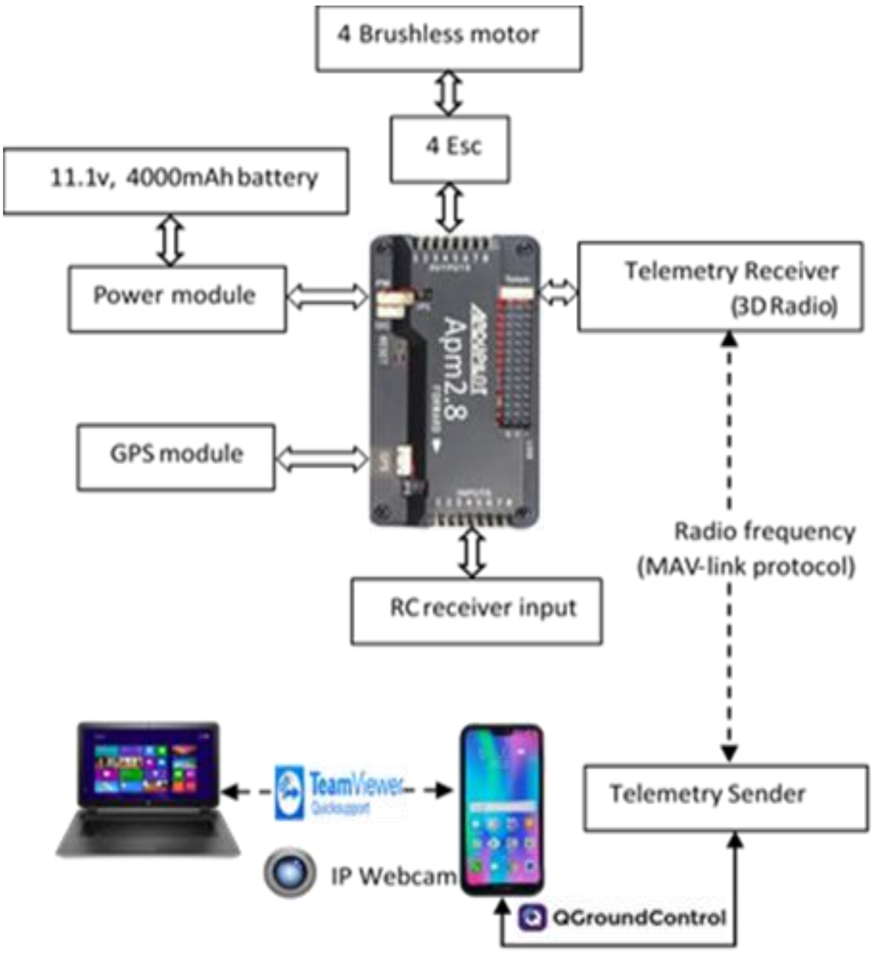


Figure 3.28: Methodology for preparing and connecting mobile phone

Applications such as TeamViewer which was used to control a smartphone remotely using a PC, and Qground control app which was used as an alternation to mission planner and for path tracking, both applications can be used to give additional and better functionalities to the system. The raspberry pi is prepared along with its cameral by programming them though codes and connections.

(3) Preparing datasets

The datasets used for detecting objects, landmines, and fires, or classifying people as wearing masks or not -used in this COVID-19 period- can be summarized in Table 3.2.

Table 3.2: Datasets with descriptions

Dataset name	Number of images	Description	Designed/Implemented
Landmine dataset	2000	A designed dataset for this work	Designed and implemented
COCO dataset (Common Objects in Context)	330,000	Common dataset used in computer vision tasks with 1.5 object instances and 80 classes	Implemented
Fire dataset	1,315	Custom dataset to classify fire and smoke	Implemented
Mask/No mask dataset	1,376	A dataset with two classes: With mask: 690 images, and without a mask: 686 images.	Implemented

(4) Preparing computer vision networks and algorithms

Some state of arts neural networks and algorithms were used here to perform different tasks related to computer vision which is a hot area of research as an important part of machine learning. Table 3.3 summarizes networks and algorithms along with descriptions and uses in this work.

Table 3.3: Neural networks and algorithms used in this work

Neural network/Algorithm	Description	Application in this work
YOLOv3	One of the states of arts object detection networks, which is continuously under development.	Used for detecting 80 types of objects include people, animals, vehicles, etc.
Mobilnetv2	With MobileNetV2 as backbone for feature extraction, state-of-the-art performances are also achieved for object detection.	Used for classifying people as wearing masks or not.

FireDetectionNet	Designed class of CNN used for building fire/smoke classifier	Build fire/smoke classifier
ACF detector	The ACF detector is a fast and effective sliding window detector (30 fps on a single core).	Used for landmines detection through MATLAB program

3.9.3 Operation of the system

The quadcopter's components were calibrated and tested, the raspberry pi was first attached to the quadcopter along with its camera, the algorithms codes were opened and connected to raspberry pi, then the complete system was ready for operation and the quadcopter was taken off, after that the various tasks were carried and the quadcopter was landed. In the next time, the mobile phone was attached to quadcopter instead of raspberry pi, be prepared for live video streams and connected to Laptop via IP webcam application and IoT, after that the same process of connecting codes, take off quadcopter and do various tasks, then landed was repeated.

When the quadcopter was taken off and flight in both cases, the response to control sticks was observed, then the PID controller is tuned according to trial and error approach and intelligent control approaches, after each tune the quadcopter was tested for control and finally, it is observed that adaptive fuzzy control gives the best control and so work with it for different tasks in this work.

CHAPTER FOUR

SIMULATION AND RESULTS

CHAPTER FOUR

SIMULATION AND RESULTS

4.1 Quadcopter Simulation and Results

Consists of simulating the quadcopter in SIMULINK as well as stabilize it using different controllers along with their results and discussions. Also, a simulation of autonomous flying was carried out.

4.1.1 Quadcopter simulation model

The quadcopter system itself is composed of three subsystems to meet the objective of simulation process, where the position is represented by two subsystems which are the attitude subsystem, consists of roll, pitch and yaw equations, and the altitude subsystem to control height (z). While the response to external input represented by angular velocity subsystem. The overall simulation model of the quadcopter is shown in Figure 4.1.

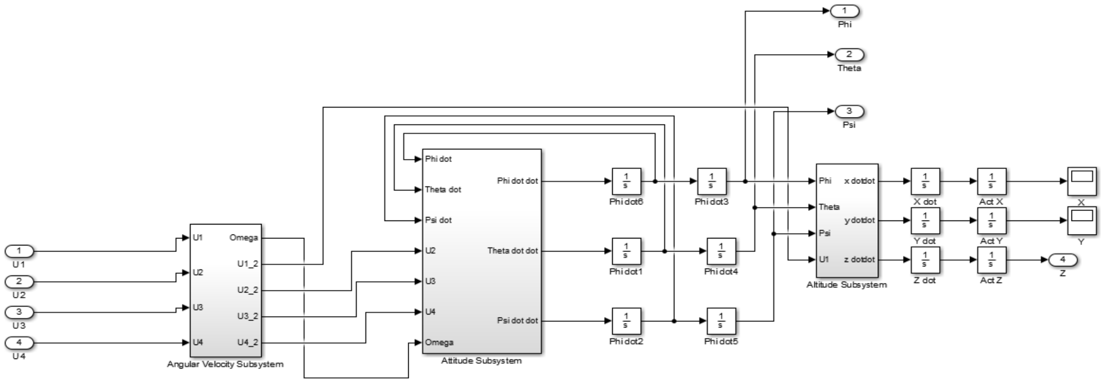


Figure 4.1: Quadcopter Simulink model

4.1.2 Simulation and results of the controllers

This section aims at controlling the above three subsystems in order to get the best performance by taking into account the presence of the 6 DOF, nonlinearity, noises and disturbances of the quadcopter which makes it difficult to control perfectly. Here three types of controllers were designed, implemented and studied to achieve best control that can be achieved which were a PD controller (the most common type of PID controller in position situations), fuzzy logic controller and adaptive fuzzy-PD controller.

(1) PD controller

Simulink models for Equations (3.60) through (3.63) were implemented, and responses were recorded as shown in Figure 4.2.

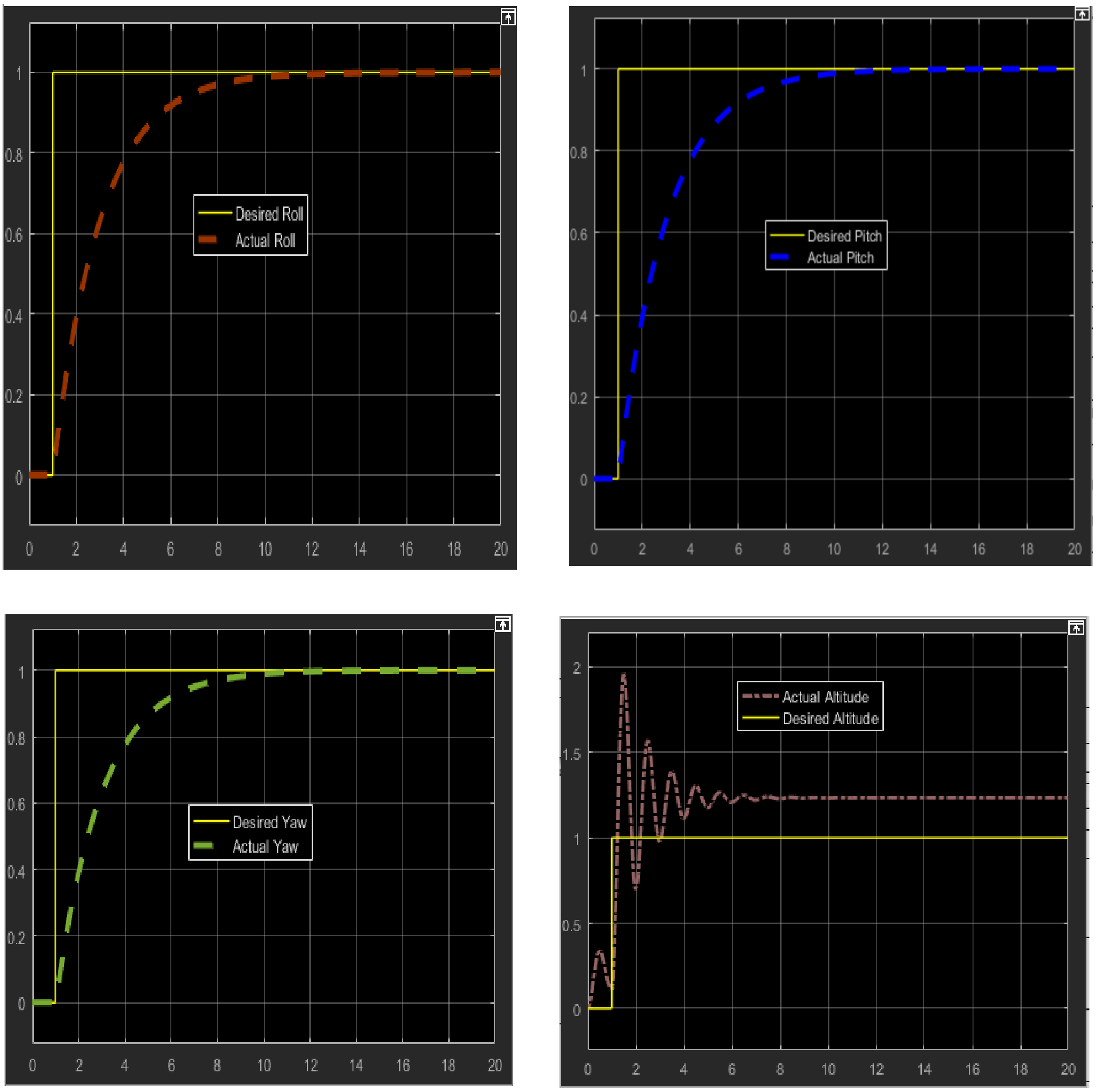


Figure 4.2: Overall responses with PD controller

Table 4.1 shows the results of applying PD controller to the quadcopter.

Table 4.1: Results of PD controller

Controller	Settling time (seconds)	Amplitude	Overshoot (%)
Roll	12	1	0
Pitch	12	1	0
Yaw	12	1	0
Altitude	10	1.2	20

By looking at responses and table above, it is observed that there is no steady state error and overshoot in roll, pitch, and yaw. Also, the settling time is high for all the controllers and there is an overshoot in altitude controller.

(2) Fuzzy logic controller

The rule base for conduction this control is shown in Table 4.2 below.

Table 4.2: Rules for a fuzzy logic controller

Position					
E \ $\dot{E}$	NB	N	Z	P	PB
N	GDM	GD	GD	S	GU
Z	GUM	GD	S	GU	GUM
P	GD	S	GU	GUM	GUM

The member ship functions are shown in Figures 4.3 through 4.5.

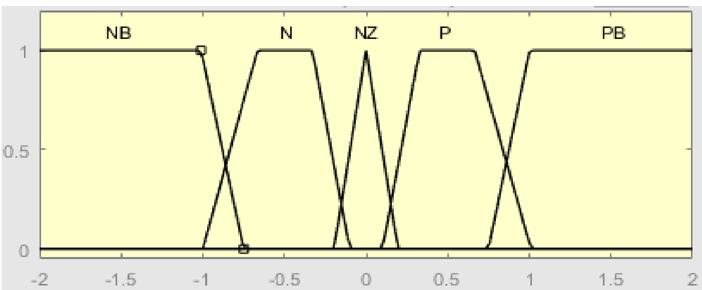


Figure 4.3: Error membership function

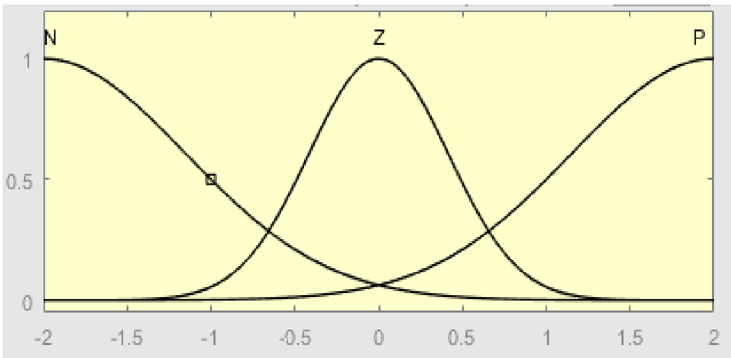


Figure 4.4: Error dot membership function

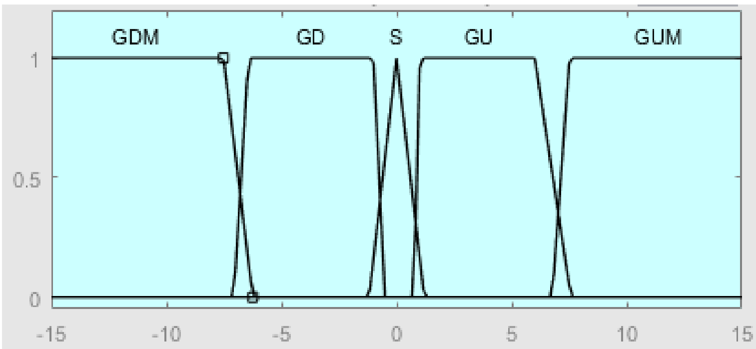
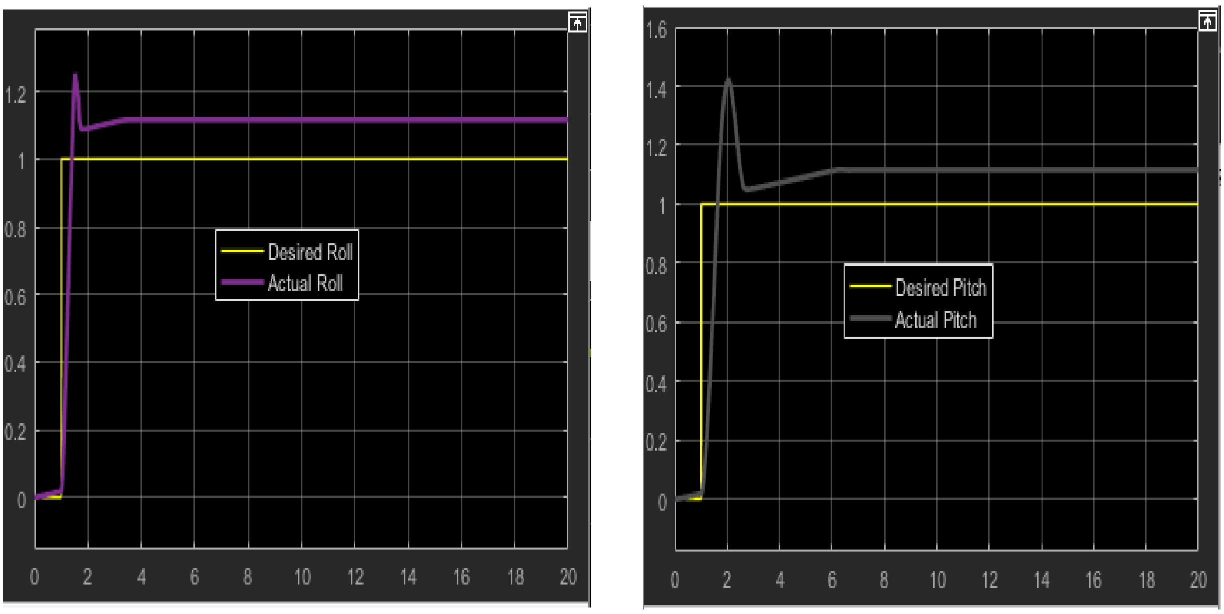


Figure 4.5: Position membership function

Responses of the quadcopter system with fuzzy controller are shown in Figure 4.6.



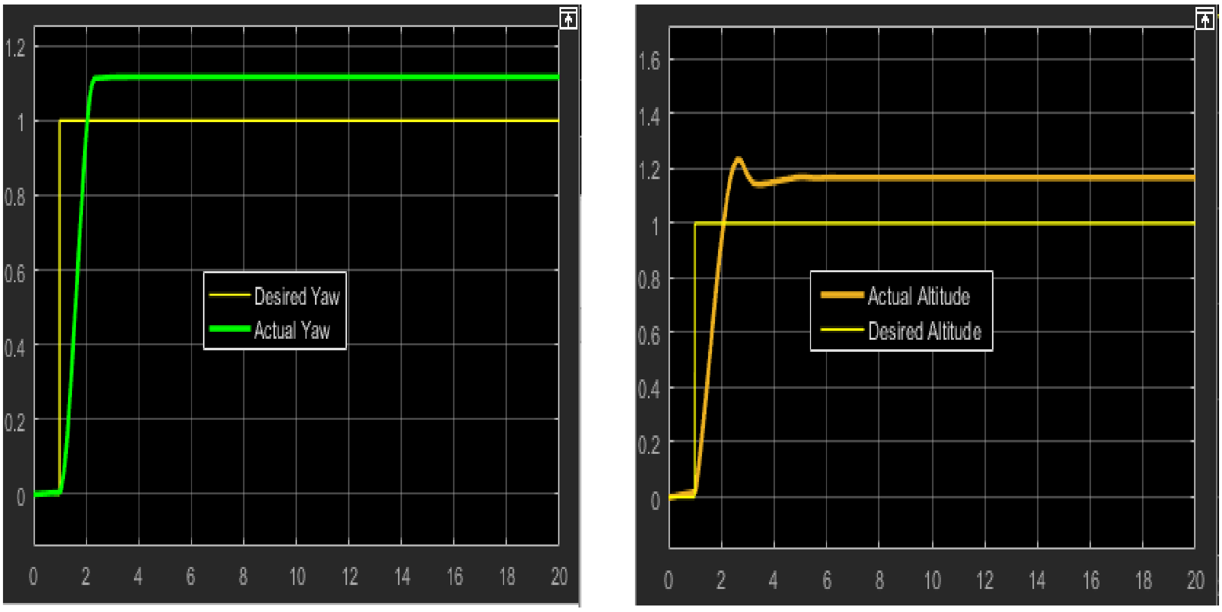


Figure 4.6: Overall responses with fuzzy logic controller

Table 4.3 shows results of implementing this controller.

Table 4.3: Results of a fuzzy logic controller

Controller	Settling time (seconds)	Amplitude	Overshoot (%)
Roll	4	1.12	12
Pitch	4	1.12	12
Yaw	2.8	1.12	12
Altitude	3.5	1.17	17

By looking at responses and table above, it is observed that is an overshoot in all controllers, but the settling time is becoming better than PD controller case.

(3) Adaptive fuzzy-PD controller

Responses of the overall system including quadcopter where fuzzy logic controller acts together with PD controller are shown in Figure 4.7

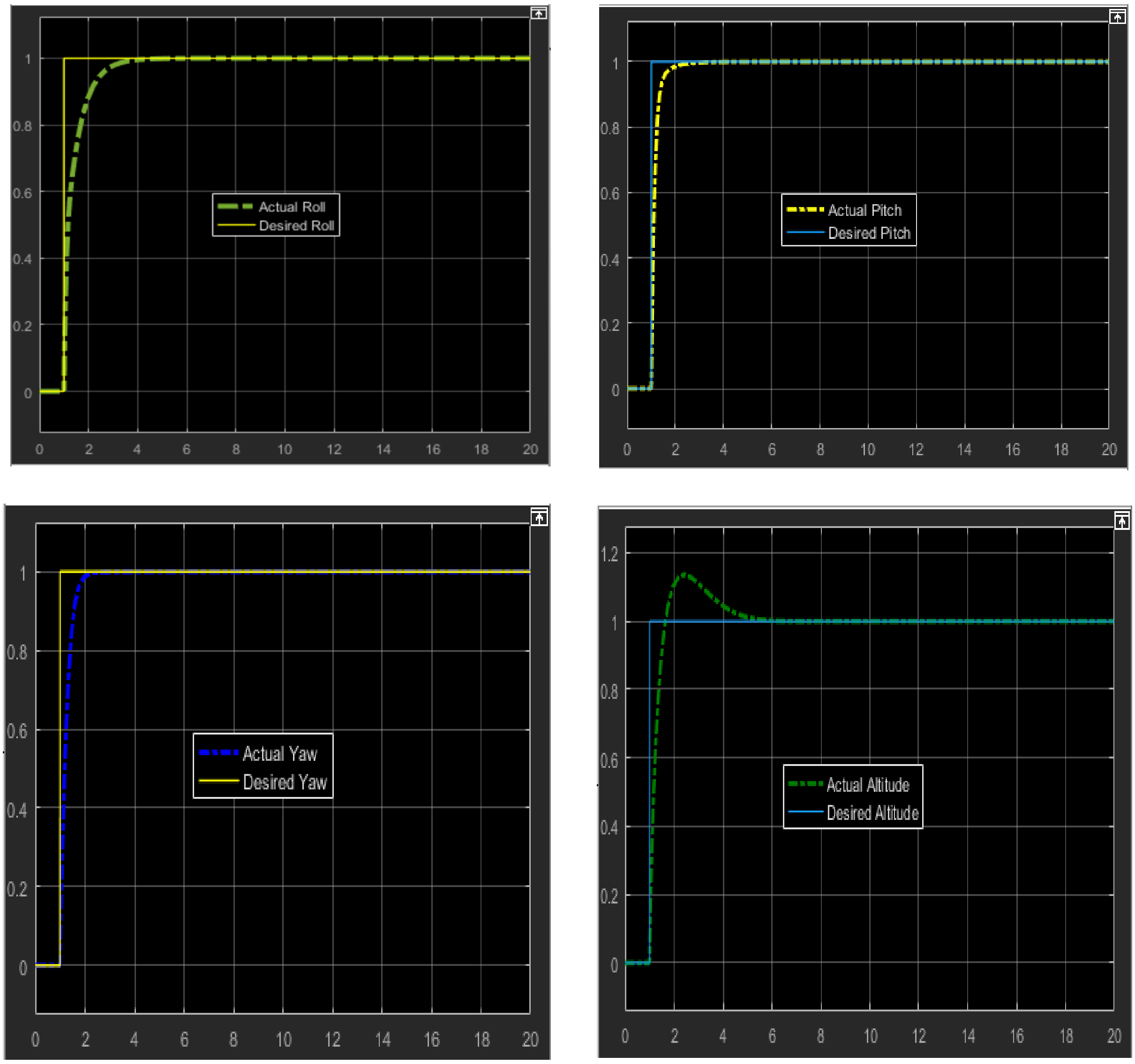


Figure 4.7: Overall responses with adaptive fuzzy-PD controller

Table 4.4 shows results of implementing this controller.

Table 4.4: Results of adaptive fuzzy-PD controller

Controller	Settling time (seconds)	Amplitude	Overshoot (%)
Roll	3.5	1	0
Pitch	1.5	1	0
Yaw	2.5	1	0
Altitude	3.5	1	0

By looking at responses and table above, it is observed that there is no steady state error as well as overshoot in all controllers. Also, the settling time is reduced to minimum value than previous two cases.

4.1.3 Comparison between controllers' performance

Here there are comparisons between different used controllers (PD, fuzzy and adaptive fuzzy-PD) when each applied to attitude and altitude subsystems.

(1) Roll control

Figure 4.8 shows effects of above controllers when applied to roll angle system.

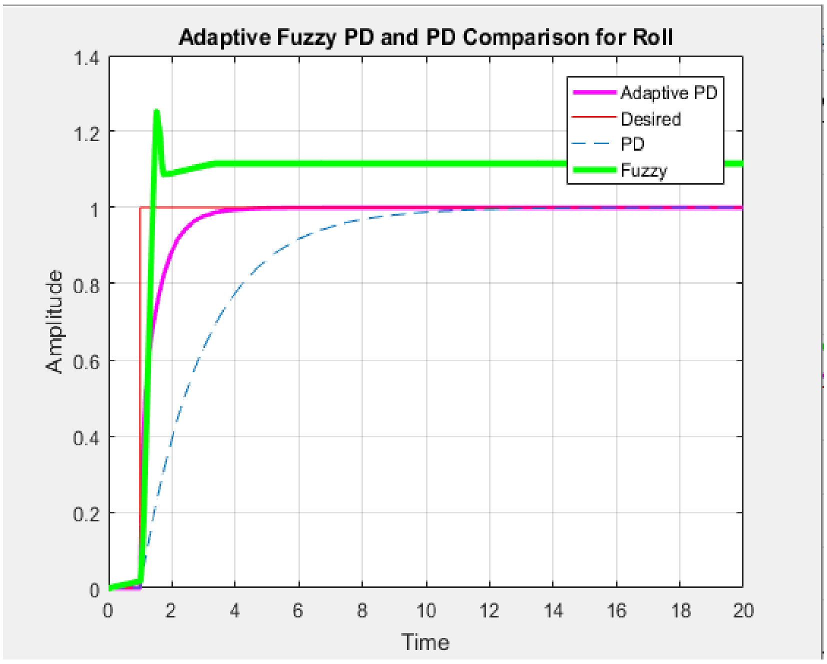


Figure 4.8: Responses from the different controllers when applied to roll angle system

From above figure the following are concluded, settling time is best in case of adaptive fuzzy-PD case and there is no overshoot or undershoot recorded in adaptive fuzzy-PD controller. Improvement in settling time and amplitude or overshoot can be seen in Table 4.5 below.

Table 4.5: Results from applying the different controllers to roll system

System	Settling Time (sec)			Overshoot (%)		
Roll	PD	Fuzzy	Adaptive fuzzy	PD	Fuzzy	Adaptive fuzzy
	12	4	3.5	0	12	0

(2) Pitch control

Figure 4.9 show how three above controllers affect pitch system.

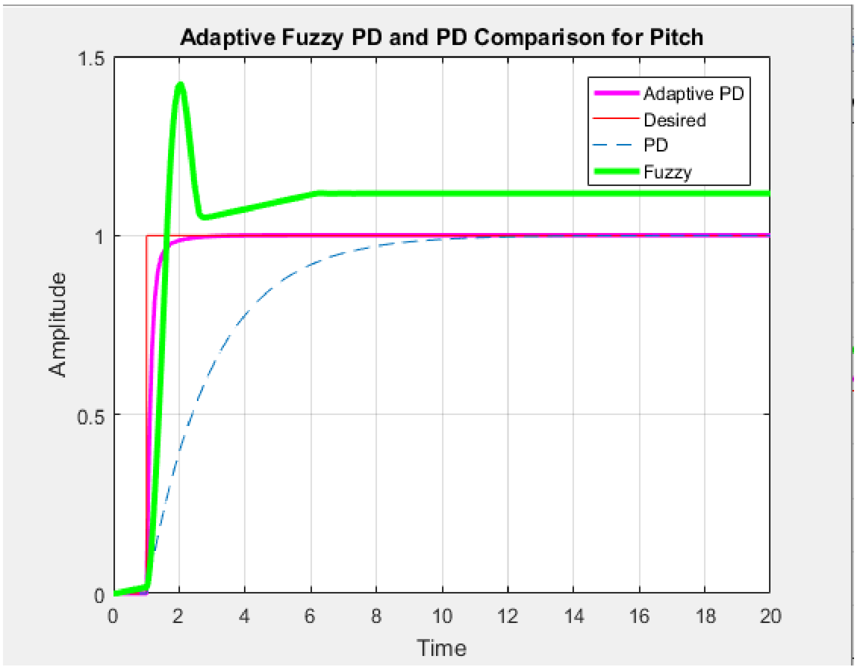


Figure 4.9: Responses from the different controllers when applied to pitch angle system

According to previous figure the following are concluded, no oscillations were observed and the response of the system was almost the same as the input which wasn't the case in fuzzy control technique. The values of settling time and overshoot for the three controllers are given in Table 4.6.

Table 4.6: Results from applying the different controllers to pitch system

System	Settling Time (sec)			Overshoot (%)		
Pitch	PD	Fuzzy	Adaptive fuzzy	PD	Fuzzy	Adaptive fuzzy
	12	4	1.5	0	12	0

(3) Yaw control

Figure 4.10 shows the responses from apply the three controllers to yaw system.

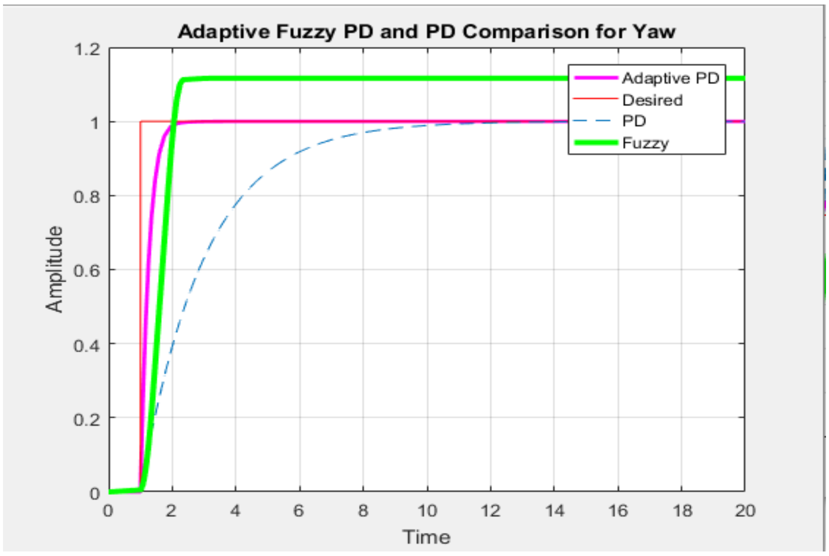


Figure 4.10: Responses from the different controllers when applied to yaw angle system

From above the following are concluded, it can be clearly observed that the adaptive fuzzy technique is the most suitable technique because the issue of large settling time in PD technique and the amplitude of the response above that required response in the fuzzy control technique is resolved. Table 4.7 shows comparison for these two figures of merits.

Table 4.7: Results from applying the different controllers to yaw system

System	Settling Time (sec)			Overshoot (%)		
Yaw	PD	Fuzzy	Adaptive fuzzy	PD	Fuzzy	Adaptive fuzzy
	12	2.8	2.5	0	12	0

(4) Altitude control

Figure 4.11 shows how above controllers affects altitude response.

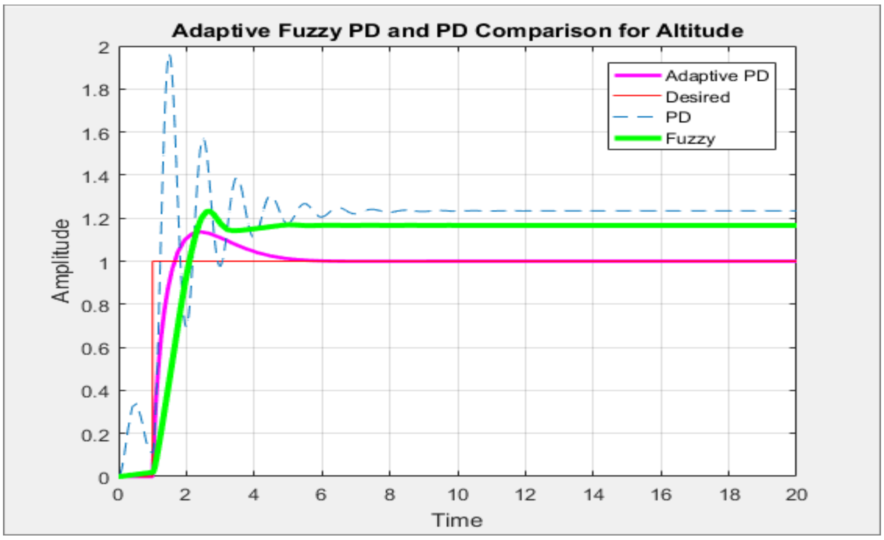


Figure 4.11: Responses from the different controllers when applied to altitude system

From above figure the following are observed, controlling the altitude is the most difficult and challenging task in the control of quadcopter and when using PD controller there are lots of overshoot as well as undershoot in the system which is not acceptable. Table 4.8 shows variation and improvements in settling time and overshoot as each of the controllers applied.

Table 4.8: Results from applying the different controllers to altitude system

System	Settling Time (sec)			Overshoot (%)		
Altitude	PD	Fuzzy	Adaptive fuzzy	PD	Fuzzy	Adaptive fuzzy
	10	3.5	3.5	20	17	0

The final values for PD controller after tuning by different methods are shown in Table 4.9.

Table 4.9: Final PD values

Controller	K_p	K_d
PD controller	0.1308	0.02
Fuzzy controller	0.150	0.04
Adaptive fuzzy-PD controller	0.216	0.03

4.1.4 Simulation of autonomous flying

The control of autonomous flying was done using simulator panel in mission planer software (MP) to simulate an overall mission as shown in Figure 4.12.

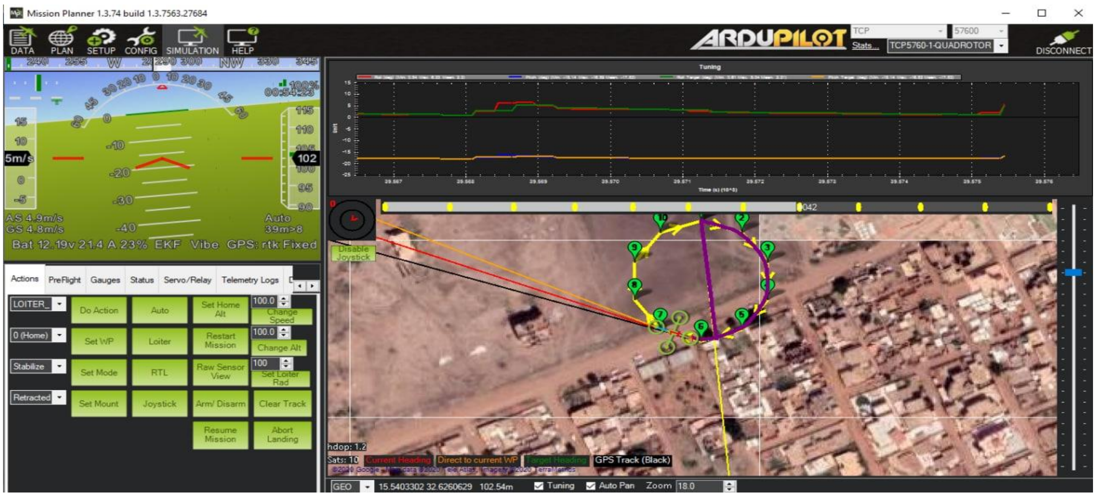


Figure 4.12: Autonomous flying over a mission

4.2 Results of detection and operation

(A) Comparison between raspberry pi and mobile phone performance

Table 4.10 summarize the results investigated by observing the performance of mobile phone and raspberry pi.

Table 4.10: Comparison between mobile and raspberry pi performance

Comparison term	Raspberry Pi	Mobile
Number of hardware required	2	1
GPS	Yes	No
Speed of detection	Slow	Fast
IoT	Yes	Yes
Trajectory control	Difficult to do	Easy to do
Effect on stabilization (by weight effect)	Little effect	Large effect
Flight time	Doesn't affect much	Reduce flight time much
Accuracy	Less Accurate	More accuracy

(B) Object detection using YOLOv3

Object detection process using YOLOv3 network is shown in Figure 4.13.

Figure 4.13: Practical object detection using YOLOv3

The mean average precision (mAP) was used to evaluate the performance since it is the most common metric for object detection algorithms, and for the training with six epochs the mAP values were as follow in Table 4.11.

Table 4.11: mAP values from training

Epochs	mAP
Epoch (1)	0.3
Epoch (2)	1.7
Epoch (3)	2.7
Epoch (4)	5.8
Epoch (5)	6.2
Epoch (6)	6.7

To measure and calculate mAP, the used intersection over union (IoU) of a value [0.5,0.95]. From the table it is observed that as epochs increased the mAP gets higher which means better detection precision and identification of classes and this is because YOLOv3 uses 53 convolutional layers. For detection, 53 more layers are added, giving a total of 106 layers, which means accurate features extraction with almost all features extracted.

(C) Mask/no mask classification using MobileNetv2

The operation is as follows in Figure 4.14.

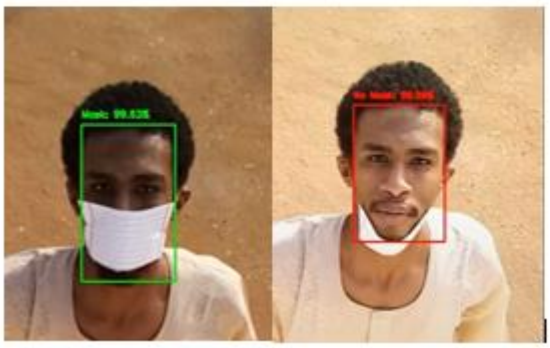


Figure 4.14: Practical mask classifier

The facemask classifier was built and tested which gives excellent results as shown in Figure 4.15 and Table 4.12.

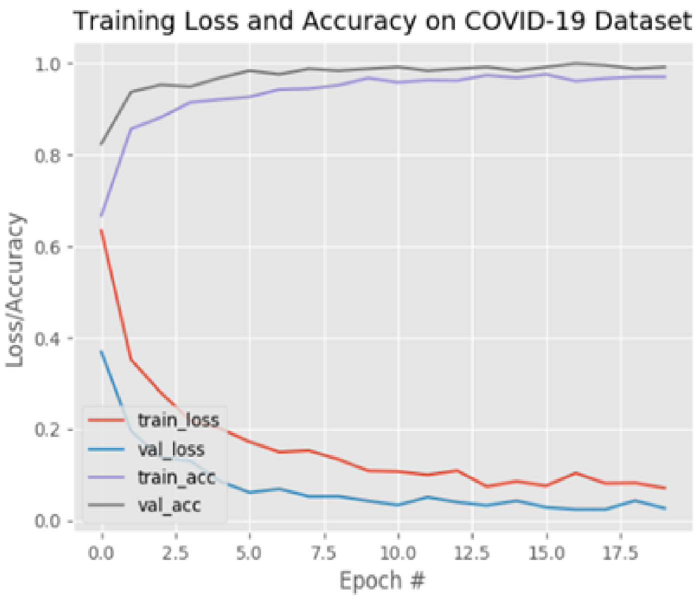


Figure 4.15: Training and validation accuracies and losses for mask classifier

From above figure it can be seen that as the number of epochs increased the training and validation accuracies are increased, while training and validation losses are decreased which means that the network or the classifier has a very good generalization for unseen data.

Table 4.12: Classification report

	Precision	Recall	F1-score	Support
With mask	0.99	1.00	0.99	138
Without mask	1.00	0.99	0.99	138
Accuracy	-	-	0.99	276
Macro avg	0.99	0.99	0.99	276
Weighted avg	0.99	0.99	0.99	276

From above table it can be seen that the classifier has an excellent performance as precision, recall and accuracy for the two classes shows and this is due to using depth-wise separable convolutions as efficient building blocks, linear bottlenecks between the layers which prevents nonlinearities from destroying too much information and using shortcut connections between the bottlenecks.

(D) Landmine detection using ACF detector

The detection operation is shown in Figure 4.16.

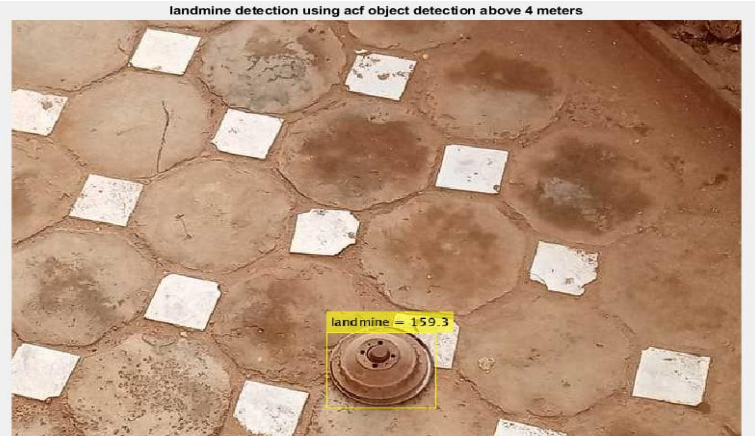


Figure 4.16: Detecting landmine

The results are shown in Table 4.13.

Table 4.13: Landmine detection results

Number of training stages	Precision	Recall
4 stages (constant)	0.9	0.88

From the table, it is cleared that ACF detector achieve very good results on custom datasets which prove its effectiveness.

(5) Fire/smoke classifier

An illustration of classification process is shown in Figure 4.17.

Figure 4.17: Fire/smoke classifier

Figure 4.18 shows training and validation accuracies and losses during training.

Figure 4.18: Training and validation accuracies and losses for fire/smoke classifier

The performance metrices are shown in Table 4.14.

Table 4.14: Performance metrices for the fire/smoke classifier

	Precision	Recall	F1-score	Support
Non-Fire	0.96	0.90	0.93	674
Fire	0.83	0.94	0.88	351
Accuracy	-	-	0.91	1001
Macro avg	0.90	0.92	0.91	1001
Weighted avg	0.92	0.91	0.91	1001

From above Figure and table, it is observed that the designed CNN is very good and has a very good generalization to unseen data. Also, it is observed that the classifier has high performance metrices which prove that the classifier can be used in real world applications effectively.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

While control engineering world today runs toward the use of intelligent and adaptive control methods, the machine learning word in a parallel advance take huge steps toward the development of computer vision algorithms and techniques due to current uses in important and critical situations all over the world.

Keeping that as a foundation for this project, firstly a system was designed which was chosen to be a quadcopter system due to many reasons which satisfy project objectives perfectly from both control view and machine vision view. Then a quadcopter vision system was implemented using some computer vision algorithms and networks – to detect objects, classify images and detecting landmines- to be in the same line with current development in quadcopter systems. At the same time to investigate the powerful effects of intelligent and adaptive control methods fuzzy and adaptive fuzzy controllers were applied to control and stabilize this non-linear, complex system with uncertainties and noises along with 6DOF which make control difficult, do so to reduce these issues as much as possible and achieve optimal performance.

The mathematical model was developed which was also considered rotor dynamics and aerodynamics effects and simulated in MATLAB/Simulink with three controller types to study them and make comparison between results. Also, the results for the vision system performance were recorded and discussed.

5.2 Recommendations

- Use more training epochs, do data exploratory analysis and use more advanced data processing techniques to achieve superior results.
- Implement other state of arts object detection and classification algorithms and networks.
- Using Generative Adversarial Networks (GANs) to edit images of a dataset or generate a dataset with any size for training any algorithm which gives better generalization.
- Use more powerful infra-red camera for the purpose of detecting buried landmines.
- Implement other intelligent control methods such as evolutionary algorithms or artificial neural networks to tune PID controller.
- Implement other control system design techniques such as linear quadratic regulator (LQR), sliding mode control (SMC), back stepping motor control and robust control techniques.
- To make the performance of the fuzzy controller better more fuzzy rules and membership functions can be utilized or make a mix between artificial neural networks and fuzzy controllers to generate neuro-fuzzy controller and apply it.
- Implement internet of things (IoT) technology for increase distance that the quadcopter can achieve by using drone kit on raspberry pi.

REFERENCES

AND

APPENDICES

REFERENCES

[1] Ashish Tewari, “Modern Control Design with MATLAB and SIMULINK”, John Wiley and Sons, printed in the United States of America, 2002.

[2] R. C. Dorf, R. H. Bishop, “Modern Control Systems”, NJ, Upper Saddle River:Pearson Prentice Hall, 2005.

[3] G. AurÈlien, “Hands-On Machine Learning with Scikit-Learn and Tensor Flow: Concepts Tools and Techniques to Build Intelligent Systems”, O’Reilly Media, Newton, MA, USA, 2017.

[4] Chollet, Francois, “Deep Learning with Python”, Manning Publications, 2017.

[5] R. K. Sinha, R. Pandey, and R. Pattnaik, “Deep Learning For Computer Vision Tasks : A review”, 2017.

[6] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks”, Proc. IEEE Conf. Comput. Vision Pattern Recognit., pp. 4510-4520, 2018.

[7] Redmon, J., Divvala, S., Girshick, R., Farhadi, A., “You only look once: Unified, real-time object detection”, in CVPR, 2016.

[8] P. Joshi, D. M. Escriva and V. Godoy, “OpenCV By Example”, Packt Publishing, 2016.

[9] Ja Iled, Fares and Ilia Voronkov, “Object Detection using Image Processing”, 2016.

[10] R. Jain, R. Kasturi and B. G. Schunck, “Machine Vision”, McGraw-Hill, New York, 1995.

[11] E. N. Kaur and E. Y. Kaur, “Object classification Techniques using Machine Learning Model”, no. January, doi: 10.14445/22312803/IJCTT-V18P140, 2019.

[12] R. Appel, S. Belongie, P. Perona, and P. Doll, “Fast Feature Pyramids for Object Detection”, pp. 1–14, 2017.

[13] S. M. Program, “Quadcopter Flight Mechanics Model and Control Algorithms”, Space Masters Program Faculty of Electrical Engineering Department of Control Engineering, no. May, 2016.

[14] Group1, “Autonomous Navigation of a Quadrotor Helicopter Using GPS and Vision Control”, presentation presented at; 2009.

[15] O. J. Al-zogphy, O. I. Khalil, and A. Monir, “Design and Manufacturing of Quadcopter”, no. August, 2019.

[16] C. O. F. A. Quadcopter, M. Iqbal, and B. I. N. Mustaparudin, “Control of a Quadcopter”, no. February, 2016.

[17] R. Bello, “Literature Review on Landmines and Detection Methods”, vol. 3, no. 1, pp. 27–42, doi: 10.5923/j.fs.20130301.05, 2013.

[18] Veloni A, Palamides A, “Control System Problems Formulas, Solutions, and Simulation Tools”,1st ed, The Taylor and Francis Group, 2012.

[19] H. O. Bansal, R. Sharma, and P. R. Shreeraman, “PID Controller Tuning Techniques : A Review”, no. May, 2017.

[20] Y.C. Shin, C. Xu, “Intelligent systems: Modeling, optimization, and control”, Taylor and Francis Group, LLC, 2009.

[21] Ch. R. Alavala, “Fuzzy Logic and Neural Networks: Basic Concepts and Applications”, New Age Int. Pvt. Ltd. Publishers, December 2008

[22] Salazar C., “Internet of Things-IOT : Definition , Characteristics , Architecture , Enabling Technologies , Application & Future

Challenges", January, 2019.

[23] D. Kotarski, P. Piljek and M. Krznar, "Mathematical modelling of multirotor UAV", International Journal of Theoretical and Applied Mechanics, vol. 1, pp. 233-238, 2016.

[24] R. Sovon, "A thesis on Quadcopter", 2017.

[25] K. Govender, "Electrical Design Project – Final Report Quadcopter", 2014.

[26] N. H. Abbas and A. R. Sami, "Tuning of PID Controllers for Quadcopter System using Hybrid Memory based Gravitational Search Algorithm – Particle Swarm Optimization", vol. 172, no. 4, pp. 9–18, 2017.

[27] T. Bresciani, "Modelling , Identification and Control of a Quadrotor Helicopter", no. October, 2008.

[28] A. Zulu and S. John, "A Review of Control Algorithms for Autonomous Quadrotors", no. December, pp. 547–556, 2014.

[29] A. Sarhan and S. Qin, "Adaptive PID Control of UAV Altitude Dynamics Based on Parameter Optimization with Fuzzy Inference", vol. 6, no. 4, pp. 246–251, doi: 10.7763/IJMO.2016.V6.534, 2016.

[30] T. Luukkonen, "Modelling and control of quadcopter", Independent research project in applied mathematics, 2011.

APPENDIX A

QUADCOPTER PARAMETERS AND PLOTTING

A.1 Quadcopter Parameters

```
global Jr Ix Iy Iz b d l m g Kpz Kdz Kpp Kdp Kpt Kdt Kpps Kdps
ZdF PhidF ThetadF PsidF ztime phitime thetatime psitime Zinit Phiinit
Thetainit Psiinit Uone Utwo Uthree Ufour Ez Ep Et Eps
% Quadrotor constants
Ix = 7.5*10^(-3);  % Quadrotor moment of inertia around X axis
Iy = 7.5*10^(-3);  % Quadrotor moment of inertia around Y axis
Iz = 1.3*10^(-2);  % Quadrotor moment of inertia around Z axis
Jr = 6.5*10^(-5);  % Total rotational moment of inertia around the propeller
b = 3.13*10^(-5);  % Thrust factor
d = 7.5*10^(-7);  % Drag factor
l = 0.23;  % Distance to the center of the Quadrotor
m = 0.65;  % Mass of the Quadrotor in Kg
g = 9.81;   % Gravitational acceleration
```

A.2 Plotting

```
sim('QuadPID_V1');
sim('SAHI_HAI_G');
sim('ll');
figure(1)
p=plot(Adap_Phi(:,1),Adap_Phi(:,2),'r',Adap_Phi(:,1),Adap_Phi(:,3),'m',PD_
PHI(:,1),PD_PHI(:,3),'--',F_PHI(:,1),F_PHI(:,3),'g');
p(1).LineWidth = 2;
p(4).LineWidth=3;
```

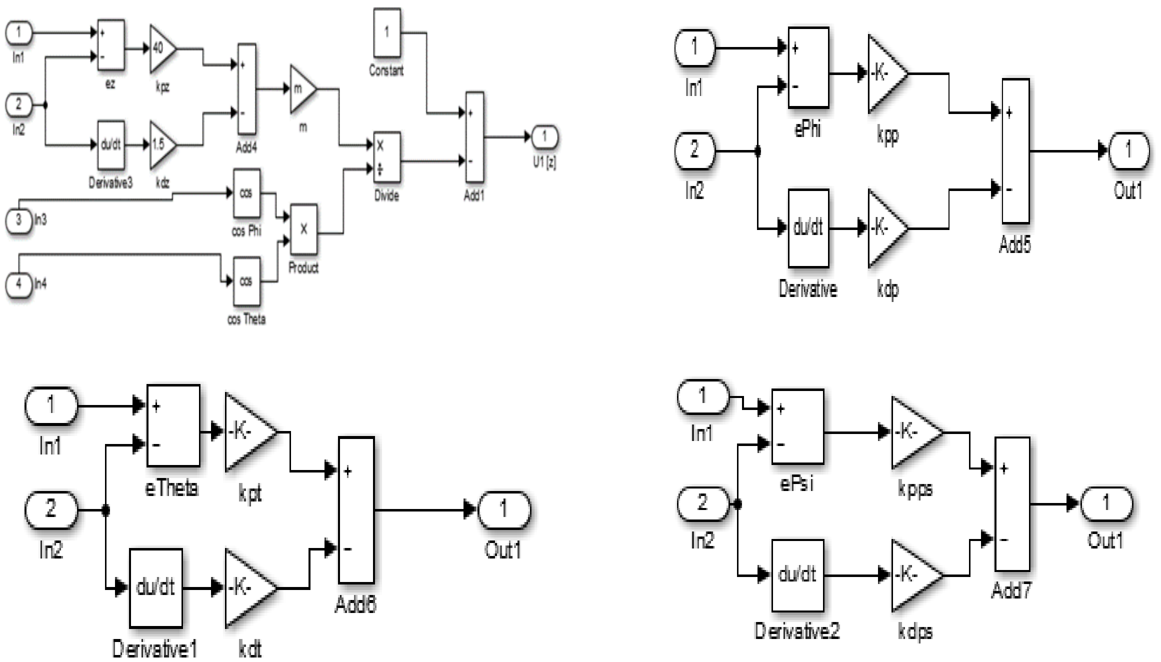
```
grid on
title('Adaptive Fuzzy PD and PD Comparison (PHI)')
xlabel('Time')
ylabel('Amplitude')
legend('Adaptive PD','Desired','PD','Fuzzy')
figure(2)
q=plot(Adap_Theta(:,1),Adap_Theta(:,2),'r',Adap_Theta(:,1),Adap_Theta(:,3)
,'m',PD_Theta(:,1),PD_Theta(:,3),'--',F_THETA(:,1),F_THETA(:,3),'g');
q(1).LineWidth = 2;
q(4).LineWidth=3;
grid on
title('Adaptive Fuzzy PD and PD Comparison (THETA)')
xlabel('Time')
ylabel('Amplitude')
legend('Adaptive PD','Desired','PD','Fuzzy')
figure(3)
r=plot(Adap_Psi(:,1),Adap_Psi(:,2),'r',Adap_Psi(:,1),Adap_Psi(:,3),'m',PD_Ps
i(:,1),PD_Psi(:,3),'--',F_PSI(:,1),F_PSI(:,3),'g');
r(1).LineWidth = 2;
r(4).LineWidth=3;
grid on
title('Adaptive Fuzzy PD and PD Comparison (PSI)')
xlabel('Time')
ylabel('Amplitude')
legend('Adaptive PD','Desired','PD','Fuzzy')
figure(4)
s=plot(Adap_Z(:,1),Adap_Z(:,2),'r',Adap_Z(:,1),Adap_Z(:,3),'m',PD_Z(:,1),P
D_Z(:,2),'--',F_Z(:,1),F_Z(:,2),'g');
s(1).LineWidth = 2;
s(4).LineWidth=3;
```

```
grid on
title ('Adaptive Fuzzy PD and PD Comparison (Z)')
xlabel('Time')
ylabel('Amplitude')
legend('Adaptive PD','Desired','PD','Fuzzy')
clc
```

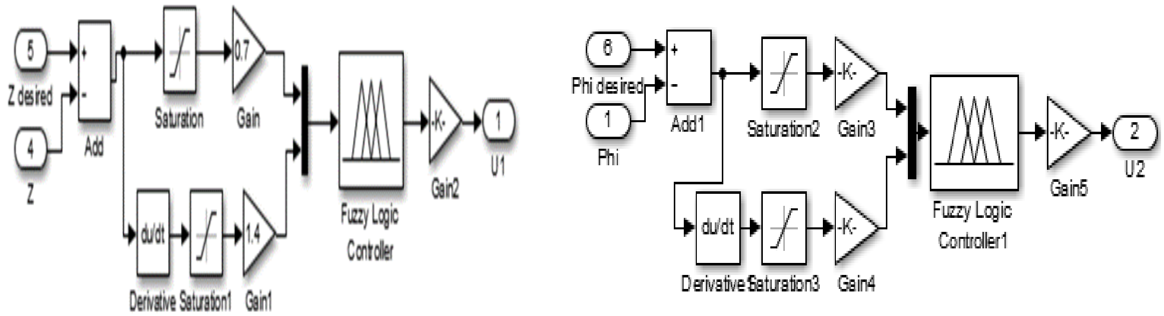
APPENDIX B

SIMULINK DIAGRAMS

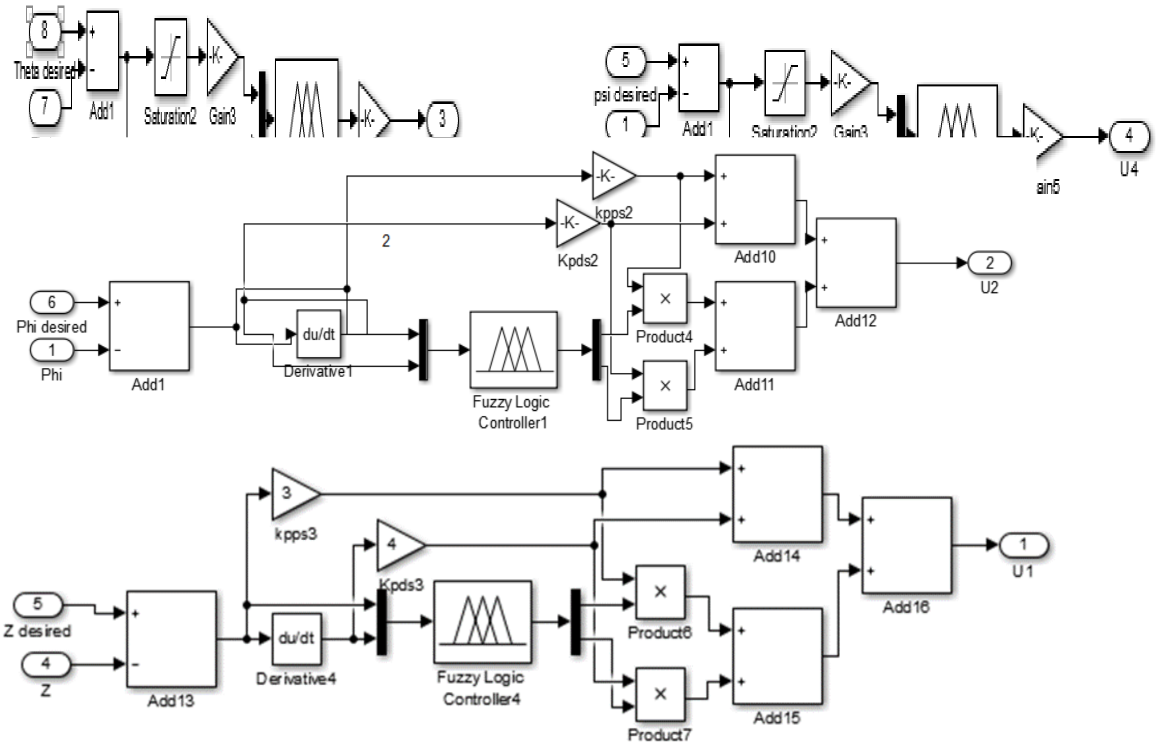
B.1 PD Controller

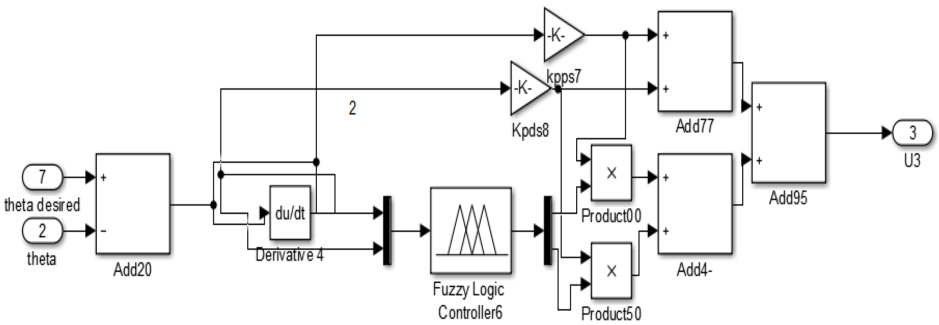


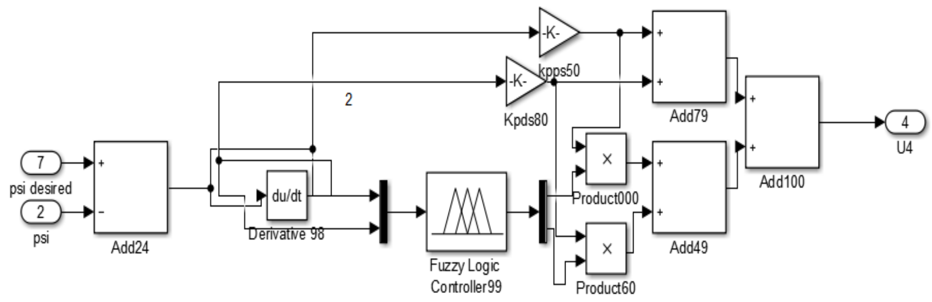
B.2 Fuzzy Controller



B.3 Adaptive Fuzzy-PD Controller







APPENDIX C

ACF OBJECT DETECTOR CODE

```
clear('web')
web= ipcam('http://10.241.37.14:8080/video');
img=snapshot(web);
dete= detect(acfDetector,img);
while 1
img=snapshot(web);
im2= rgb2gray(img);
bbox=detect(acfDetector,img);
pic=insertObjectAnnotation(img,'rectangle',bbox,'landmine');
imshow(pic);
pause(0.01);
end
```