ADDIS ABABA UNIVERSITY
ADDIS ABABA INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

TARGET TRACKING OF QUAD-ROTOR UAV USING ADAPTIVE SLIDING MODE CONTROLLER BASED ON REAL TIME IMAGE-PROCESSING

A thesis submitted to Addis Ababa Institute of Technology, School of Graduate Studies, Addis Ababa University

in partial fulfillment of the requirement for the Degree of Master of Science in Electrical Control Engineering

By

Henok Guta Woldehnna

Adviser: Lebsework Negash (PhD)

ADDIS ABABA, ETHIOPIA
March 2023

ADDIS ABABA UNIVERSITY
ADDIS ABABA INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

TARGET TRACKING OF QUAD-ROTOR UAV USING ADAPTIVE SLIDING MODE CONTROLLER BASED ON REAL TIME IMAGE-PROCESSING

By Henok Guta Woldehanna

APPROVED BY BOARD OF EXAMINERS

_____	_____
Dean, School of Graduate Committee	Signature
Lebsework Negash (PhD)	_____
Advisor	Signature
_____	_____
Internal Examiner	Signature
_____	_____
External Examiner	Signature

ACKNOWLEDGMENT

I would like to thank God, for letting me through all the difficulties. I have experienced your guidance day by day.

I would like to acknowledge and give my warmest thanks to my advisor LEBSEWORK NEGASH (PhD) who made this work possible. His guidance and advice carried me through all the stages of starting from selecting the title to writing my thesis. The countless times he spent his invaluable time, advice, and assistance on a very busy schedule will not be forgotten.

I would also like to give special thanks to my family and friends as a whole for their continuous support and understanding when undertaking my research and writing this thesis. Your prayer for me was what sustained me this far.

Finally to my caring, loving and supportive wife, RUTH HADDIS ZEWDE: my deepest gratitude. Your encouragement when the times got rough are much appreciated and duly noted. It was a great journey of comfort and relief to know that you were willing to provide management of our household activities while I completed my work and studies. I have nothing to offer other than heartfelt thanks.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

3DR	three-dimensional recording
AGL	above ground level
ALT	altitude
ARM	application reference model
AR	augmented reality
ASMC	adaptive sliding mode control
BLDC	brushless direct current
CLM	climb
CNN	convolution neural network
COORDs	coordinates
CPU	central processing unit
CUDA	compute unified device architecture
DES	descent
DJI	da-jiang innovations
EMMC	embedded multi-media card
FAA	federal aviation administration
FOV	field of view
FPV	first person view
GPS	global positioning system
GPU	graphics processing unit
GSD	ground sampling distance
IMU	inertial measurement unit
LAT	latitude
LOG	longitude
LPDDR	low power double data rate
LTI	linear time invariant
MAV	micro aerial vehicle
NED	north east down
OPS	operations

PID	proportional integral derivative
RCNN	region-based convolution neural network
RPM	revolution per second
SMC	sliding mode control
SODIMM	small outline dual in-line memory module
UAV	unmanned aerial vehicle
VOC	visual object classes
UPS	united state postal service
VSC	variable structure control
YOLO	you only look once

ABSTRACT

Surveillance plays a crucial role in various military and civilian operations, including search and rescue missions. In recent times, unmanned aerial vehicles (UAVs) have gained significant popularity and are considered the ideal resources for such applications.

The quad-rotor offers distinct advantages over fixed-wing light vehicles, including cost-effectiveness, compact size, and the ability to hover, as well as perform vertical takeoff and landing. However, to ensure reliable performance in various tasks, it is crucial to have a specialized controller that can effectively account for the quad-rotor's nonlinear dynamics, under-actuated characteristics, and uncertainties related to parameters and external disturbances.

To address these challenges and enhance controller robustness, a slide mode control technique is employed, which offers advantages over traditional PID and other nonlinear controllers. This control design also incorporates image processing capabilities to enable real-time identification and tracking of user-defined targets, providing efficient and accurate performance.

This thesis aims to develop real-time target identification and tracking system based on image processing. The system utilizes fast decision-making capabilities and robust flight control techniques to ensure optimal trajectory for a quad-rotor. The image processing component relies on an onboard camera and employs the You Only Look Once (YOLO) algorithm to identify and estimate the continuous motion coordinates of the target. The identification process primarily relies on user-defined criteria such as shape, size, and color. The graphics processor of the embedded software is responsible for accurately calculating the target dynamics relative to a common reference frame of Earth's geographic coordinates. Furthermore, the system selects efficient maneuvers based on time and energy considerations, leveraging the capabilities of the YOLO algorithm.

In the presence of parametric uncertainties and external disturbances, the controller effectively minimizes tracking errors within a short time frame, ensuring obstacle clearance and reducing redundancy costs. Through simulation results, the designed controller demonstrates minimal altitude and attitude tracking errors, achieving precise identification and tracking of a user-defined ground target.

Key words: - Quad-rotor UAV, ASMC, Target Tracking, YOLO Algorithm and Navigation.

CHAPTER 1
INTRODUCTION

1.1. BACKGROUND

Quad-rotor is the most popular configuration of multi-rotor unmanned aerial robot, extensively for civilian applications. This is possibly attributed to their certain features, such as simple mechanical structure, hovering, and agile maneuverability. With the increasing requirement of autonomous flight under different conditions, control of the quad-rotor is an important challenge. Compared with traditional manned airplane and unmanned fixed-wing light vehicles, quad-rotor has its unique advantages, such as low cost, small size, and hovering and vertical takeoff and landing. To perform tasks with high reliability, the quad-rotor UAV requires good light control capabilities. therefore, the of a specialized controller, which is able to take into account the quad-rotor's modeling nonlinearity with strongly coupled dynamics, under-actuated characteristics, as well as parametric uncertainty, is always desired.

Sliding mode control (SMC) is a robust and effective method of controller design in nonlinear systems under uncertain conditions. SMC is designed for a class of under-actuated system, and as a typical example, the algorithm is verified in position and attitude stabilization control of a quad-rotor UAV.

Parametric uncertainties and external disturbance have not been considered simultaneously under the framework of SMC [1]. As a matter of fact, the parameters of a quad-rotor, such as the moments of inertia, are difficult to be measured directly. It is also hard to get the exact values of these parameters even if they can be measured in certain ways. Meanwhile, quad-rotor UAVs are inevitably subject to external disturbances during flight, such as wind gusts. Particularly, robustness issues may be critical for the control of quad-rotor since they are subjected to the coupled dynamics and external disturbances.

Although the quad-rotor has a series of advantages, it is unstable and under-actuated dynamic system with high nonlinearity and strong coupling. Moreover, it is easily affected by near-surface airstream. In particular, the actuator dynamic and aerodynamic effects must be investigated to establish a reliable dynamic model of quad-rotor. Control method dealing with the nonlinearity and coupling property of quad-rotor has to be proposed for precise flight control.

In this work, the quad-rotor system is considered subject to nonlinear and coupling dynamics, parametric uncertainties, and external disturbances [2]. A robust adaptive sliding mode controller (ASMC) is developed to asymptotically reduce the altitude and attitude tracking errors. Reliable nonlinear dynamic model is presented based on the analysis of actuator dynamic, aerodynamic effect, and rigid body dynamic. The gyroscope effect of the rotors is considered by dividing the quad-rotor into body part and rotor part. It makes the dynamic model more reliable to take actuator dynamic and aerodynamic effect into account. The dynamic model is a combination of actuator dynamic, aerodynamic effect, and rigid body dynamic [3].

Flying insects have tiny brains and mostly possess compound eyes which can get panoramic scene to provide an excellent flying performance. Comparing with state-of-the-art artificial visual sensors, the optics of compound eye provides very low spatial resolution. Nevertheless, the behavior of flying insects is mainly dominated by visual control. They use visual feedback to stabilize flight, control flight speed, and measure self-motion. On the other hand, highly accurate real-time stabilization and navigation of unmanned aerial vehicles (UAVs) or micro-aerial vehicles (MAVs) is becoming a major research interest, as these flying systems have significant value in surveillance, security, search, and rescue missions [4].

The yaw rate control is totally based on the visual feedback of the on-board camera. The reference velocity value is provided by the on-board inertial measurement unit (IMU) [5].

The visual feedback will assist the controller that the actual geo-location of the UAV is coincides with the target location and where the UAV should satisfy the user defined operational specifications. This can be achieved through information gathered by onboard camera and the image processing unit, which is expected to primarily identify the target using you only look once (YOLO) algorithm and provides and updates the target pixel location on the image frame of the live picture continuously. Simultaneously this image pixel location will mapped to actual ground position using photo-grammetry technique. Finally the trajectory generator will kicks in and send a maneuver signal to the controller to track the target [6]. The process will continue looping until the target image frame center overlaps on the UAV geometric location, which is also cross checked by GPS unit onboard.

1.2. PROBLEM DESCRIPTION

The design of a robust quad-rotor UAV for ground target tracking poses an intriguing challenge. The initial hurdle arises in developing an efficient and autonomous flight controller that can operate reliably. Currently, many UAVs are operated by a human, which introduces inefficiencies and errors due to factors such as fatigue, stress, or distractions. Robustness becomes crucial when considering operation in various challenging scenarios, including adverse weather conditions like wind gusts, low visibility due to snow or rain, congested environments, or difficult terrain topologies.

Traditional approaches to address these constraints often lead to sophisticated and costly designs. However, by incorporating modern control theory concepts and artificial intelligence systems, we can overcome these limitations while minimizing costs and significantly increasing system efficiency. In this research thesis, we propose a controller design and target tracking techniques that are not only cost-effective but also robust. These advancements aim to enhance the overall performance and reliability of the quad-rotor UAV in ground target tracking applications.

1.3. OBJECTIVIES OF THE THESIS

1.3.1. General objective

This thesis is designed to develop a Adaptive Slide Mode Controller for Quad-rotor where independent and coupled altitude and attitude (Pitch, Roll and Yaw) controllers' design and stability analysis will be accomplished. An embedded visual feedback will assist the controller to identify and geo-locate the target for tracking.

1.3.2. Specific objective

The specific objectives of the thesis are;

i. To design a Adaptive Slide Mode Controller for a Quad-rotor
ii. To eliminate the altitude and attitude tracking errors that converge asymptotically
iii. To identifies and Track a user defined ground target
iv. To evaluate the performance on MATLAB/ SIMULINK and Python

1.4. SCOPE OF THE THESIS

Designing a robust UAV flight controller, perform stability analysis and verification through simulation under random external disturbances is the main focus of this thesis. Additionally recommendation on implementation of the entire system is also included at the end which, can be accomplished in near future.

To verify the operational capability of the controller integrating with the image processing unit, with respect to a user defined ground target.

1.5. SIGNIFICANCE OF THE THESIS

i. Search and rescue, traffic vehicle pursuit and surveillance operations can be augmented with such kind of autonomous systems for better and efficient delivery. Since the target and operational area is user defined, additional applications can be accomplished just by extending the system capability.

ii. The thesis proposes a robust and adaptive controller can be used for other advanced military purposes.

1.6. OUTLINE OF THE THESIS

The intent of this thesis is

i. Design and analyze ASMC controller for a quad rotor UAV, which includes: mathematical modeling, stability analysis and performance evaluation using MATLAB/SIMULINK simulation.

ii. Design a navigation and trajectory generation units for control input

iii. To design image processing YOLO algorithm tool to identify and track a user defined target in real time.

iv. To integrate the controller and the image processing unit and simulate to evaluate the entire system performance on user defined scenario.

CHAPTER 2

LITERATURE REVIEW

2.1. A BRIEF HISTORY OF THE UAV

The first heavier-than-air aerodyne to take off vertically was a four-rotor helicopter designed by Louis Breguet. It was tested only in tethered flight and to an altitude of a few feet. Beginning in mid-1800s, winged designs were also taking shape. One such example is the Aerial Steam Carriage, shown in Figure 1, which was built in 1848 by John String, fellow and William Henson in England. Their initial model flew successfully for roughly 60 yards. In 1868, the pair also developed a model based on a tri-winged structure attached to a wire guide that could effectively generate lift and only used the wire guide to steer around obstacles.

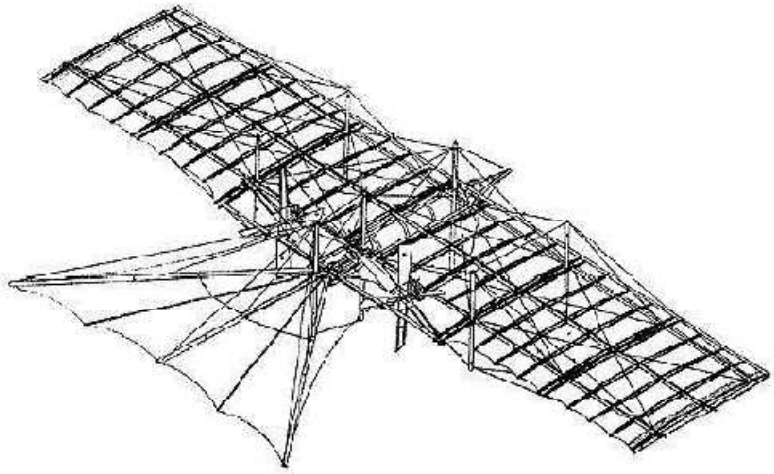


Figure 2.1: Aerial Steam Carriage

Etienne Oehmichen experimented with rotorcraft designs in the 1920s. Among the designs he tried, his helicopter No.2 had four rotors and eight propellers, all driven by a single engine. The Oehmichen No.2 used a steel-tube frame, with two-bladed rotors at the ends of the four arms. The angle of these blades could be varied by warping. Five of the propellers, spinning in the horizontal plane, stabilized the machine laterally. Another propeller was mounted at the nose for steering. The remaining pair of propellers functioned as its forward propulsion. The aircraft exhibited a considerable degree of stability and increase in control-accuracy for its time, and made over a thousand test flights during the middle 1920s. By 1923 it was able to remain airborne for several minutes at a time, and on April 14, 1924 it established the first-ever distance

record for helicopters of 360 m. It demonstrated the ability to complete a circular course and later, it completed the first 1km closed-circuit flight by a rotorcraft [7].

In the first decades of the 2000s, the quad-rotor layout has become popular for small scale unmanned aerial vehicles or drones. The need for aircraft with greater maneuverability and hovering ability has led to a rise in quad-rotor research. The four-rotor design allows quad-rotors to be relatively simple in design yet highly reliable and maneuverable. Research is continuing to increase the abilities of quad-copters by making advances in multi-craft communication, environment exploration, and maneuverability. If these developing qualities can be combined, quad-rotors would be capable of advanced autonomous missions that are currently not possible with other vehicles.

Probably the easiest to recognize military drone to this day is the famous Predator, which was used in Afghanistan to launch missiles and search for Osama Bin Laden. The drone operated at an altitude of 25,000 ft and had a maximum range of 740km, while being able to fly for 40 hours.

In 2006 First industrial and commercial uses of quad-rotors and other UAVs had its infancy with the first commercial drone permit released by the FAA. For the following 8 years they issued an average of 2 permits per year. The same year, Frank Wang, a mainland-born student at Hong Kong University of Science and Technology has an idea to revolutionize how useful quad-rotors are for the mainstream population. The company is called DJI.

The French drone manufacturer Parrot unveiled its AR Drone in 2010, the first of its kind to be controllable directly from a smart phone. It had self stabilizing gyros and an FPV camera.

Since 2013 the likes of Jeff Bezos with Amazon and other delivery companies like FedEx and UPS declare their intention to use drones for delivering their products in record times though airborne means. The only issue is that the technology, battery life and regulation haven't yet caught up with their ambitions [8].

The Phantom 3 is released and becomes one of the best sold consumer drones in history in 2015. It comes with a stabilized 4k camera, the ability to see FPV feed on your phone and fly it for up to 5 km and 23 minutes battery life, which made it the best choice at the time by far. 3DR also released a revolutionary drone that worked with the GoPro Hero and had many Smart flight modes never seen before.

Starting year 2016, the DJI Mavic was a revolutionary drone that introduced the working concept of a foldable consumer quad-rotor for the first time. This meant it was one of the most portable

quad-rotors ever made, while not compromising on any features, including camera Gimbal stabilization and a long flight time; while it's being easy to fit in any backpack.

Figure 2.2**:** DJI Mavic Pro - First Foldable consumer drone

Drones have been a staple during the corona virus outbreak, helping with medical supply deliveries, police work in social distancing and quarantine. Police uses them to maintain social distancing and firefighters use drones to find entry spots in buildings without putting themselves in the harm's way.

2.2. BACKGROUND

Drones have become more accessible to the masses, but the places they shine most are industrial uses, relief from disasters and especially future airborne transportation. This history of drones shows us that there's been an incredible acceleration in technological advancement in recent years, and it has become almost impossible to guess how things will be in the following 20-50 years.

Nonlinear control laws have been implemented for sophisticated flight control systems on board VOTL aircraft including UAVs; adaptive nonlinear control laws also incorporated either singly or in cooperation.

There are different types of nonlinear control systems, and they can be broadly classified into the following categories:

- Feedback Linearization: This approach aims to transform a nonlinear system into an equivalent linear system through an appropriate change of variables. The advantage of

feedback linearization is that it allows the application of well-established linear control techniques, such as PID control or state feedback, to control the system.

- Sliding Mode Control: Sliding mode control employs a discontinuous control law to force the system trajectory to slide along a specific manifold. The key idea is to create a sliding motion on a predefined surface, where the system behavior is more easily controllable. Sliding mode control is known for its robustness against uncertainties and disturbances.
- Adaptive Control: Adaptive control systems adjust their control parameters based on real-time measurements of the system's behavior. They aim to adapt and optimize the control actions according to the system's changing dynamics or uncertainties. Adaptive control is particularly useful when the system parameters vary over time or are not precisely known.
- Neural Network Control: Neural network-based control systems utilize artificial neural networks to learn and approximate the system dynamics and control policy. These networks are trained using data from the system and can adapt and generalize to different operating conditions. Neural network control is suitable for complex and highly nonlinear systems.

One of the most extensively used non-linear control method is Sliding Mode Controller (SMC). SMC has proved its robustness in terms of trajectory tracking, stability and insensitiveness to external disturbances and parametric uncertainties. In which it makes it the best UAV controller design strategy. The only drawback of using SMC is chattering, which is high frequency oscillations.

SMC or Variable structure control (VSC) is a form of discontinuous nonlinear control [9]. The method alters the dynamics of a nonlinear system by application of a high-frequency switching control. The state-feedback control law is not a continuous function of time; it switches from one smooth condition to another. So the structure of the control law varies based on the position of the state trajectory; the method switches from one smooth control law to another and possibly very fast speeds. VSC and associated sliding mode behaviour was first investigated in early 1950s in the Soviet Union by Emelyanov and several coresearchers.

With the rapid advancement of control technologies, there have been various intelligent control schemes established for complex systems with or without uncertain dynamics. Given a certain

control problem, desirable qualities such as simplicity, applicability, adaptability, and robustness are the touchstones of control design so as to ensure excellent control performance against system parameter variations and unpredicted external disturbances. Amongst many robust control techniques, Sliding Mode Control (SMC) has been increasingly receiving a great deal of attention in both theoretical and applied disciplines owing to its distinguishing features such as insensitivity to bounded matched uncertainties, order reduction of sliding motion equations, decoupling design procedure, and zero-error convergence of the closed-loop system, just to name a few. Nevertheless, the shortcomings inherent in conventional SMC approaches are yet to be fully addressed. For one, the chattering phenomenon has not been uprooted without compromising on the zero-error convergence. More importantly, from the control design perspective, there are certain constraints in the design of SMC, such as prior information about the bounds of uncertainties is often required, this in turn has greatly restrained the applications of SMC in many practical circumstances. Therefore, how to make the best use of SMC in order to develop a simple but effective SMC technique has remained a big challenge for both researchers and engineers in the areas of control engineering and the related technologies.

To put things in form, the major task of SMC system design is the selection of an appropriate control law in a way that alters the dynamics of a complex system by application of a discontinuous control input that drives the system states to reach the sliding hyper-surface and slide along the surface for all subsequent time. The underlying feature of the method is that once the sliding mode is reached, the system dynamics, by proper choice of a predefined hyper-surface, exhibit desirable behavior which is inherently invariant to disturbances.

Indeed, most of the control strategies developed for the complex systems requires prior knowledge of uncertain system dynamics. In practice, this is often not possible as the information about uncertain system dynamics is not achievable. In consequence, such control strategies may not be applicable to large-scale systems. It goes without saying there is still an urgent need to focus on the development of a robust controller to deal with real-time complex systems without relying on the information related to the system uncertainties and disturbances. This has piqued more intense interests in the development of a robust SMC control scheme to overcome the following major issues:

- Zero-error convergence, finite-time stability, and strong robustness.

- Lack of information about the bounds of system uncertainties and external
- Disturbances Workability and applicability to real control problems
- Ease of implementation

In order to develop a simple but effective control scheme for uncertain dynamic systems, all the existing issues mentioned earlier will be addressed appropriately in this thesis. The proposed control algorithms not only guarantee an asymptotic stability of the closed-loop system, but also allow the closed-loop system to boast a strong robust property with respect to system uncertainties and disturbances. The huge advantages of the proposed control algorithms are that the controller designs do not require prior information about uncertain system dynamic to be known, meanwhile, the chattering phenomenon that frequently appears in conventional SMC system is greatly eliminated without deteriorating the robustness of the closed-loop system.

The use of UAVs to detect, track, and locate targets is of great significance for the construction of smart cities. A UAV is used in diverse collection scenarios and can collect global target information of complex road sections, intersections, or multiple city features. In addition, it can obtain high-definition videos from the vertical upper perspective of the landmark and obtain key parameters that cannot be extracted by conventional monitoring methods.

The dynamic and complex UAV videos pose severe challenges for video detection technology. According to the principles of traditional video target detection, the texture of a target is used as a feature for detection. However, it is easily affected by light. The other is a texture less detection method that uses gradient calculations, and it does not perform well in complex environments and occlusion conditions.

In the application scenario of a UAV aerial video, the performance of the tracking algorithm is affected by many factors such as the changes in the illumination, scale, and occlusion. Unlike fixed cameras, tracking tasks in aerial videos are hampered by the low sampling rate and resolution and image jitter, which can easily lead to drift. When flying at a high altitude, the very small size of the ground objects is also a major challenge for the tracking task. In general, there are two types of video-based target tracking models, namely, the generative model and the discriminative model. The theoretical idea in the generative tracking model is that given a certain

video sequence, for the target that needs to be tracked in the video, a model is built according to the tracking algorithm. Following this, the response area that is most similar to the target in the subsequent video sequence is found and is used as the target area. In this way, the tracking task continues further. The commonly used generative tracking algorithms include the optical flow method, the particle filter method, the mean-shift algorithm, the continuously adaptive mean shift algorithm, and so on.

A generative algorithm focuses mainly on tracking the characteristic information of the target itself and conducting an in-depth search on the target characteristics. However, such algorithms often ignore the influence of other factors on tracking performance. For example, severe scale changes, background information interference, or occlusion of the target can easily lead to a situation where the target cannot be tracked. The difference between the discriminative algorithm and the generative algorithm is that the former considers the influence of the background information on the target tracking task and then distinguishes the background information from the target information. In other words, to model a discriminative tracking model, it is necessary to distinguish the target and background information in a given video sequence. After the model is established, the subsequent video sequences are searched to determine further whether the searched target or background is found or not. The common discriminative tracking algorithms include correlation filtering methods and deep learning methods. Due to the success of the deep CNN in visual recognition tasks, a large number of studies have been performed using CNN for tracking algorithms. These studies show that the accuracy of a CNN-based tracker is better than the tracking algorithm based on manual feature extraction [10].

Here target detection and tracking algorithm based on the currently available mainstream deep learning image processing algorithms, and a target location estimation model is considered. In particular, based on the “you only look once” (YOLO) algorithm, a detection model with superior robustness characteristics. One of the main advantages of YOLO is its speed. It can process images at a rate of 45 to 155 frames per second, much faster than other state-of-the-art object detection algorithms. The system detects objects on the Pascal VOC 2012 dataset and it can detect the 20 Pascal object classes: person, animals and so on [11].

2.3. SUMMARY

Advancements in motor technology, lightweight materials, and flight control algorithms have greatly improved the performance of quad-rotors. They are now capable of agile maneuvers, high-speed flight, and autonomous navigation. Additionally, the integration of sensors such as accelerometers, gyroscopes, and cameras enables improved stability, obstacle detection, and object tracking capabilities.

However, despite these advancements, there are still certain gaps and challenges in the current design of quad-rotors:

- Limited flight endurance: The limited battery capacity restricts the time they can spend in the air, which can be a limitation for certain applications that require longer flight durations.
- Safety concerns: which remains an important consideration, especially in crowded areas or when flying in close proximity to people and objects.
- Autonomous navigation and collision avoidance: While advancements have been made in autonomous navigation systems, there is still room for improvement. Ensuring robust obstacle detection, collision avoidance, and accurate mapping capabilities are crucial for safe and efficient autonomous quad-rotor flight.
- Environmental impact: Exploring more sustainable power sources and optimizing flight paths can help reduce the ecological footprint of quad-rotor operations.

Addressing these gaps and challenges will further enhance the capabilities and applications of quad-rotors. This thesis is consider to enhance quad-rotor performance by focusing on key aspects such as improving flight endurance through optimized maneuvers that minimize energy consumption. Additionally, the aim is to enhance safety features, refine autonomous navigation algorithms, and reduce the environmental impact of quad-rotors. These advancements will have a positive impact on the continued growth and integration of quad-rotors across diverse industries and domains.

CHAPTER 3

SYSTEM MODELING

3.1. DESCRIPTION OF UAVs OPERATIONAL PRINCIPLES

The schematic diagram of the quad-rotor is shown in Figure 3.1, where $_E$ and $_B$ represent the earth and body frames, respectively. The quad-rotor UAV, rotors 2 and 4 are controlled to rotate clockwise, while rotors 1 and 3 are controlled to rotate counterclockwise. It is worth noticing that the left or right motion of the quad-rotor UAV will create a roll angle by increasing (decreasing) the speeds of rotors 1 and 3 and decreasing (increasing) the speeds of rotors 2 and 4. Meanwhile, in order to obtain the pitch angle around the *y*-axis, the quad-rotor can be controlled by increasing (decreasing) the speeds of rotors 1 and 2 and decreasing (increasing) the speeds of rotors 3 and 4. Furthermore, to rotate the quad-rotor around the *z*-axis, it is suggested to increase (decrease) the speeds of rotors 1 and 4 and decrease (increase) the speeds of rotors 2 and 3. In addition, taking-off and landing tasks can be accomplished by increasing or decreasing all the rotors' speed uniformly.

The following assumptions are given to make the consequent controller design and analysis rigorous

I. Structure of the quad-rotor UAV is rigid, and there are no internal forces or deformations.
II. Structure of the quad-rotor UAV is symmetrical.
III. Center of mass and the origin of body frame coincide.
IV. Earth frame is an inertial frame

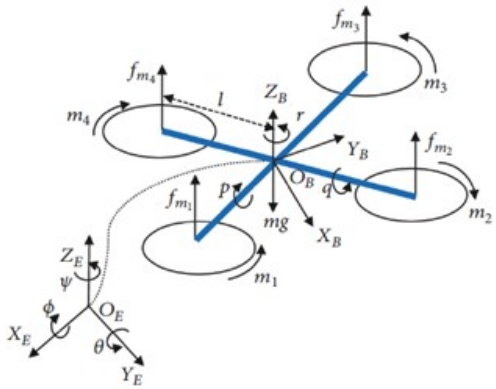


Figure 3.1: The UAV free body Diagram

Figure 3.2: Lift, Roll, Pitch and Yaw Motions of the UAV

For the quad-rotor, it is possible to use two reference systems. The first is fixed and the second is mobile. The fixed coordinate system, called also inertial, is a system where the first Newton's law is considered valid. As fixed coordinate system, we use the O_{NED} systems, where $_{NED}$ is for North-East-Down. As we can observe from the following Figure and its vectors are directed to North, East and to the center of the Earth.

The mobile reference system is coupled with the center of the quad-rotor. Here we called O_{ABC} system, where $_{ABC}$ is for Aircraft Body Center.

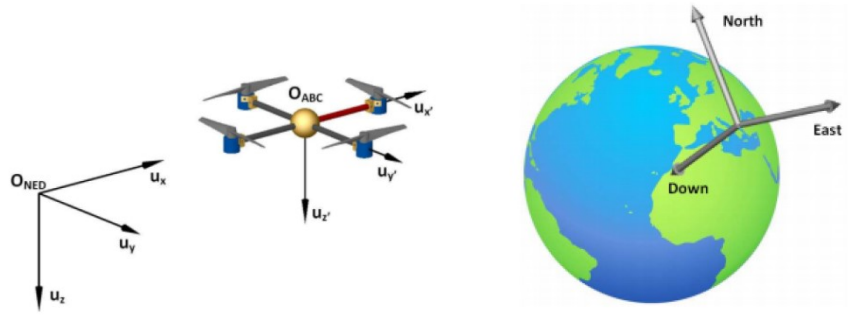


Figure 3.3: The Two Coordinate reference system

3.2. MATHEMATICAL MODELING OF UAV

3.2.1. EULER ANGLE

The Euler angles are three angles introduced by Leonhard Euler to describe the orientation of a rigid body. To describe such an orientation in the 3-dimensional Euclidean space, three parameters are required. They can be given in several ways; we will use xyz Euler angles.

They are also used to describe the orientation of a frame of reference relative to another and they transform the coordinates of a point in a reference frame in the coordinates of the same point in another reference frame [12]. The Euler angles are typically denoted as $\emptyset \in \{-\pi, \pi\}$,

$\theta \in \left\{-\frac{\pi}{2}, \frac{\pi}{2}\right\} and\ \varphi \in \{-\pi, \pi\}$.

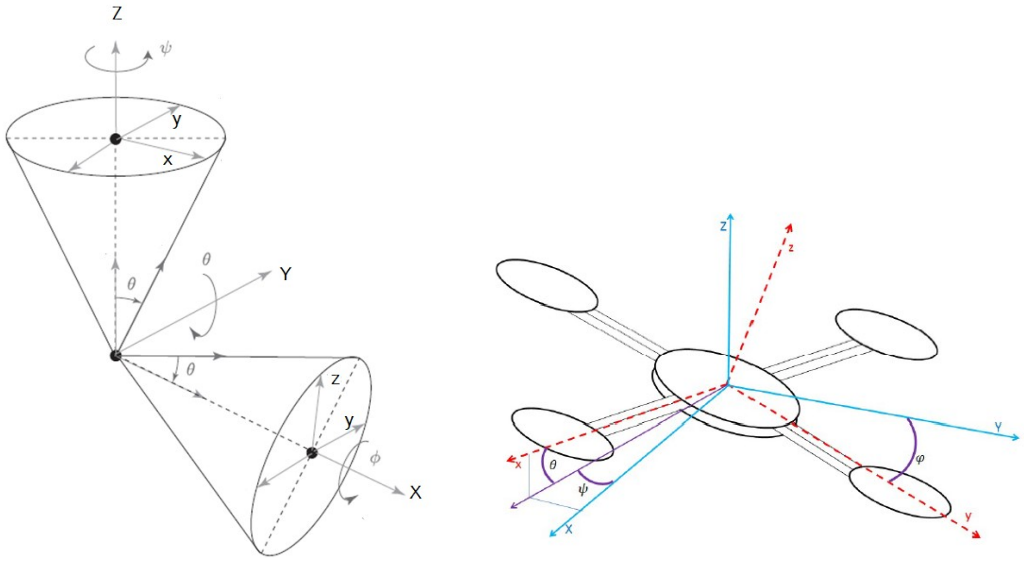


Figure 3.4: The Euler Angles

$$R_{\emptyset} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\emptyset & \sin\emptyset \\ 0 & -\sin\emptyset & \cos\emptyset \end{bmatrix}, R_{\varphi} = \begin{bmatrix} \cos\varphi & 0 & -\sin\varphi \\ 0 & 1 & 0 \\ \sin\varphi & 0 & \cos\varphi \end{bmatrix} and\ R_{\theta} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

3.2.2. REFERENCE FRAME

Before deriving mathematical formulations of the problem, the following reference frames are defined to express state variables.

I. Earth Reference Frame (F_E): A frame locally fixed to a world. Absolute states of the UAV and the target are expressed in this frame. A NED (North-East-Down) frame fixed on the Earth surface is used in this paper. Neglecting rotation and curvature of the Earth, the NED frame can be considered as an inertial frame.

II. UAV Body Frame (F_B): A frame fixed to a UAV body. Its origin is at the center of gravity of the body. The X-axis and the Z-axis are pointing forward and downward, respectively.

III. Frame (F_C): A frame fixed to the camera. Its origin is at the camera's focus, and its Z-axis aligns with the camera's optical axis. The X-axis and the Y -axis are parallel to the 2D image frame axes defined below.

- Image Frame (F_i): A 2D frame defined on the image plane. Its x-axis and y-axis are taken in the direction of image width and length, respectively.

3.2.3. DYNAMICS OF THE UAV

Let X_v, V_v be the position and velocity vectors of the UAV and a_v be its acceleration input, expressed in the inertial frame F_E. Then, the UAV dynamics are given by

$$\dot{X}_v, = V_v \text{ and } \dot{V}_v = a_v \tag{3.2}$$

As such, the rotation matrix from the body frame to the earth frame can be expressed as:

$$R_{BE} = [R_\emptyset R_\theta R_\varphi]^T \tag{3.3}$$

First we address the translational dynamics. Notice that the total lift force generated by the rotors can be given in the body frame as:

$$F_T^b = (0,0,U_1)^T \tag{3.4}$$

With $U_1 \triangleq (F_1 + F_2 + F_3 + F_4)$ being the actual altitude control input, where F_i is the lift force generated by the i^{th} rotor,

The total lift force in the earth frame is then

$$F_l^e = R_{BE} F_T^b \tag{3.5}$$

The air resistance is proportional to the flight speed of the quad-rotor:

$$F_r^e = (k_x \dot{x}, k_y \dot{y}, k_z \dot{z})^T \tag{3.6}$$

The effect of gravity:

$$F_g^e = (0,0,mg)^T \tag{3.7}$$

In the outdoor flight, the quad-rotor is generally exposed to lateral wind gusts. The quad-rotor will be yawed or overturned if it is subjected to the lateral wind gusts. Stating precisely, this leads to additional lateral air flow acting on the propeller, as shown in Figure 7.

$$f_{Ti} = 2\rho A \hat{V} V_p \tag{3.8}$$

Where ρ is the air density and A is the propeller area. V_p is the induced wind speed of the propeller and $\hat{V}$ is the total wind induced speed of the rotor. The relationship between them is given by

$$\hat{V} = [(V_w \cos\alpha + V_p)^2 + (V_w \sin\alpha)^2]^{1/2} \tag{3.9}$$

Since the maximum gust wind UAV encounter is when $\alpha = 90°$ in which the above equation reduces to:

$$\hat{V} = [(V_p)^2 + (V_w)^2]^{1/2} \tag{3.10}$$

Most of the gust winds are due to topological structures in which it will most of the time affects the system during takeoff and landings: For our case we assume a minimum gust conditon in which the total frictional force becomes the air resistance only. For Robustness gust wind is also included on the kinematics equation as additional disturbance[13].

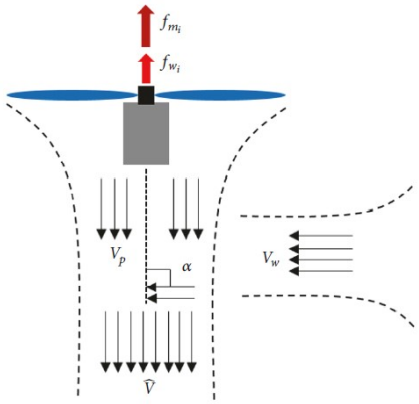


Figure 3.5: The effect of Lateral gust wind on UAV Motion

Using Newton's second law, the translational dynamics of the quad-rotor UAV subjected to external disturbance can be obtained by:

$$\begin{aligned}
\ddot{x} &= \frac{U1(\sin\theta\cos\phi\cos\varphi + \sin\phi\sin\varphi) - k_x\dot{x}}{m} + \frac{d^2x'}{dt^2}, \\
\ddot{y} &= \frac{U1(\sin\theta\cos\phi\sin\varphi - \sin\phi\cos\varphi) - k_y\dot{y}}{m} + \frac{d^2y'}{dt^2}, \\
\ddot{z} &= \frac{U1(\cos\phi\cos\theta) - k_z\dot{z} - mg}{m} + \frac{d^2z'}{dt^2},
\end{aligned} \tag{3.11}$$

Where $\frac{d^2x'}{dt^2}$, $\frac{d^2y'}{dt^2}$ and $\frac{d^2z'}{dt^2}$ are external disturbances respectivly.

Let us call $[x\ y\ z\ \emptyset\ \theta\ \varphi]^T$ the vector containing the linear and angular position of the quad-rotor in the earth frame and $[u\ v\ w\ p\ q\ r]^T$ the vector containing the linear and angular velocities in the body frame.

$$\begin{aligned}
V &= R \cdot v_b \\
\omega &= T \cdot \omega_b \\
T &= \begin{bmatrix} 1 & \sin\emptyset\tan\theta & \cos\emptyset\tan\theta \\ 0 & \cos\emptyset & -\sin\emptyset \\ 0 & \frac{\sin\emptyset}{\cos\theta} & \frac{\cos\emptyset}{\cos\theta} \end{bmatrix}
\end{aligned} \tag{3.12}$$

Where $V = [\dot{x}\ \ \dot{y}\ \ \dot{z}]^T$ $\omega = [\dot{\emptyset}\ \ \dot{\theta}\ \dot{\varphi}\ ]^T$ $V_b = [u\ \ v\ \ w]^T$ $\omega_b = [p\ \ q\ \ r]^T$ and T is angular transformation matrix [14].

From the above equations mathematical operations we will have the following kinematic equations:

$$\dot{x} = w[\sin\emptyset\sin\varphi + \cos\emptyset\cos\varphi\sin\theta] - v[\cos\emptyset\sin\varphi - \cos\varphi\sin\emptyset\sin\theta] + u[\cos\varphi\cos\theta]$$
$$\dot{y} = v[\cos\emptyset\cos\varphi + \sin\emptyset\sin\varphi\sin\theta] - w[\cos\varphi\sin\emptyset - \cos\emptyset\cos\varphi\sin\theta] + u[\cos\theta\sin\varphi]$$
$$\dot{z} = w[\cos\emptyset\cos\theta] - u[\sin\theta] + v[\cos\theta\sin\emptyset]$$
$$\dot{\emptyset} = p + r[\cos\emptyset\tan\theta] + q[\sin\emptyset\tan\theta] \quad (3.13)$$
$$\dot{\theta} = q[\cos\emptyset] - r[\sin\emptyset]$$
$$\dot{\varphi} = r\left[\frac{\cos\emptyset}{\cos\theta}\right] + q\left[\frac{\sin\emptyset}{\cos\theta}\right]$$

The rotational dynamics of the quad-rotor UAV are addressed. Let the vector [*p, q, r*] represent the quad-rotor's angular velocity in the body frame. The angular motion dynamics of the quad-rotor can be expressed as:

$$M_x = J_x\dot{p} + (J_z - J_y)qr,$$
$$M_y = J_y\dot{q} + (J_x - J_z)pr, \quad (3.14)$$
$$M_z = J_z\dot{r} + (J_y - J_x)pq,$$

Where M_x, M_y, and M_z are the components of the resultant torque acting on the quad-rotor in the three directions of *x, y,* and *z*, respectively, and J_x, J_y, and J_z are the inertias of the quad-rotor around *x, y,* and *z*, respectively[15].

Let M_{lx}, M_{ly}, and M_{lz} denote the components of the lifting torque of the quad-rotor in the three directions of *x, y,* and *z*, respectively. Then we will have the following

$$M_{lx} = l(F_1 + F_3 - F_2 - F_4)$$
$$M_{ly} = l(F_4 + F_3 - F_2 - F_1) \quad (3.15)$$
$$M_{lz} = lfm(F_1 + F_4 - F_2 - F_3)$$

Where fm is a scaling factor of force-moment and l is arm length. Likewise the system is subject to gyroscopic torque during flight and can be modeled as:

$$M_{gx} = J_r q\Omega,$$
$$M_{gy} = J_r p\Omega \quad (3.16)$$
$$M_{gz} = 0$$

Where $\Omega = \Omega_2 + \Omega_3 - \Omega_1 - \Omega_4$ *the sum of the angular speed of the propellers*

J_r *is the inertia of each propeller*

The angular motion dynamics of the quad-rotor can be rewritten as follows:

$$\begin{aligned} J_x\dot{p} &= (J_z - J_y)qr - J_r q\Omega + l(F_1 + F_3 - F_2 - F_4) \\ J_y\dot{q} &= (J_x - J_z)pr + J_r q\Omega + l(F_4 + F_3 - F_2 - F_1) \\ J_z\,\dot{r} &= (J_y - J_x)pq + lfm(F_1 + F_4 - F_2 - F_3) \end{aligned} \tag{3.17}$$

From the above kinematics equation will have the following matrix

$$\begin{bmatrix} \dot{\emptyset} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\emptyset\tan\theta & \sin\emptyset\tan\theta \\ 0 & \cos\emptyset & -\sin\emptyset \\ 0 & \frac{\sin\emptyset}{\cos\theta} & \frac{\cos\emptyset}{\cos\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.18}$$

To insure attitude stability of the UAV, during the flight it better to keep the attitude angles very small in which the above equation can be approximated as $\begin{bmatrix} \dot{\emptyset} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} \cong \begin{bmatrix} p \\ q \\ r \end{bmatrix}$

$$\begin{aligned} \ddot{\phi} &= \frac{(J_y - J_z)\dot{\theta}\dot{\varphi} - J_r\dot{\theta}\Omega + l(F_1 + F_3 - F_2 - F_4)}{J_x} + \frac{d^2\emptyset'}{dt^2}, \\ \ddot{\theta} &= \frac{(J_z - J_x)\dot{\emptyset}\dot{\varphi} + J_r\dot{\emptyset}\Omega + l(F_4 + F_3 - F_2 - F_1)}{J_y} + \frac{d^2\theta'}{dt^2}, \\ \ddot{\varphi} &= \frac{(J_x - J_y)\dot{\emptyset}\dot{\theta} + lfm(F_1 + F_4 - F_2 - F_3)}{J_z} + \frac{d^2\varphi'}{dt^2}, \end{aligned} \tag{3.19}$$

Where $d^2\emptyset', d^2\theta'$ *and* $d^2\varphi'$ the effects of the gust wind on the angular accelerations and can be considered as additional external disturbances [16].

3.3. STATE SPACE MODEL

Organizing the state's vector in the following way:

$$x = [x\ y\ z\ p\ q\ r\ \emptyset\ \theta\ \varphi\ u\ v\ w]^T \in R^{12} \tag{3.20}$$

ASMC scheme is proposed for the quad-rotor UAV system with the goal of tracking the desired altitude and attitude in the presence of parametric uncertainties and consistent external disturbance.

Since, the changes of the state ϕ and θ will result in movement in the x and y directions, respectively. Therefore, instead of considering the full six degrees of freedom of the UAV, it is sufficient to control the four degrees of freedom ($\phi, \theta, \varphi,$ and z) only [17].

Then, the dynamic system of the quad-rotor will be:

$$\dot{X} = \begin{cases} \dot{x} \\ \frac{(F_1+F_3+F_2+F_4)(sin\ \theta\ cos\ \emptyset\ cos\ \varphi + sin\ \emptyset\ sin\ \varphi) - kx\ \dot{x}}{m} + \frac{d^2x\prime}{dt^2} \\ \dot{y} \\ \frac{(F_1+F_3+F_2+F_4)(\sin\theta\cos\emptyset\sin\varphi - \sin\emptyset\cos\varphi) - ky\dot{y}}{m} + \frac{d^2y\prime}{dt^2} \\ \dot{z} \\ \frac{(F_1+F_3+F_2+F_4)(\cos\emptyset\cos\varphi) - kz\,\dot{z} - mg}{m} + \frac{d^2z\prime}{dt^2} \\ \dot{\emptyset} \\ \frac{(J_y - J_z)\theta\varphi - J_r\theta\Omega + L(F_1+F_3-F_2-F_4)}{J_x} + \frac{d^2\emptyset\prime}{dt^2} \\ \dot{\theta} \\ \frac{(J_z - J_x)\emptyset\varphi + J_r\emptyset\Omega + L(F_4+F_3-F_2-F_1)}{J_y} + \frac{d^2\theta\prime}{dt^2} \\ \dot{\varphi} \\ \frac{(J_x - J_y)\emptyset\varphi + Lfm(F_1+F_4-F_2-F_3)}{J_z} + \frac{d^2\varphi\prime}{dt^2} \end{cases} \quad (3.21)$$

CHATER 4

CONTROLLER DESIGN AND ANALYSIS

3.1 SLIDING MODE CONTROL (SMC)

In control systems, sliding mode control (SMC) is a nonlinear control method that alters the dynamics of a nonlinear system by applying a discontinuous control signal (or more rigorously, a set-valued control signal) that forces the system to "slide" along a cross-section of the system's normal behavior. The state-feedback control law is not a continuous function of time. Instead, it can switch from one continuous structure to another based on the current position in the state space. Hence, sliding mode control is a variable structure control method. The multiple control structures are designed so that trajectories always move toward an adjacent region with a different control structure, and so the ultimate trajectory will not exist entirely within one control structure. Instead, it will slide along the boundaries of the control structures.

Intuitively, SMC uses practically infinite gain to force the trajectories of a dynamic system to slide along the restricted sliding mode subspace. The main strength of sliding mode control is its robustness. Because the control can be as simple as a switching between two states ("on"/"off" or "forward"/"reverse"), it need not be precise and will not be sensitive to parameter variations that enter into the control channel. Additionally, because the control law is not a continuous function, the sliding mode can be reached in finite time. The method has several advantages such as robustness against matched external disturbances and unpredictable parameter variations [18].

SMC is insensitive to model inaccuracies, fast response and disturbance rejection, simple implementation, minimal tuning requirements and reliable than PID controller; it has a better tracking performance, which fulfills the very important requirement of UAV controller.

In practical applications, sliding mode control may experience undesirable phenomenon of oscillations having finite frequency and amplitude, which is known as `chattering'. It leads to low control accuracy, high wear of moving mechanical parts, and high heat losses in power circuits. There are two reasons which can lead to chattering.

i. Fast dynamics which were neglected in the ideal model. These dynamics with small time constants are usually disregarded in models of servomechanisms, sensors and data processors.

ii. Utilization of digital controllers with finite sampling rate, which causes so called `discretization chatter'.

Theoretically the ideal sliding mode implies infinite switching frequency. Since the control is constant within a sampling interval, switching frequency cannot exceed that of sampling, which lead to chattering as well. To reduce chattering in sliding-mode control, a boundary layer around the switching surface is used, and a continuous control is applied within the boundary [19].

Without loss of generality, we consider the following linear time-invariant (LTI) system:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{4.1}$$

Where $x \in R^n$ is the system state vector, $u \in R^m$ is the control input vector, $A \in R^{nxn}$ and $x \in R^{nxm}$ are constant system matrices. It is assumed that $n > m, B$ is of full rank m, the pair (A, B) is completely controllable, that is, the controllability matrix $[B \quad AB \ldots A^{n-1}B]$ has full rank m. Define a sliding variable vector $s(t) \in R^m$ passing through the state space origin

$$s(t) = Cx(t) \tag{4.2}$$

Where $C \in R^{mxn}$ is the sliding mode parameter vector and $\|\mathrm{CB}\| \neq 0$.

The system above is said to attain a sliding mode surface when the state variable vector reaches and remains on the intersection of the m switching plane variables. The method of equivalent control is a way to determine the system motion restricted to the sliding mode surface $s(t) = 0$. On the sliding mode surface, $s(t) = 0$ and $\dot{s}(t) = 0$, using the above two expressions, we have

$$\dot{s}(t) = C\dot{x}(t) \tag{4.3}$$

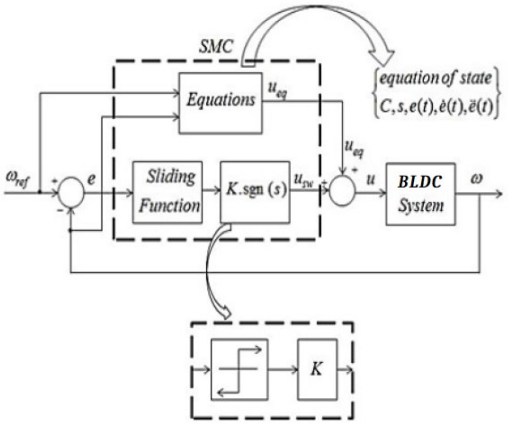
$$C\left(Ax(t) + Bu_{eq}(t)\right) = 0 \tag{4.4}$$

The dynamical behavior of the equivalent system is independent of the control input. Thus, the determination of the matrix C may be completed without prior knowledge of the form of control input. Generally, the sliding parameter C is designed in a manner that the system response confined on the sliding mode surface has a desired behavior such as asymptotic stability and prescribed transient response [20].

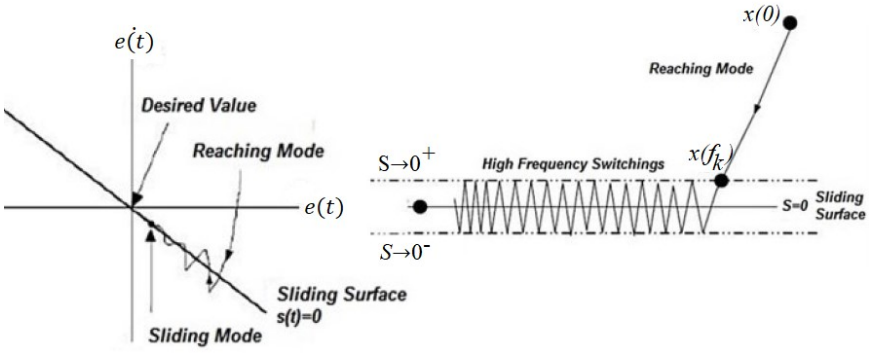
SMC design includes reaching phase and sliding phase. The reaching phase is crucial in the sense that the system dynamics are guaranteed to reach the sliding surface and be retained on it thereafter. In fact, the condition for the switching plane variables to reach the sliding mode surface is a convergence problem. Therefore, the Lyapunov's direct method has been widely used in SMC designs as a stability condition to ensure the convergence of the sliding mode variable onto the sliding surface during the reaching phase. There are three types of reaching rule: Exponential rate, power rate and constant rate. For our case it better to use exponential rate reaching, by adding the proportional rate term hs , the states are forced to approach the switching manifold faster when s is large.

$$\dot{s} = ksign(s) + hs \tag{4.5}$$

Where $k > 0 \; and \; h > 0$



(a)



(b) (c)

Figure 4.1: (a) SMC block diagram (b) sliding surface and reaching (c) chattering effect

4.2. PROPOSED CONTROL SYSTEM DESIGN

The controller consists of mainly altitude and attitude controllers designed using ASMC and modeled external disturbances. On other side a trajectory generator mostly dependent on the image processing unit and feedback signals, which also provide critical information to the ASMC to actuate when required and make corrections for significant error.

An adaptive system is one in which the controller parameters are adjusted automatically to compensate for changing process conditions. The foundation of adaptive control is parameter estimation, which is a branch of system identification. Common methods of estimation include recursive least squares and gradient descent. Both of these methods provide update laws that are used to modify estimates in real-time. Lyapunov stability is used to derive these update laws and show convergence criteria later in this chapter.

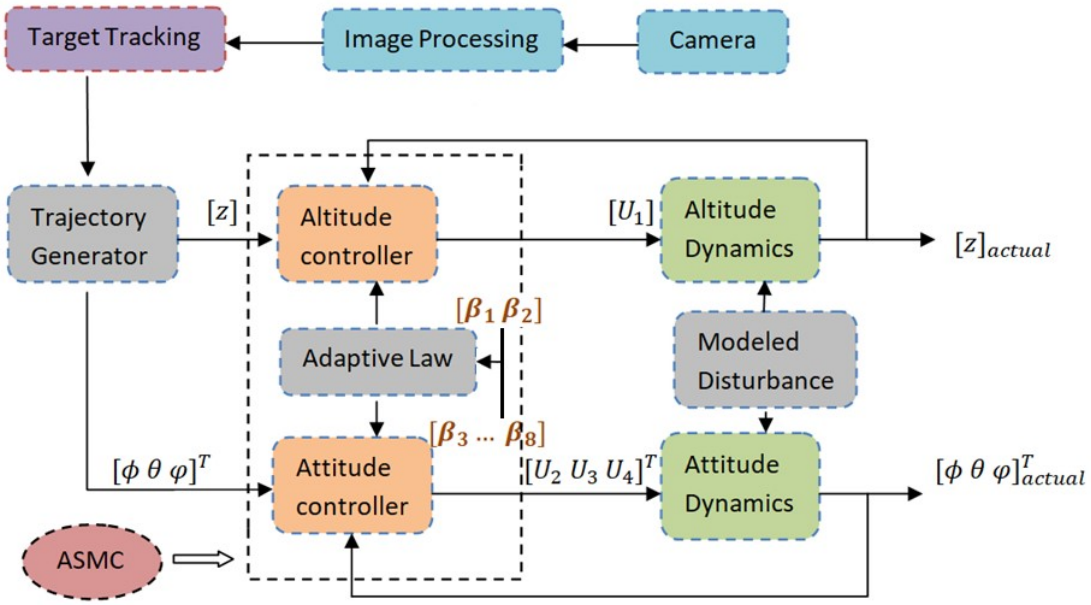


Figure 4.2: Block Diagram of the controller design

The controller adaptive block of the design [21] expected to estimate eight parameters of the UAV for a good performance. Here are the two assumptions

1. All disturbances are uniformly bounded
2. For the stability and avoidance of singularities $\emptyset \; and \; \theta$ should be in between $-\pi/2 \; and \; \pi/2$

The Adaptive law inputs:

Estimation for the Altitude controller:

$\beta_1 \cong mass\ of\ the\ UAV(m\ in\ kg)\ and\ \widehat{\beta_1}\ be\ the\ estimate\ of\ \beta_1$

$\beta_2 \cong air\ resistance\ in\ z-axis(k_z) and\ \widehat{\beta_2}\ be\ the\ estimate\ of\ \beta_2$

Estimation for the attitude controller

$\beta_3 \cong \frac{J_x}{L}, \widehat{\beta_3} \quad \beta_4 \cong \frac{J_y - J_z}{L}, \widehat{\beta_4} \quad \beta_5 \cong \frac{J_y}{L}, \widehat{\beta_5} \quad \beta_6 \cong \frac{J_z - J_x}{L}, \widehat{\beta_6} \quad \beta_7 \cong \frac{J_z}{L}, \widehat{\beta_7} \quad \beta_8 \cong \frac{J_x - J_y}{L}, \widehat{\beta_8}$

(β_i: $inertia\ of\ the\ three\ axies\ and\ the\ effect\ of\ each\ axis\ on\ each\ other\ per\ unit\ length, [\ i = 3 \dots 8]$)

wind Gust involved disturbances are

$$\widetilde{d_z} \approx m d_z, \qquad \widetilde{d_\emptyset} \approx \frac{J_x}{L} d_\emptyset, \qquad \widetilde{d_\theta} \approx \frac{J_y}{L} d_\theta, \qquad \widetilde{d_\varphi} \approx \frac{J_z}{L} d_\varphi$$

Then estimation error becomes

$$\widetilde{\beta_n} = \beta_n - \widehat{\beta_n}\ where\ n = 1,2,3, \dots ,8$$

4.3. ALTITUDE CONTROLLER DESIGN AND STABILITY ANALYSIS

We want altitude output tracking error of the closed loop to converge to zero asymptotically according to ***LaSalle's*** *invariance principle* (i.e. $\lim_{t\to\infty} e_{ALT} = 0$)

Altitude error

$$e_{ALT} = z - z_{actual} \tag{4.6}$$

Here we can assume the sliding surface to be

$$s_1 = c_1 e_{ALT} + \dot{e}_{ALT} \tag{4.7}$$

Let us address the control law U_1

$$U_1 = Gravity + Drag + Torque\ coupling\ and\ Inertia + Disturbance + lift\ to\ desired\ height$$

$$U_1 = \frac{1}{\cos\theta \cos\emptyset}\left(\widehat{\beta_1}\left(c_1 \dot{e}_{ALT} + \ddot{Z} + g\right) + \widehat{\beta_2}\dot{z} + \widetilde{d_z} sgn(s_1) + h_1 s_1\right) \tag{4.8}$$

$where\ Sgn(.) is\ the\ signum\ function\ and\ h_1\ is\ a\ postive\ constant$

Now define the ***Lyapunov*** function for the ASMC stability analysis

$$V_z = \frac{1}{2} m s_1^2 + \frac{1}{2}\left(\tilde{\beta}_1^2 + \tilde{\beta}_2^2\right) \tag{4.9}$$

Which

$$\dot{V}_z = m s_1 \dot{s}_1 + \left(\dot{\widetilde{\beta_1}}\tilde{\beta}_1 + \dot{\widetilde{\beta_2}}\tilde{\beta}_2\right) \tag{4.10}$$

Since

$$\dot{s}_1 = c_1 \dot{e}_{ALT} + \ddot{e}_{ALT} \tag{4.11}$$

Implies $\dot{s}_1 = c_1\dot{e}_{ALT} + \ddot{z} - \ddot{Z}$

$$\dot{V}_z = ms_1(c_1\dot{e}_{ALT} + \ddot{z} - \ddot{Z}) + \left(\dot{\widetilde{\beta_1}}\tilde{\beta}_1 + \dot{\widetilde{\beta_2}}\tilde{\beta}_2\right), where\ \ddot{Z} = U_1\cos\theta\cos\emptyset - k_z\dot{z} - mg + \widetilde{d_z}$$

$$= s_1(m(c_1\dot{e}_{ALT} + \ddot{z}) - (U_1\cos\theta\cos\emptyset - k_z\dot{z} - mg + \widetilde{d_z})) + \left(\dot{\widetilde{\beta_1}}\tilde{\beta}_1 + \dot{\widetilde{\beta_2}}\tilde{\beta}_2\right)$$

$$= m(s_1(c_1\dot{e}_{ALT} + \ddot{z} + g)) + s_1k_z\dot{z} - s_1(U_1\cos\theta\cos\emptyset) - s_1\widetilde{d_z} + \left(\dot{\widetilde{\beta_1}}\tilde{\beta}_1 + \dot{\widetilde{\beta_2}}\tilde{\beta}_2\right)$$

$$= \tilde{\beta}_1(s_1(c_1\dot{e}_{ALT} + \ddot{z} + g)) + \tilde{\beta}_2s_1\dot{z} - s_1(U_1\cos\theta\cos\emptyset) - s_1\widetilde{d_z} + \left(\dot{\widetilde{\beta_1}}\tilde{\beta}_1 + \dot{\widetilde{\beta_2}}\tilde{\beta}_2\right)$$

$$= \tilde{\beta}_1(s_1(c_1\dot{e}_{ALT} + \ddot{z} + g)) + \tilde{\beta}_2s_1\dot{z} - s_1\left(\widehat{\beta_1}(c_1\dot{e}_{ALT} + \ddot{z} + g) + \widehat{\beta_2}\dot{z} + \widetilde{d_z}sgn(s_1) + h_1s_1\right)$$

$$- s_1\widetilde{d_z} + \left(\dot{\widetilde{\beta_1}}\tilde{\beta}_1 + \dot{\widetilde{\beta_2}}\tilde{\beta}_2\right)$$

$$= -s_1\left(\widetilde{d_z}sgn(s_1) + h_1s_1\right) - s_1\widetilde{d_z} + \left(\dot{\widetilde{\beta_1}}\tilde{\beta}_1 + \dot{\widetilde{\beta_2}}\tilde{\beta}_2\right)$$

Finally our mass is constant $\dot{\widetilde{\beta_1}} = 0$ and k_z is constant, that means $\dot{\widetilde{\beta_2}} = 0$

$$\dot{V}_z = -s_1\left(\widetilde{d_z}sgn(s_1) + h_1s_1\right) - s_1\widetilde{d_z} \quad (4.12)$$

$$\dot{V}_z = -\widetilde{d_z}|s_1| - h_1{s_1}^2 - s_1\widetilde{d_z}, \quad (4.13)$$

$\dot{V}_z \le -h_1{s_1}^2 \le 0$ Hence $\dot{V}_z = 0$ when $s_1 = 0$

Therefore the output tracking error of the closed loop is converging to zero asymptotically [22].

4.4. ATTITUDE CONTROLLER DESIGN AND STABILITY ANALYSIS

In aviation, attitude is the orientation of the aircraft in space, measured by the values of these three angles: pitch (around the lateral axis), roll (around the longitudinal axis), and yaw (around the vertical axis).

The yaw controller primary source for trajectory generation is the camera onboard unless a manual mode is required. The controller design for all the three axes is the same except some variable changes. Implies the design and stability analysis is almost similar.

4.4.1. ROLL CONTROLLER DESIGN AND STABILITY ANALYSIS

Roll angle tracking error

$$e_{ROL} = \emptyset - \emptyset_{actual} \quad (4.14)$$

Here we can assume the sliding surface to be

$$S_2 = C_2 e_{ROL} + \dot{e}_{ROL} \tag{4.15}$$

The control law U_2

$$U_2 = Torque\ Coupling - Inertial\ difference\ of\ y\ and\ z\ axies + Disturbance + Desired\ ROLL$$

$$U_2 = \hat{\beta}_3\left(C_2\dot{e}_{ROL} + \ddot{\emptyset}\right) - \hat{\beta}_4\dot{\theta}\dot{\varphi} + \widetilde{d_\emptyset} sgn(s_2) + h_2 s_2 \tag{4.16}$$

where $Sgn(.)$ *is the signum function and* h_2 *is a postive constant*

Likewise we want attitude output tracking error of the closed loop to converge to zero asymptotically according to ***LaSalle's*** *invariance principle* (i.e. $\lim_{t\to\infty} e_{ROL} = 0$)

Our chosen ***Lyapunov*** function for the ASMC

$$V_\emptyset = \frac{J_x}{2L} s_1^2 + \frac{1}{2}\left(\tilde{\beta}_3^2 + \tilde{\beta}_4^2\right)$$

Which

$$\dot{V}_\emptyset = \frac{J_x}{L} s_1 \dot{s}_1 + \left(\dot{\tilde{\beta}}_3\tilde{\beta}_3 + \dot{\tilde{\beta}}_4\tilde{\beta}_4\right) \tag{4.17}$$

Since $\dot{s}_2 = C_2\dot{e}_{ROL} + \ddot{e}_{ROL}$

Implies $\dot{s}_2 = C_2\dot{e}_{ROL} + \ddot{\emptyset} - \ddot{\bar{\emptyset}}$

$$\dot{V}_\emptyset = s_2(C_2\dot{e}_{ROL} + \ddot{\emptyset} - \ddot{\bar{\emptyset}}) + \left(\dot{\tilde{\beta}}_3\tilde{\beta}_3 + \dot{\tilde{\beta}}_4\tilde{\beta}_4\right), \text{where } \ddot{\bar{\emptyset}} = U_2 + \tilde{\beta}_4\dot{\theta}\dot{\varphi} + \widetilde{d_\emptyset}$$

$$= s_2(C_2\dot{e}_{ROL} + \ddot{\emptyset}) - s_2(U_2 + \tilde{\beta}_4\dot{\theta}\dot{\varphi} + \widetilde{d_\emptyset}) + \left(\dot{\tilde{\beta}}_3\tilde{\beta}_3 + \dot{\tilde{\beta}}_4\tilde{\beta}_4\right)$$

$$= s_2(C_2\dot{e}_{ROL} + \ddot{\emptyset}) - s_2(\tilde{\beta}_3\left(C_2\dot{e}_{ROL} + \ddot{\emptyset}\right) - \tilde{\beta}_4\dot{\theta}\dot{\varphi} + \widetilde{d_\emptyset} sgn(s_2) + h_2 s_2 + \tilde{\beta}_4\dot{\theta}\dot{\varphi} + \widetilde{d_\emptyset}) + \left(\dot{\tilde{\beta}}_3\tilde{\beta}_3 + \dot{\tilde{\beta}}_4\tilde{\beta}_4\right)$$

$$= -s_2\left(\widetilde{d_\emptyset} sgn(s_2) + h_2 s_2\right) - s_2\widetilde{d_\emptyset} + \dot{\tilde{\beta}}_3\tilde{\beta}_3 + \dot{\tilde{\beta}}_4\tilde{\beta}_4, \quad \text{Where } \dot{\tilde{\beta}}_3 = 0 \text{ and } \dot{\tilde{\beta}}_4 = 0$$

$$\dot{V}_\emptyset = -s_2\left(\widetilde{d_\emptyset} sgn(s_2) + h_2 s_2\right) - s_2\widetilde{d_\emptyset} \tag{4.18}$$

$$\dot{V}_\emptyset = -\widetilde{d_\emptyset}|s_2| - h_2 {s_2}^2 - s_2\widetilde{d_\emptyset}, \tag{4.19}$$

$\dot{V}_\emptyset \le -h_2 {s_2}^2 \le 0$ Hence $\dot{V}_\emptyset = 0$ when $s_2 = 0$

Here again the output tracking error of the closed loop is converging to zero asymptotically.

Similar to designing the control and adaptive laws for the altitude and roll blocks design, for the pitch and yaw blocks can be derived as well and also the stability analysis is the same [23].

4.4.2. PITCH CONTROLLER DESIGN

$$U_3 = Torque\ coupling - Inertial\ difference\ of\ x\ and\ z\ axies + Disturbance + Desired\ PITCH$$

$$U_3 = \widehat{\beta_5}\left(C_3\dot{e}_{PTH} + \ddot{\theta}\right) - \widehat{\beta_6}\dot{\phi}\dot{\varphi} + \widetilde{d_\theta}sgn(s_3) + h_3 s_3 \quad (4.20)$$

Where the sliding surface is:

$$s_3 = C_3 e_{PTH} + \dot{e}_{PTH} \quad (4.21)$$

4.4.3. YAW CONTROLLER DESIGN

$$U_4 = Torque\ coupling - Inertial\ difference\ of\ x\ and\ y\ axies + Disturbance + Desired\ YAW$$

$$U_4 = \widehat{\beta_7}(C_4\dot{e}_{YAW} + \ddot{\varphi}) - \widehat{\beta_8}\dot{\theta}\dot{\phi} + \widetilde{d_\varphi}sgn(s_4) + h_4 s_4 \quad (4.22)$$

Where the sliding surface is:

$$s_4 = C_4 e_{YAW} + \dot{e}_{YAW} \quad (4.23)$$

4.5. MODELING THE UAV BLDC MOTORS

It necessitates four motors for a quad-rotor. The motor is the device that converts electrical power output from the battery into mechanical power. The mechanical power turns the propellers and generates the force needed by the UAV to take off and fly. For this quad-rotor, 4 BLDC motors are used into the design. This type of motor has 2 constants, which are K_m and K_v values. K_m is the motor constant and it is a ratio of the motors torque and the square root of the resistive power loss. K_v is known as the motor's velocity constant and it measured in RPM (revolutions per minute) per volt. This last constant is the ratio of the unloaded motor's RPM to the peak voltage output. In other words, if a motor has a K_v value of 1500 RPM/volt and a supply of 9 volts, the nominal speed would be 13,500 RPM [24].

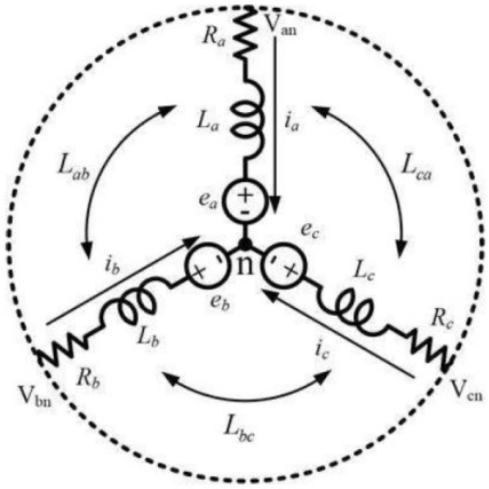


Figure 4.3: BLDC motor system model.

Since the motor has the same material parameters for all the three windings:

$$[R_a = R_b = R_c = R, L_a = L_b = L_c = L, L_{ab} = L_{bc} = L_{ca} = M \text{ and } e_a = e_b = e_c = E]$$

$$T_e = J\frac{d\omega}{dt} + B\omega + T_L \tag{4.24}$$

$$V_i = V_{an} = V_{bn} = V_{cn} = Ri + (L - M)\frac{di}{dt} + E, \text{ where } L \gg M$$

$$\text{Hence, } T_L = K_m i \text{ and } Back\ emf\ E = K_v\omega$$

$$\text{Implies, } V_i = Ri + L\frac{di}{dt} + K_v\omega \text{ and } T_e = J\frac{d\omega}{dt} + B\omega + T_L$$

Applying laplaces transform on both:

$$V_i(s) = RI(s) + LI(s)S + K_v\omega \text{ and } T_e = (JS + B)\omega + T_L$$

$$I(s) = \frac{V_i(s) - K_v\omega}{R + LS}$$

$$\omega(S) = \frac{K_m I(s) - T_L}{JS + B} = \dot{\theta}$$

$$\frac{d}{dt}\begin{bmatrix}\omega \\ I\end{bmatrix} = \begin{bmatrix} -B/J & K_v/J \\ -K_m/L & -R/L \end{bmatrix}\begin{bmatrix}\omega \\ i\end{bmatrix} + \begin{bmatrix}0 \\ 1/L\end{bmatrix}V_i$$

$$y = [1 \quad 0][\omega \quad i]^T \tag{4.25}$$

Here the input a voltage source and the out put is thr required angular velocity, such that the following transfer function block diagram can express the input output characterstics.

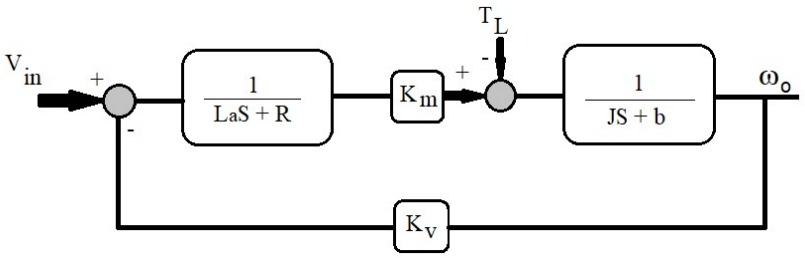


Figure 4.4: BLDC motor system block diagram

Parameter	Used BLDC value
J – moment of inertia of the motor	0.01kgm^2
B – Damping friction	0.1N.m.sec
L_a – Armature Inductance	0.5H
K_m – Torque constant	0.01N.m/A
K_v – Voltage constant	0.01V/rad/sec
R – Terminal resistance	1Ω

Table 4.1: A2212/6T 1000Kv BLDC motor datasheet [25]

A22126T incorporate of 22mm motor diameter, a motor shaft height of 12mm and 6 revolutions per pole. Since it is a 3-phase motor, low cost, high speed and efficient BLDC makes it one of the most popular motor specially designed for quad-rotors.

BLDC motor expected to have electrical parameters changes as the temperature variations, current, and voltage fluctuations, time-varying loading conditions, driving and operating conditions.

$$\frac{\omega(S)}{V_i(S)} = \frac{K}{(JS+B)(LS+R)+K^2}, where\ K = K_m = K_V \tag{4.26}$$

$$\frac{\omega}{V_i} = \frac{0.01}{0.005S^2 + 0.06S + 0.1001}, \text{applying inverse Laplace}$$

$$0.005\ddot{\omega}(t) + 0.06\dot{\omega}(t) + 0.1001\omega(t) = 0.01V(t)$$

$$\ddot{\omega}(t) = 2V(t) - 12\dot{\omega}(t) - 20.02\omega(t)$$

$$Let\ x_1(t) = \omega(t) = y\ such\ that\ \ x_2(t) = \dot{x_1}(t) = \dot{\omega}(t)\ and\ \ \ddot{x_1}(t) = \ddot{\omega}(t) = \dot{x_2}(t)$$

$$\dot{x_2}(t) = 2V(t) - 12\,x_2(t) - 20.02x_1(t) \tag{4.27}$$

Now the sliding surface $S = Ce + \dot{e} = C(\omega_{req} - \omega_{act}) + (\dot{\omega_{req}} - \dot{\omega_{act}})$

Here the objective is to bring Sliding surface $S = 0\ and\ \dot{S} = 0$,

Hence, $\dot{S} = C\dot{e} + \ddot{e} = 0$

$$C\left(\dot{\omega_{req}} - \dot{\omega_{act}}\right) + \left(\ddot{\omega_{req}} - \ddot{\omega_{act}}\right) = 0, where\ \omega_{req} is\ constant\ \dot{\omega_{req}} = \ddot{\omega_{req}} = 0$$

$$C\left(\dot{\omega(t)}\right) = -(2V(t) - 12\ \dot{\omega(t)} - 20.02\omega(t))$$

$$2V(t) = (12 - C)\dot{\omega(t)} + 20.02\omega(t)$$

$$V(t) = U = \frac{1}{2}\left[(12 - C)\dot{\omega(t)} + 20.02\omega(t)\right] \quad (4.28)$$

The control input becomes:

$$U_{SMC} = U_{eqv} + U_{SW}, \qquad where\ U_{SW} = Ksgn(S)\ and\ U_{eqv} = U$$

$$U_{SMC} = 0.5\left[(12 - C)\dot{\omega(t)} + 20.02\omega(t)\right] + Ksgn(S) \quad (4.29)$$

A careful selection of the parameters $C\ and\ K$ will bring the system to the desired output with minimum chattering effect and shorter settling time.

The above equation is the standard form of SMC controller for BLDC. For specific case of altitude controller design, small modification can be applied to fit the system requirement.

Altitude error

$$e_{ALT} = z - z_{actual},\ \ implies\ \ \dot{e}_{ALT} = -\dot{Z}$$

Here we can assume the sliding surface to be

$$s_1 = c_1 e_{ALT} + \dot{e}_{ALT}$$

$$U_1 = \frac{1}{\cos\theta\, c\quad \emptyset}\left(\widehat{\beta_1}\left(c_1\dot{e}_{ALT} + \ddot{Z} + g\right) + \widehat{\beta_2}\dot{z} + \widetilde{d_z}sgn(s_1) + h_1 s_1\right)$$

$$U = A\left[g - c_1\dot{Z} - \ddot{Z}\right] - B\dot{Z} + Psgn(s_1) + Qs_1 \quad (4.30)$$

Qs_1 Is introduced for chattering reduction, in which the above equation for altitude controller input is equivalent to the SMC equation derived earlier.

4.6. FORCE AND TORQUE DECOUPLING ANALYSIS

When considering a level flight with both translation and rotational maneuvers, the force produced by the motors will split in to translational and lift force. In which this leads to height loss and fluctuating rotational angle and this phenomena is called force and torque coupling effects [26].

To avoid this kind of error and discrepancy force decoupling technique is introduced.

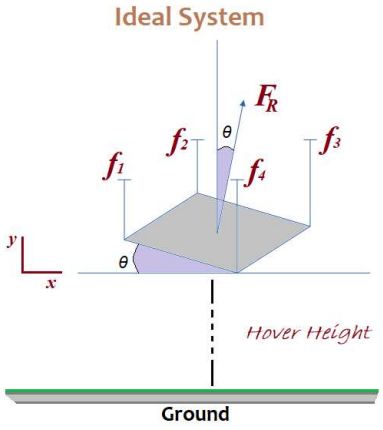


Figure 4.5: Ideal UAV system representation

$$f_1 = f_2 \; and \; f_3 = f_4$$

$$f_1 = f_{11} - f = f_2 = f_{22} - f \; and \; f_3 = f_{33} + f = f_4 = f_{44} + f$$

$$(f_{11} = f_{22} = f_{33} = f_{44} = F)$$

$$F_R = f_1 + f_2 + f_3 + f_4$$

$$F_R = f_{11} - f + f_{22} - f + f_{33} + f + f_{44} + f$$

$$F_R = F - f + F - f + F + f + F + f$$

$$F_L = F_R \cos\theta = mg$$

$$F_R = 4F = mg$$

$$F = \frac{1}{4}mg \tag{4.31}$$

Let's assume symmetry of level flight, in which every lateral motors supports a half of the UAV weight:

$$\frac{1}{2}F_R \cos\theta = (2F - f)\cos\theta = \frac{1}{2}mg$$

$$(2F - f)\cos\theta = \frac{1}{2}mg$$

$$\left(\frac{1}{2}mg - f\right)\cos\theta = \frac{1}{2}mg$$

$$f = \frac{1}{2}mg\left(\frac{\cos\theta -}{\cos\theta}\right) \tag{4.32}$$

The above force f is the required force to add and subtract from the hovering force to maintain level and constant angle maneuver flight for the required time.

Take for example 10^o Roll angle for 2.2kg UAV which results:

$$f = -0.1664N \text{ And } F = 5.3855N$$

To make a level flight the UAV needs about a total of $4F = F_L = 21.582N$.

10^o Roll angle for 2.2kg UAV gives a translation of

$$F_T = F_R \sin\theta = 4F \sin\theta = 4x(0.9369) = 3.7477N.$$

Consideration of motion in 1D:

$$S = V_0 t + \frac{1}{2}at^2, \text{ where } V_0 = 0m/s \tag{4.33}$$

θ	$F_R(N)$	$a(m/sec^2)$	$S(m)$
5^0	1.88	0.855	$0.428t^2$
10^0	3.75	1.705	$0.853t^2$
15^0	5.59	2.541	$1.271t^2$
20^0	7.38	3.355	$1.677t^2$
25^0	9.12	4.145	$2.073t^2$
30^0	10.79	4.905	$2.453t^2$

Table 4.2: Bank angle versus Decoupling Force required and Lateral displacement.

4.7 FLIGHT CONTROLLER ARCHITECTURE

The flight controller engages after the mission control activates the start of the UAV motors, at which the navigation sequence kicks in making a trajectory to the first segment center C_1. After reaching C_1, the system checks position accuracy; if required the system apply adjustments to correct the error and the image processing initiated for target identification and tracking. And it will keep looping in the operational area center positions C_i following a continuous check on each operational area segment until potential target is identified. In the mean time the system learns the correction amount to adapt the atmospheric conditions.

Anytime mission control operator considers aborting for any reason and if the system identifies any malfunctions or emergency the mission will be aborted for good and the UAV will navigate back to home base. Both manual and automatic operations are available on user discretion.

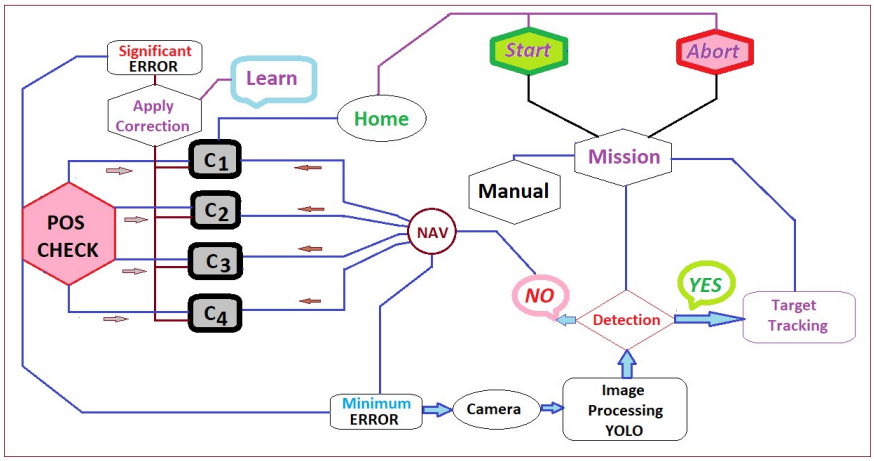


Figure 4.6: Flight Controller architecture

4.7.1 MOTOR MIXING ALGORITHM

Generally ALT controller expects to have three distinct maneuvers of **CLIMB, HOVER** and **DESCENT** implies the four motors have the same output. During climb and descent no other maneuvers are allowed. Power to respective motors will be the input sum of the require maneuvers and minimum hovering speed [27].

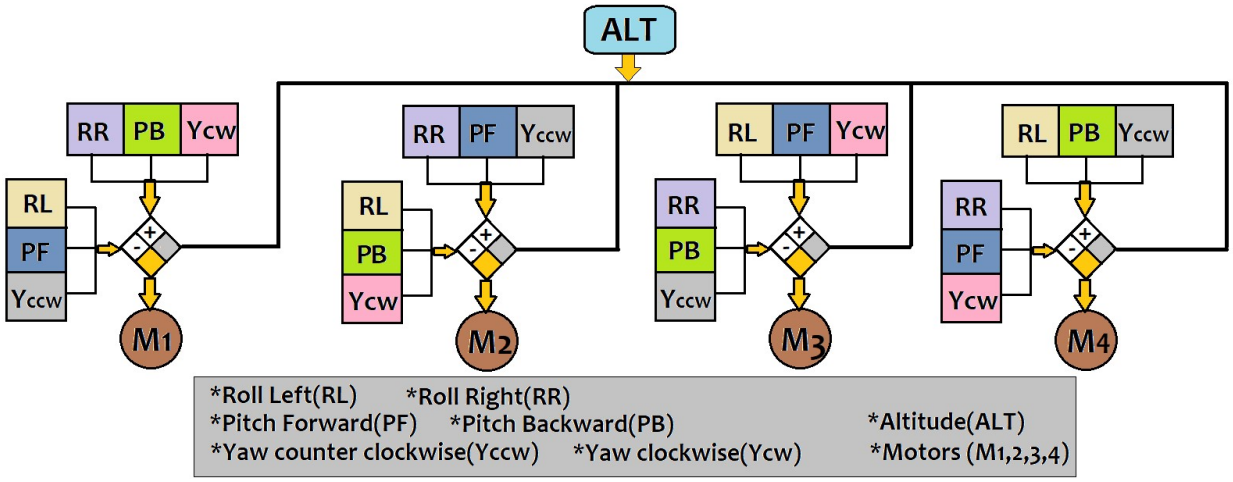


Figure 4.7: Motor mixing algorithm block diagram

From simulation the following datum table was used:

Hovering Height in meters	Motor Speed in standard units	Climb Speed and minimum hovering
4m	52.5%	1.05
8m	57.5%	1.15
12m	65.0%	1.30
16m	72.5%	1.45
20m	77.5%	1.55

Table 4.3: Simulation hovering datum

4.7.2 TRAJECTORY GENERATION AND NAVIGATION PLANNING

The Earth is not exactly a Sphere; it is an "Oblate Spheroid".

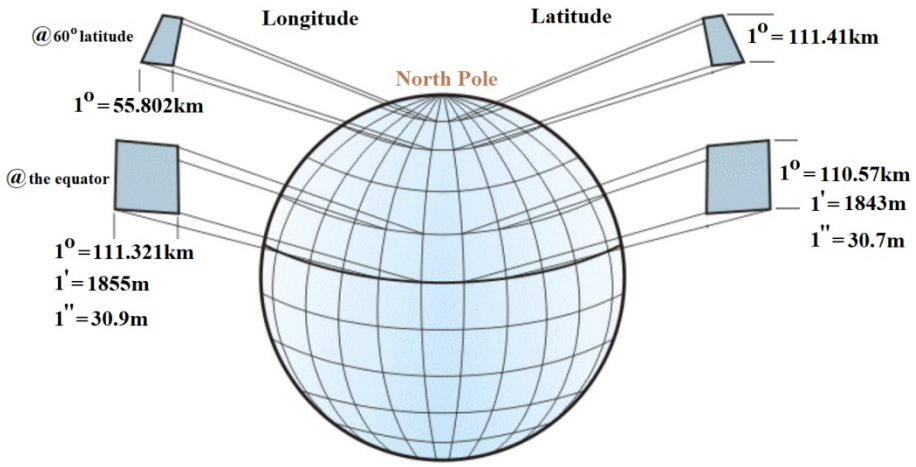


Figure 4.8: Latitude and Longitude Interpretation

Since Addis Ababa is around 10 degrees north latitude, it can be considered as equator which makes insignificant error for navigation.

4.7.2.1. PHOTO-GRAMMETRY

Photo-grammetry is the science of making measurements from photographs. This includes calculating distances and lengths, objects heights and area measurements. If the scale of an

image is known, distances or lengths of objects can be easily calculated by measuring the distance on the photo and multiplying it by the scale factor.

The scale of a photograph is determined by the focal length of the camera and the flying height above the ground. The *focal length* is the distance from the middle of the camera lens to the focal plane. Focal length is precisely measured when cameras are calibrated and is typically expressed inn millimeters (mm). The focal length of a lens determines the magnification and the angle of the light ray. The longer the focal length implies the greater the magnification of the image. Short focal length lenses cover larger areas. The area captured by a camera is known as the Field of View (FOV), which is typically expressed in degrees. Field of View is a function of the focal length of the lens and the size (sometimes called format) of digital sensors.

The scale of a photo is equal to the ratio between the camera's focal length and the UAV altitude above the ground level (AGL) being photographed. If the focal length and flying altitude above the surface is known, the scale can be calculated using the following formula:

$$Scale = \frac{Camera\ sensor\ pixel}{Ground\ distance\ pixel} = \frac{f}{H} \tag{4.34}$$

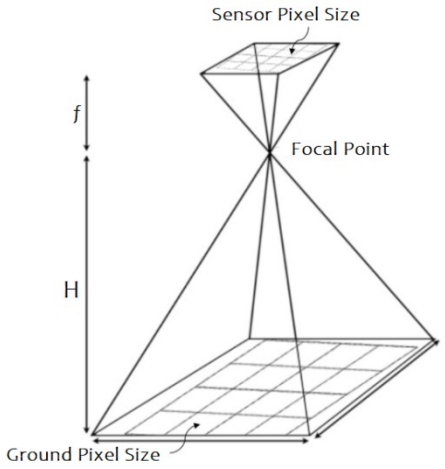


Figure 4.9: Camera Image to Actual area Mapping

Ground Sampling Distance (GSD) refers to how big each pixel is on the ground. This linear measurement represents the ground width of a pixel and is typically expressed in meters [28].

$$GDS = \frac{sensor\ pixel}{Ground\ distance\ pixel} = \frac{f}{H} \tag{4.35}$$

To make it more practical and reasonable, choosing a suitable camera is important; a perfect choice considering cost and weight will be 600TVL 170 degree mini FPV AV camera having Sensor pixel length of 1.8mm 3.5964 X 2.7084mm and 1280 X 960 resolutions, in which;

$$GDS = \frac{3.6x10^{-3}x\ 10m}{1.8x10^{-3}} = 20m$$

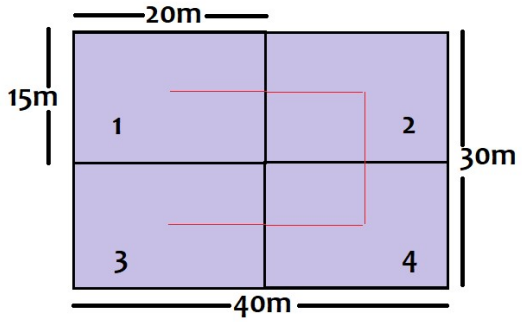


Figure 4.10: A four segment operational area

A displacement of 40m longitude is about 1.64sec and 30m latitude makes 1 sec. let's consider a target dimension of 30cm x 30cm which implies 10x10 pixel size on camera image. From this totally expected targets operating 30m by 40m area will be 12551.

4.7.2.2. HILBERT CURVE

Is a type of space-filling curves that folds one dimensional axis into a two dimensional space, but still keeps the locality. It has advantages to visualize data with long axis in following two aspects:

I. Greatly improve resolution of the visualization from n to √n;
II. Easy to visualize clusters because generally data points in the axis will also be close in the 2D space.

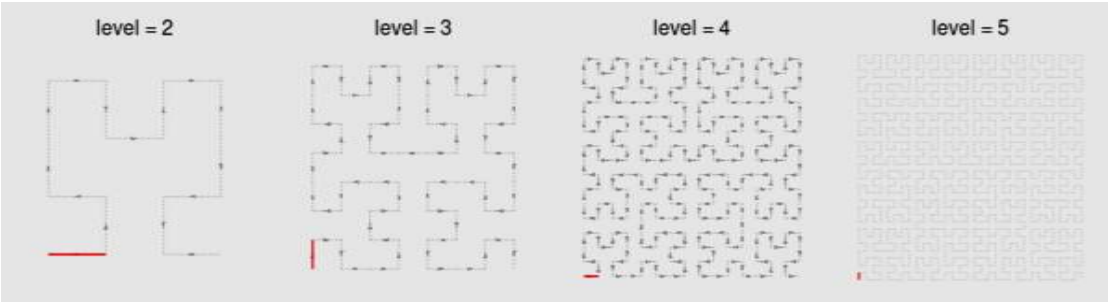
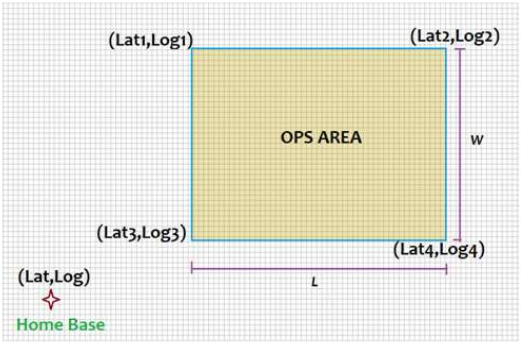
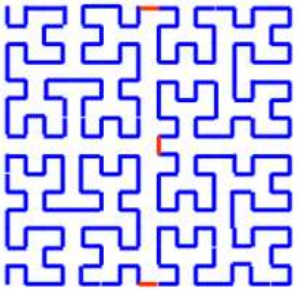


Figure 4.11: plots show Hilbert curves with level 2, 3, 4 and 5

As shown in the above plots, as level increases, the length of the curve becomes longer and the curve folds more densely. The number of segments (one segment is marked in red) on the Hilbert curve is $4^{\text{level}} - 1$. If a Hilbert curve with level 11 is used it can map to human chromosome 1, the resolution would be **249250621** or 4^{11} (approximately 59bp per segment). Hilbert curve folds one-dimensional axis into a two-dimensional space while still preserves the locality of the data points [29].





(a) (b)

Figure 4.12: (a) Operational area (b) Expected flight path Hilbert curve

User defined home base $[Lat, Log]$ and operational area of:

$$OPS\ COORDs = \begin{bmatrix} [Lat_1, Log_1] & [Lat_2, Log_2] \\ [Lat_3, Log_3] & [Lat_4, Log_4] \end{bmatrix} \quad (4.36)$$

Depends on the camera selected (will be discussed in the next chapter) a 10m hover height UAV which will have ground coverage of 15m x 20m; implies the operational area defined by the user should be segmented with respect to this dimensions specified.

$$length(l) = Log_2 - Log_1 \text{ And } width(w) = Lat_3 - Lat_1 \quad (4.37)$$

The actual number of ground segment of a single unit of 15m x 20m will be $a\ x\ b$ segment operational area:

$$\text{Hence, } a = \frac{l}{20}\ and\ b = \frac{w}{15} \quad (4.39)$$

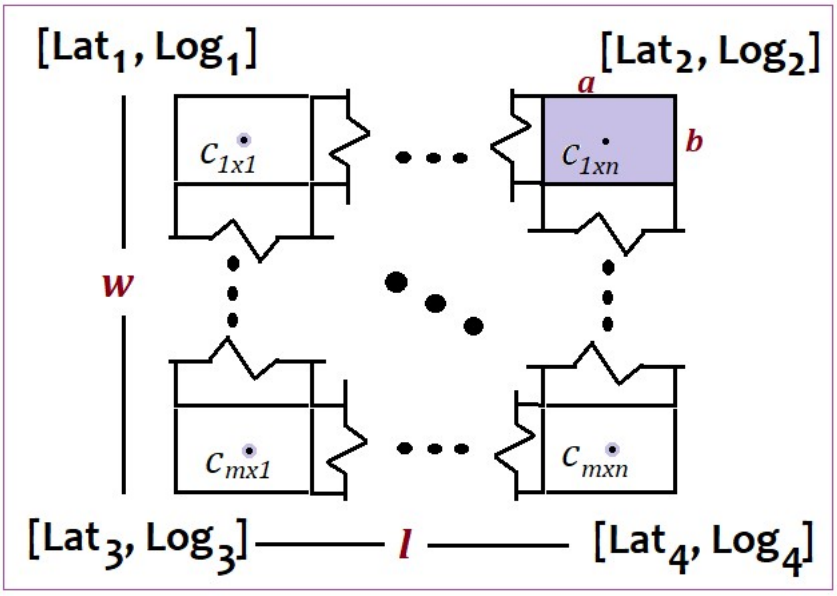


Figure 4.13: Segmentation using Hilbert curve technique

The Quad-rotor is always considered to hover exactly over the segment center C_i, where the next center will be either $a\ or\ b$ distance away, which makes$[m\ x\ n]$ matrix. These translations will be converted to arc-length then added to the coordinate system to get the new center of the segment.

$$Lat_{1'} = Lat_1 + \frac{b}{2}\ and\ Log_{1'} = Log_1 + \frac{a}{2}$$

Where $C_1 = [Lat_{1'}, Log_{1'}]$

$$C_i = \begin{bmatrix} C_{11} & C_{12} & \dots & C_{1n} \\ C_{21} & C_{22} & \dots & C_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{m1} & C_{m2} & \dots & C_{mn} \end{bmatrix} \tag{4.40}$$

The next center coordinate can be calculated just by adding $a\ and\ b$ respectively.

$$C_{21} = [Lat_{1'}, Log_{1'} + a]$$

$$C_{1n} = [Lat_{1'}, Log_{1'} + (n-1)a]$$

$$C_{n+1} = [Lat_{1'} + a, Log_{1'}]$$

$$C_{2n} = [Lat_{1'} + b, Log_{1'} + (n-1)a]$$

$$C_{mn} = [Lat_{1'} + (m-1)b, Log_{1'} + (n-1)a]$$

$$C_{mn} = [Lat_1 + \frac{b}{2} + (m-1)b, Log_1 + \frac{a}{2} + (n-1)a]$$

$$C_{mn} = \left[Lat_1 + \frac{(2m-1)b}{2}, Log_1 + \frac{(2n-1)a}{2}\right] \tag{4.41}$$

This computed equation can be also called the refined ***Mercator*** projection. Here a database will be assigned after calculating all segment centers and used to cross-check with GPS later on for accuracy. Generally the navigation system is expected to follow the Hilbert curve path of finite segment center to center flight trajectory.

Now let us consider to calculate target position with respect to Quad-rotor frame/camera using Hilbert curve. The gray scale of the image frames should be manipulated and compared to get the target pixel coordinate using the Hilbert curve. In which the UAV always try to keep the target at the center of camera image frame with additional help of the optical flow vector.

Case 1:

When the target is stationary

- On this scenario the Hilbert coordinate expected to be constant and the UAV should keep the target on the center of the optical flow and orbit around it until further change of momentum by the target. The trajectory generator will command the attitude controller to give continues coordinated roll and yaw inputs until the optical flow vector detects any change of momentum by the target.

Case 2:

When the target is on motion and with no change of direction of motion:

- Here the UAV tries to follow the moving target with a constant speed and distance while keeping it at the center of the optical flow. Translational displacement is required to follow the target, which can be accomplished by generating translational input forces either on the x or y axis depending up on the target moving direction.

Case 3:

When the target is on motion and with change of direction of motion:

- This final scenario is handled with coordination of both translational and rotational maneuvers of the UAV such that the target is kept in the center of the optical flow. The trajectory generator provide command to both the x and y axis input forces and coordinated roll and yaw inputs.

For the Yaw controller the design has uses the camera input and the controller design proposed earlier. It would be better to use the camera image processed control signal as input to the yaw controller for good maneuvering capabilities.

CHAPTER 5

IMAGE PROCESSING

5.1. INTRODUCTION

The eye and brain are biological systems that have the capability to adapt to changes, act like sensors and a very reliable processor respectively that are able to enhance, segment, register and recognize features of images. Computer vision attempts to emulate these capabilities. Image enhancement, feature extraction, segmentation and registration plays a big role in analyzing images in field such as medical, geosciences, remote sensing, facial extraction, biometrics and many more. The main challenges in these type of application is to find the most suitable, accurate, faster and robust algorithm. These days most algorithms have been design for customary use only thus there is a need to find a more adaptable one.

5.2. TARGET DETECTION AND IDENTIFICATIONS

Object detection is the process of detecting a target object in an image or a single frame of the video. Object detection will only work if the target image is visible on the given input. If the target object is hidden by any interference it will not be able to detect it. Here it is advisable to use a trainable image detection and identification algorithm implemented with ***Python***. YOLO (you only See Once) Algorithm is the best technique for such kind of applications. Object tracking is trained to track the trajectory of the object despite the occlusions [30].

The following considerations are taken for defining the target parameters:

(i) The approximate size of the object is known;

(ii) The target's gray-level is in general significantly higher than the background.

(iii) When the algorithm selects more than one position, the one closest to the predicted target position is chosen.

(iv) If the detected position is far from the predicted one, the observation is discarded.

5.3. YOLO ALGORITHM

Compared to the approach taken by object detection algorithms before YOLO, which repurpose classifiers to perform detection, YOLO proposes the use of an end-to-end neural network that makes predictions of bounding boxes and class probabilities all at once.

The YOLO algorithm works by dividing the image into ***n*** grids or Residual blocks, each having an equal dimensional region of ***a x a***. Each of these ***n*** grids is responsible for the detection and localization of the object it contains. Correspondingly, these grids predict ***b*** bounding box coordinates relative to their cell coordinates, along with the object label and probability of the object being present in the cell.

This process greatly lowers the computation as both detection and recognition are handled by cells from the image, but it brings forth a lot of duplicate predictions due to multiple cells predicting the same object with different bounding box predictions. It makes use of Non Maximal Suppression to deal with this issue.

Two-stage object detection refers to the use of algorithms that break down the object detection problem statement into the following two-stages:

1. Detecting possible object regions.
2. Classifying the image in those regions into object classes.

Popular two-step algorithms like Fast-RCNN and Faster-RCNN typically use a Region Proposal Network that proposes regions of interest that might contain objects.

YOLO is used here because of its high speed, accuracy and learning capability [31]

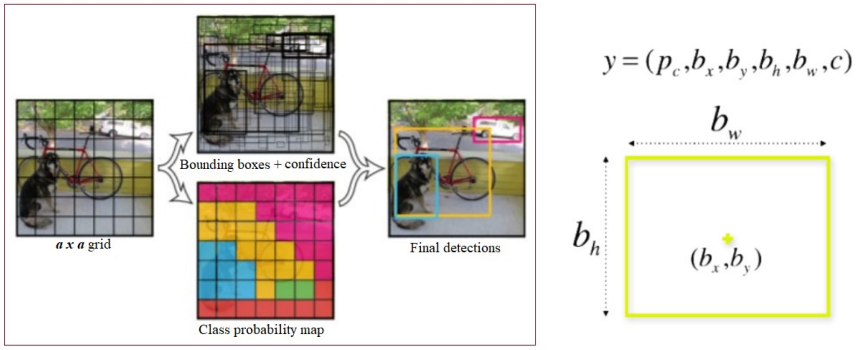


Figure 5.1: YOLO algorithm operational description

Parameter assignment for the final detection outputs of 6 x 1 matrixes [Detection P_c, Anchors center point b_x, b_y, dimension b_h, b_w and Class assignment C].

After having the pixel coordinates b_x, b_y, the trajectory will be to keep these coordinates at the center of the image frame [19], which means the processor will actuate the motors by sending PWM signals. These all data processing needs a very fast and low power processor to mitigate with longer surveillance time and target tracking, so we need to consider carefully choosing the GPU.

The user specifies home base location, coordinates for operational area and information about the target of its dimensions, form and color.

The first two inputs will be utilized by the Trajectory Generation and Navigation systems, while the third input is specifically intended for the Image Processing unit. Upon receiving the information, the Trajectory generator and Navigation unit will promptly compute the following parameters:

I. The number of segments in the operational area
II. The center coordinates of each segment relative to the home base
III. The distance between each center point and the home base

Once the calculations are completed, the controller will be prompted to navigate towards the center point of the first segment and activate the camera for target tracking. If the image processing unit detects the target, it will perform the necessary calculations and advise the trajectory generator on the actual real-time displacement needed to maintain the target at the center and track it. Alternatively, if there is no detection, the trajectory unit will signal the controller to proceed to the next segment and apply the same conditions.

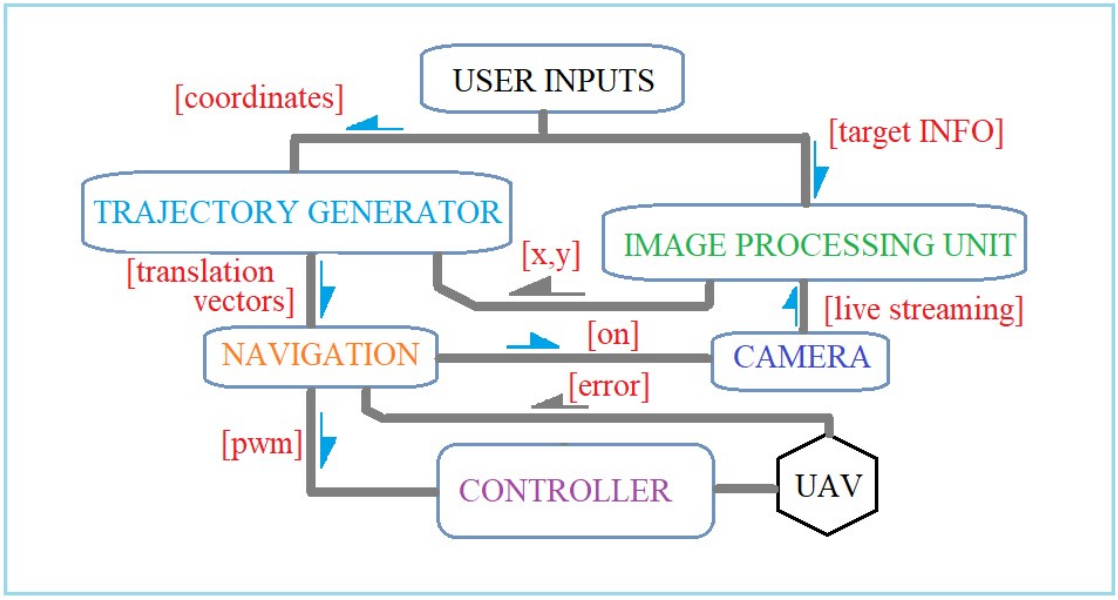


Figure 5.2: Operational flow chart

CHAPTER 6

SIMULATION RESULTS

6.1. PARAMETER ASSIGNMENT

The proposed ASMC are implemented via MATLAB/*SIMULINK*, the model parameters of the quad-rotor and the design parameters of the ASMC are listed in the following Table.

Variable	Value	Variable	Value
β_1	40	β_2	0.005
β_5	2.2	β_6	100
d_z	10	$d_\emptyset$	10
C_1	200	C_2	200
h_1	2500	h_2	100
β_3	2.2	β_4	100
β_7	50	β_8	100
d_θ	10	d_φ	10
C_3	200	C_4	4
h_3	100	h_4	1000

Table 6.1: Controller Parameter Assignment for the ASMC

Variable	Value
Image pixel in lateral axis (x)	320
Image pixel in longitudinal axis (y)	240
Gravitational Acceleration (g)	9.81

Table 6.2: Controller Parameter assignment for Trajectory Generator

The input signal used to simulate all the ***SIMULINK*** tests is a step input of up to 2 units depending on the requirement to check the controller robustness and operational validity.

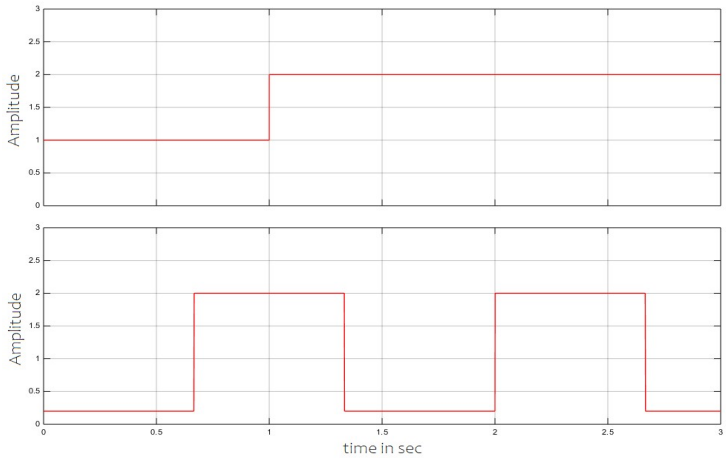


Figure 6.1: Control input used during the ***SIMULINK*** simulations

6.2 ALTITUDE CONTROLLER SIMULATION RESULTS

During the power up and take off maneuver the UAV has the following angular speed and altitude output response.

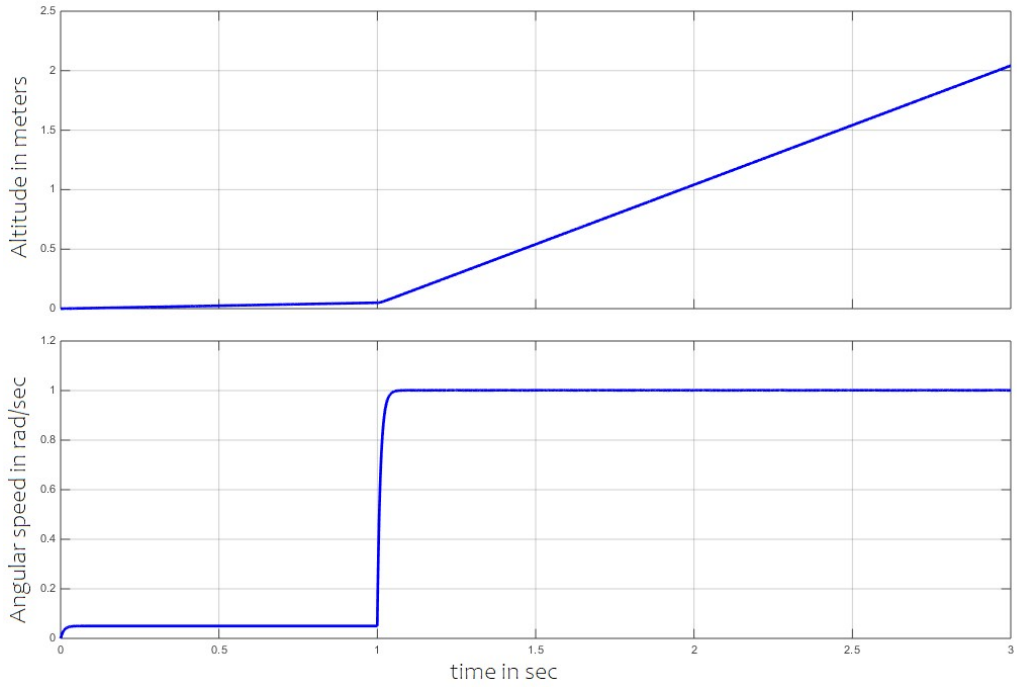


Figure 6.2: Power up and Take off output profiles

The ascent or climb phase of the UAV begins as the speed of all the BLDC motors gradually increases over time. As the motor speed decreases, the rate of climb also diminishes until it reaches the hovering RPM. If the motor speed drops below 0.05 times the maximum attainable speed, the UAV transitions into a descent mode. Therefore, the minimum speed required for hovering is 0.05 times the maximum attainable motor speed. In all scenarios, there will be a saturation climb altitude, which denotes the maximum altitude the UAV can reach during the ascent phase.

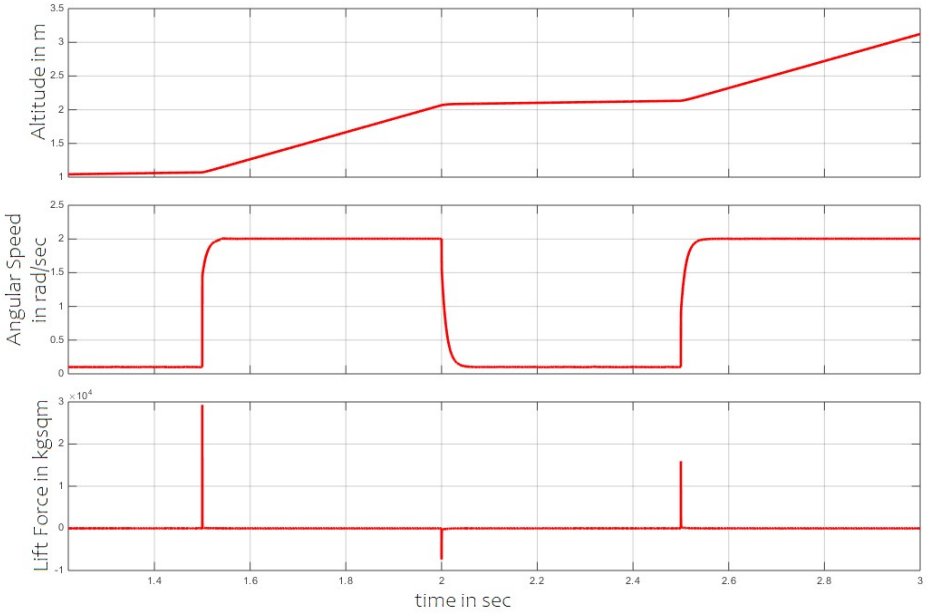


Figure 6.3: Climb and cruise maneuvers of the UAV output response.

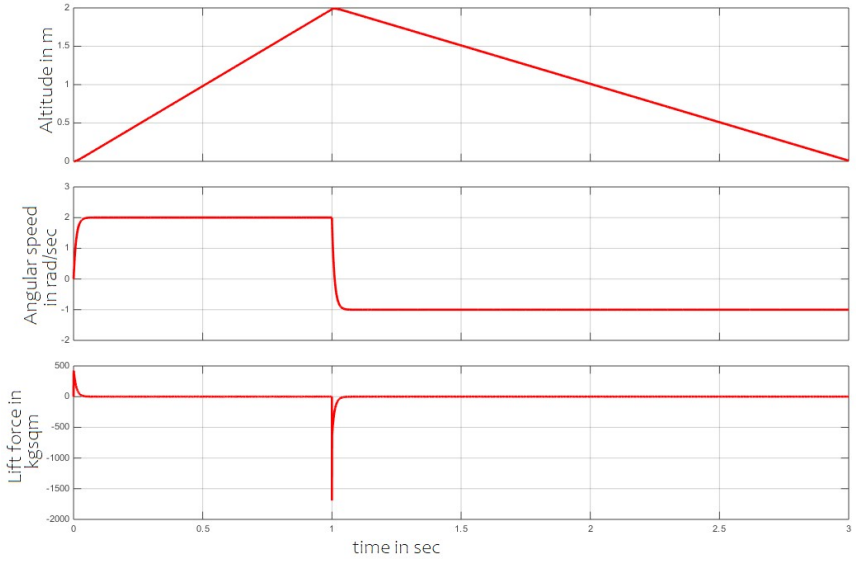


Figure 6.4: Decent and landing maneuvers output response

If the motor speed of the UAV is intentionally or unintentionally reduced below the critical hovering threshold, a significant consequence is the inevitable loss of altitude. As the motors slow down, the lift generated by the rotors diminishes, causing a gradual descent of the UAV. Without sufficient motor power to counteract gravity, the aircraft will continue to descend until it eventually reaches the ground. This highlights the crucial role of maintaining the appropriate motor speed for hovering, as any deviation can lead to an uncontrollable descent and potential damage to the UAV

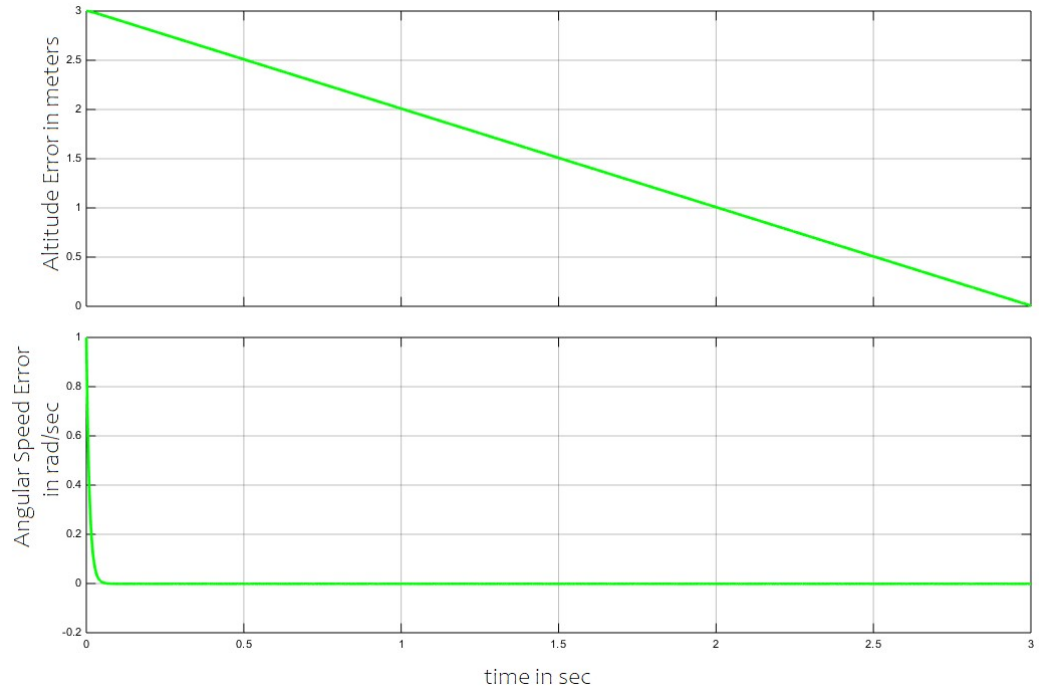


Figure 6.5: Error response of the UAV during Climb and decent

The quad-rotor utilizes the upward thrust generated by its four rotors to maintain a desired altitude. Roll and pitch maneuvers involve tilting the quad-rotor along its lateral (roll) or longitudinal (pitch) axes. By independently changing the speed of the rotors on opposite sides, the quad-rotor generates differential thrust, causing it to tilt in the desired direction. These maneuvers enable the quad-rotor to move horizontally, change direction, or execute agile movements.

When performing Roll or Pitch maneuvers, the quad-rotor needs to consider altitude maintenance to ensure stability and controlled flight. As the quad-rotor tilts, the overall thrust vector also changes, affecting the altitude. The flight controller compensates for this by adjusting

the collective pitch of the rotors while simultaneously controlling the individual rotor speeds to maintain both the desired altitude and the desired roll/pitch angles.

This integration between altitude maintenance and roll/pitch maneuvers allows the quad-rotor to perform complex flight tasks, such as flying at a specific altitude while executing precise lateral or longitudinal movements. The flight controller continuously adjusts the rotor inputs based control algorithms to ensure the quad-rotor's stability, responsiveness, and accurate execution of altitude and maneuver commands.

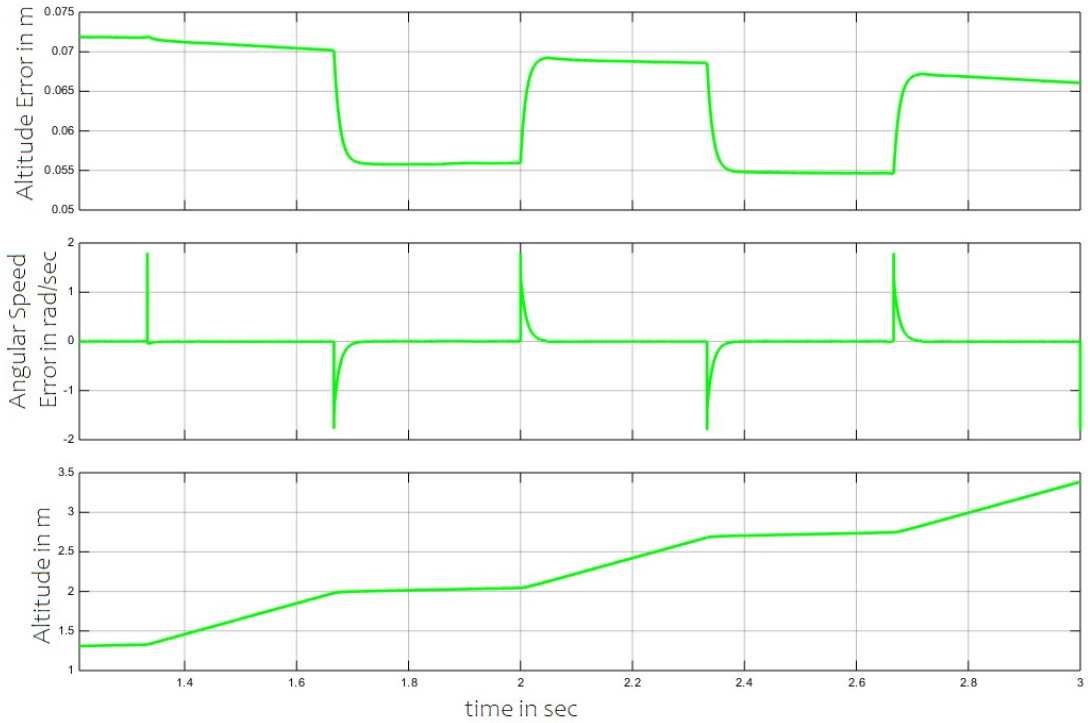


Figure 6.6: Cruise Altitude and Angular speed Error plots

During the simulation, the maximum altitude error is only 7cm. The coupling effect does not exhibit significant cumulative errors. This is achieved through the implementation of a robust controller that effectively decouples the effects and compensates for them, even in the presence of random external disturbances that may perturb the maneuver.

6.3. ATTITUDE CONTROLLER SIMULATION RESULTS

Roll, Pitch and Yaw change rates depends on the speed change rates of respective UAV motors. The control objective is to minimize and eventually eliminate the overshoot of the required attitude subject to unpredictable external disturbance.

When a specific maneuver initiated, the controller has to pick an angle to execute and predict the output linear and angular translation to update and hold the requested attitude with minimum transient overshoot and insignificant steady state error.

***For simulation purpose 30 degree roll angle is used.*

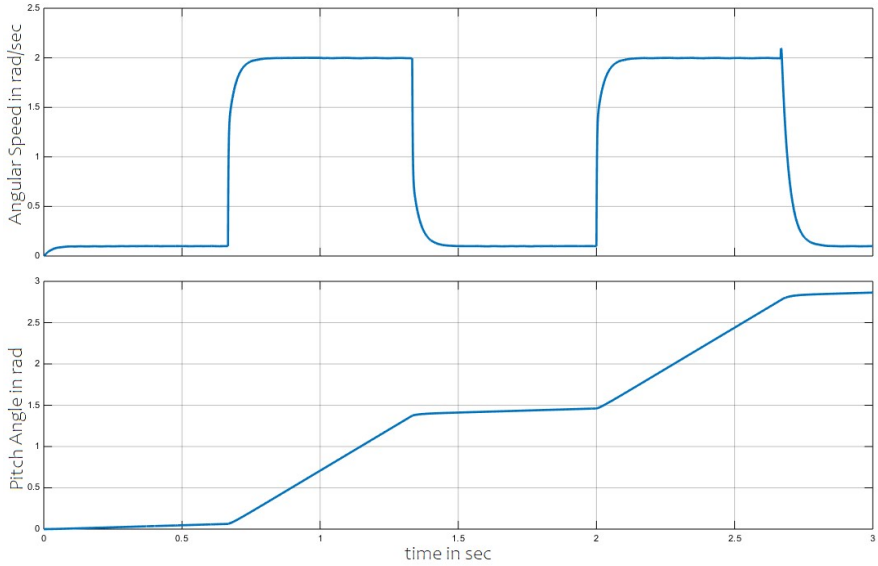


Figure 6.7: Pitch angle attitude plots

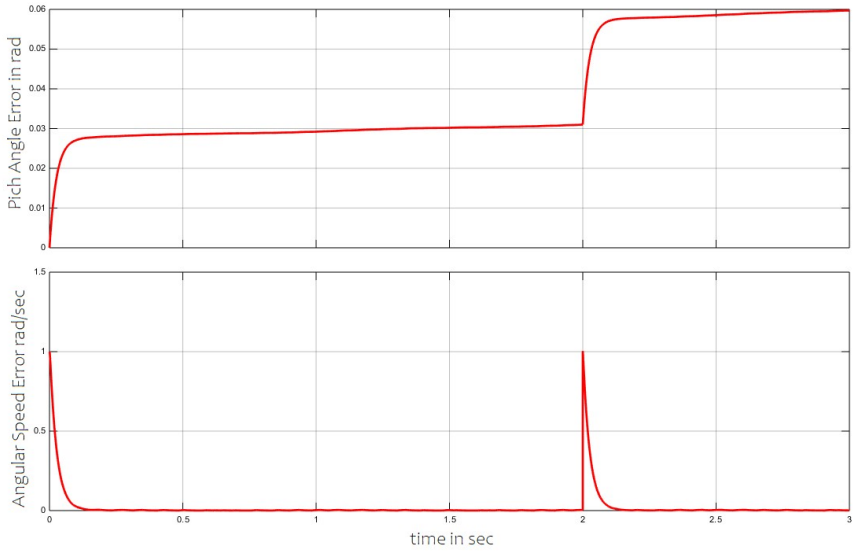


Figure 6.8: Pitch angle and Angular speed error plots

The simulation results showed that the UAV has a maximum pitch angle error of 1.5^o during climb cruise and decent maneuvers which, causes a maximum translational altitude climb or descent error of 7cm.

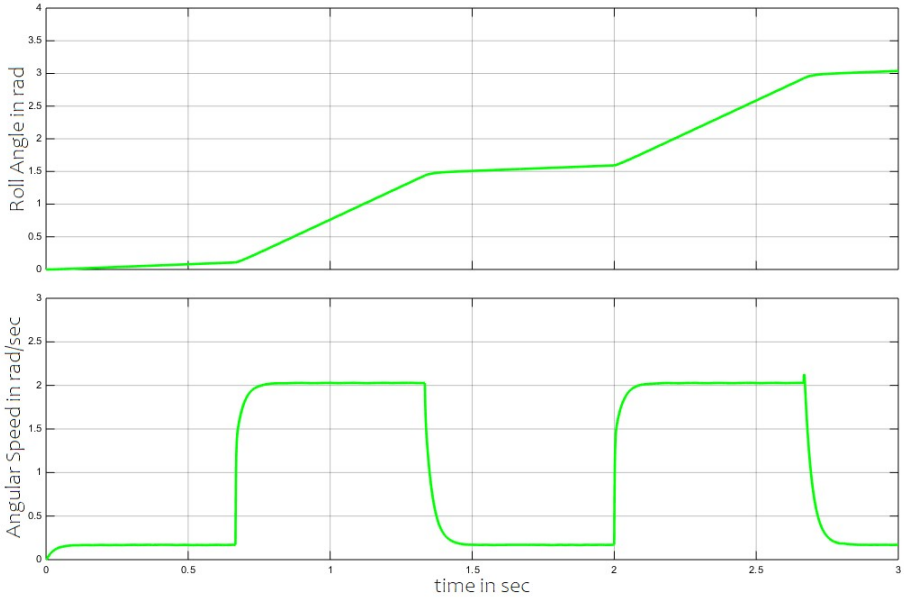


Figure 6.9: Roll Angle attitude plots

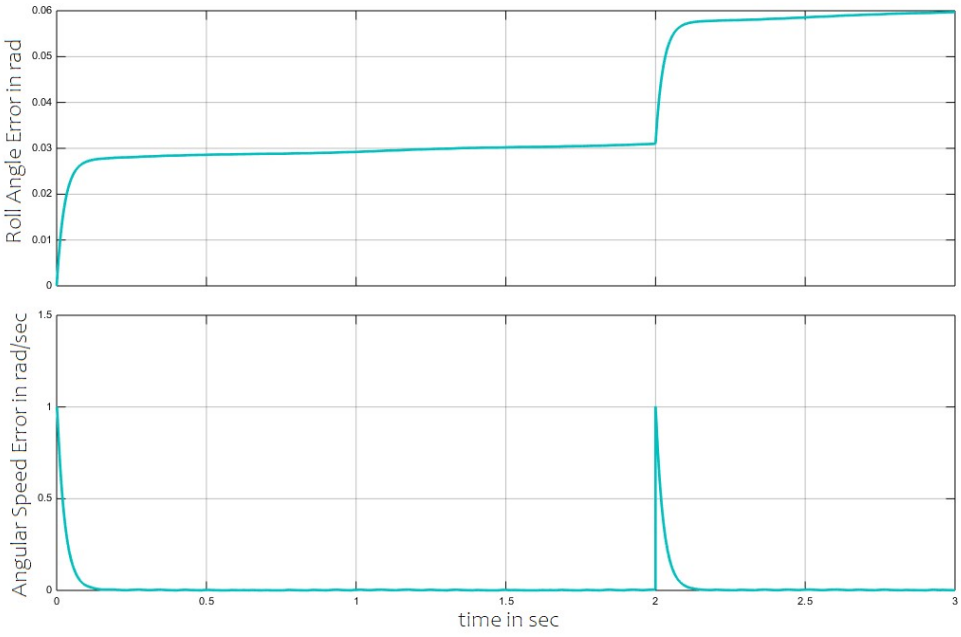


Figure 6.10: Roll angle and Angular speed error plots

From the performed simulation the maximum Roll angle error is 3.42^o the maneuver produces a lateral translation in the roll axis which makes a maximum of 6cm error.

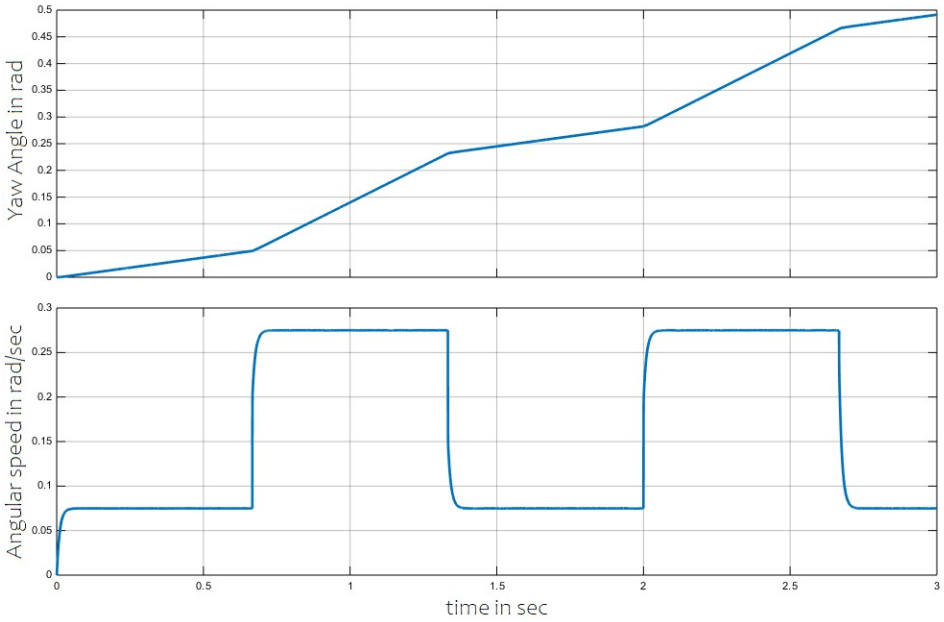


Figure 6.11: Yaw angle of the UAV for centering the target

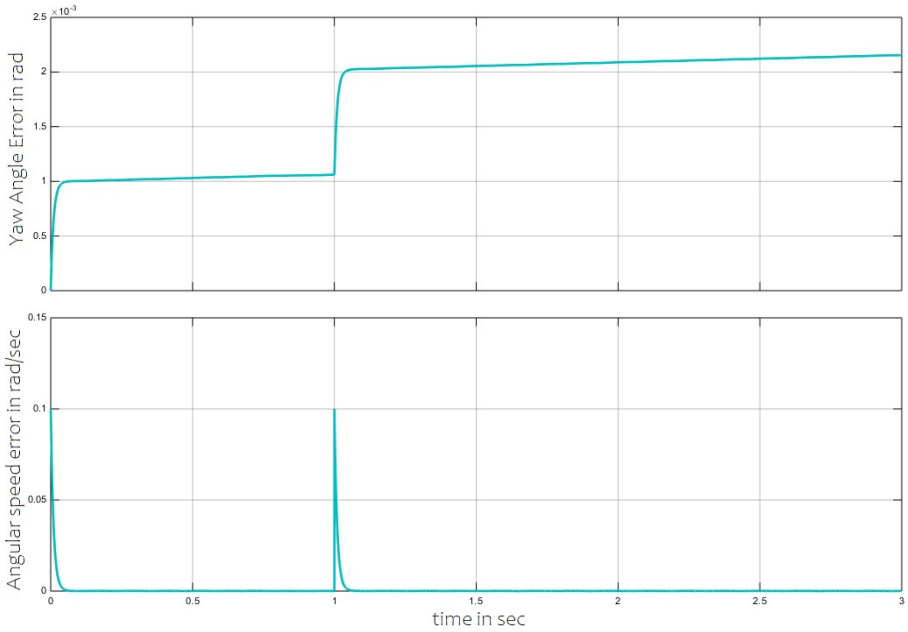


Figure 6.12: Yaw angle and angular speed Error plots

From the simulation the ASMC design of yaw controller has a maximum error angle of 0.00175rad which is equivalent to **0.09975 degrees**. Which makes **φ** equals **44.9 degrees** implies a **0.124% error.** For the case listed in the table of having a target at 250th pixel, the UAV has to travel 250*0.208*0.707 meters to make the target on center. Having error of 0.124% the UAV will be short of **4.56cm,** which is insignificant.

6.4. TRACKING A TOY CAR

The objective is to track a yellow toy car amidst environmental disturbances caused by other multiple toy cars and obstacles that obstruct the target's visibility.

Initially, the quad-rotor takes off from its starting point and travels P_1, P_2, P_3, P_4 *and* P_5. The purpose of this simulation is to monitor the path of the quad-rotor and assess the effectiveness and performance of the controller.

Figure 6.14 can serve as a reference input signal for one of the UAV's motors, ensuring that the maneuver is executed with minimal error. In this case, the system is expected to complete a total maneuver angle of 141 degrees or 2.47 radians, starting from the origin and reaching point P_5, while maintaining a hover height of 5 meters.

The total simulation time elapsed was 7 seconds, which included a 2-second climb operation. The remaining 4.98 seconds, rounded to 5 seconds, were dedicated to the Roll/Pitch maneuvers.

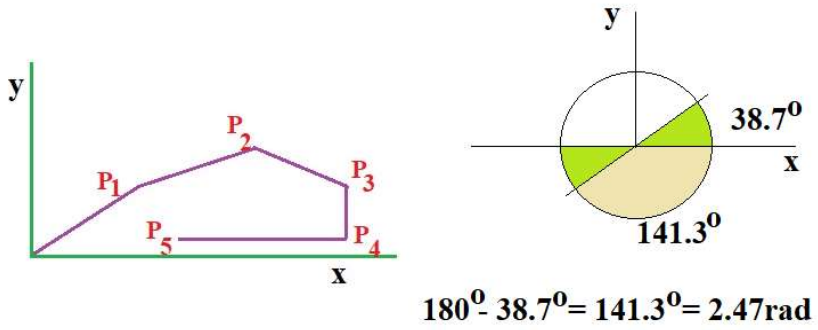


Figure 6.13: Quad-rotor simulation path and total rotational angle covered.

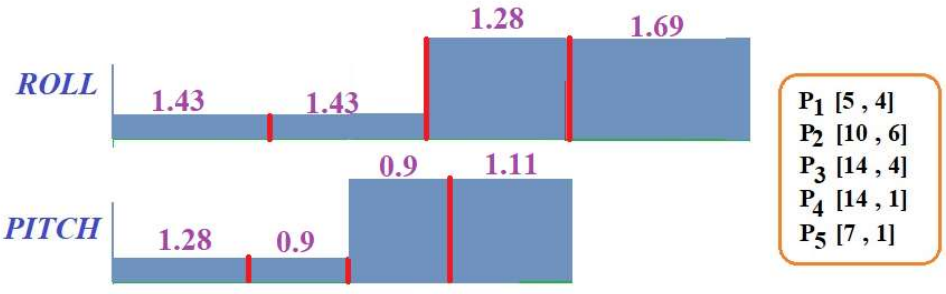


Figure 6.14: Single motor power requirement versus time in seconds

The signal sequencing mentioned above was utilized as an input to the system using MATLAB/SIMULINK. This signal was generated through the combined efforts of a camera and an image processing unit, which tracked the motion of the yellow toy car. The resulting signal was then fed into the controller, and the subsequent outputs were observed.

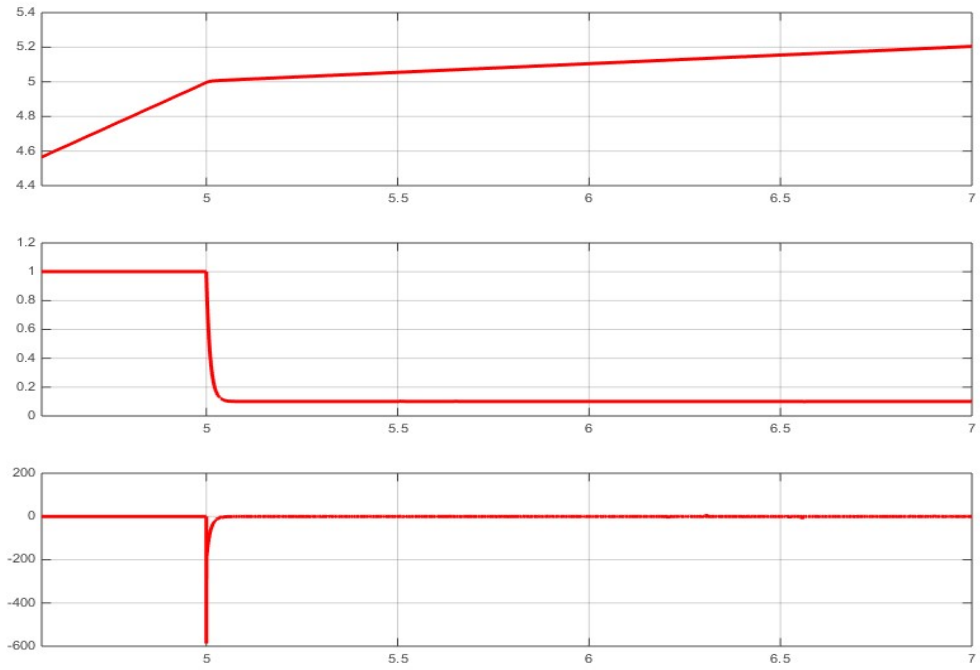


Figure 6.15: Altitude, Motor speed and Lift Force versus time in sec.

Based on the altitude simulation result plot above, at the conclusion of the maneuver, the UAV maintained an altitude of 5.2m instead of the intended 5.0m. This indicates an altitude error of 4cm for every meter traveled, resulting in a cumulative error of 20cm or 4%. It is worth noting that as the altitude increases, this error reduces gradually, reaching below 1% until it eventually saturates. This saturation occurs when the UAV is unable to climb further without increasing the minimum hover speed.

As mentioned previously, the quad-rotor was anticipated to travel 2.47 radians to complete the operations with minimal error. However, based on the observed simulations, the UAV actually reached 2.50 radians at the conclusion of the maneuver. This implies an error of approximately 0.07 radians or 4 degrees, resulting in a total displacement of 145 degrees, which corresponds to an error of 1.2% on both Roll and Pitch maneuvers.

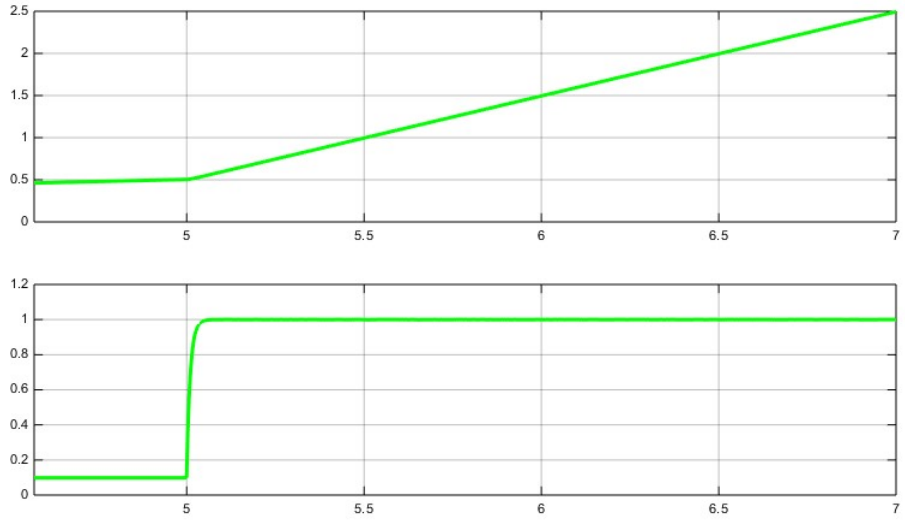


Figure 6.16: Total Roll angle and speed versus time in sec.

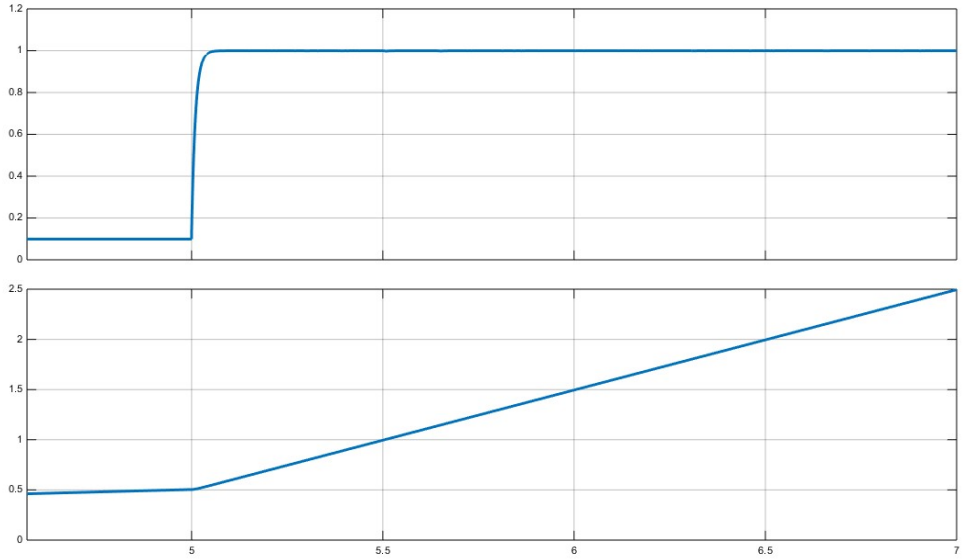


Figure 6.17: Total pitch angle and speed versus time in sec.

CHAPTER 7

DISCUSSION AND CONCLUSION

7.1. DISCUSSION

A robust ASMC scheme is developed for a quad-rotor UAV, which guarantees the asymptotic convergence of the tracking error in altitude and attitude controller. Adaptive laws are designed for estimation of unknown parameters involved in SMC. A proof of stability of the proposed control algorithm is accomplished through Lyapunov analysis. Numerical simulations have been carried out to verify the effectiveness of the proposed controller.

YOLO algorithm image processing unit has successfully identified and tracked the defined target. The simulated tracking has demonstrated that the integrated system will will function as intended.

Real implementation of the controller requires the embedment of video camera real time image processing and based on that a trajectory control input signal sequence will generated and command the UAV. This include the optical flow and pixel wise gray scale comparison accomplished to triangulate the exact location coordinates of the target which will be updated in some interval to avoid error.

7.2. RECOMMENDATION

For future this design can also extended to consider working on different terrain features, wide area reconnaissance operations and magnetic signature guided object scanning and identifications.

Searching rescue operations, wild fire detection and fighting and also urban traffic surveillance can be effectively conducted based on such systems.

Finally the implementation and actual tests can be done with recommended GPU and additional accessories on the appendix section attached in the thesis for future work.

REFERENCES

1. H. Zhou, H. Kong, L. Wei, D. Creighton, and S. Nahavandi, "Efficient road detection and tracking for unmanned aerial vehicle," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 297–309, 2015.
2. C. Li, Y. Zhang, and P. Li, "Full control of a quad-rotor using parameter-scheduled back-stepping method: implementation and experimental tests," *Nonlinear Dynamics*, vol. 89, no. 1, pp. 1–20, 2017.
3. B. Abci, G. Zheng, D. Efimov, and M. E. B. E. Najjar, "Robust altitude and attitude sliding mode controllers for quad-rotors," in *Proceedings of World Congress of the International Federation of Automatic Control*, pp. 2720–2725, Toulouse, France, July 2017.
4. H. G. Krapp and R. Hengstenberg, "Estimation of self-motion by optic flow processing in single visual interneurons," *Nature*, vol. 384, no. 6608, pp. 463–466, 1996.
5. N. Franceschini, F. Ruffier, and J. Serres, "A bio-inspired flying robot sheds light on insect piloting abilities," *Current Biology*, vol. 17, no. 4, pp. 329–335, 2007.
6. Siam, M.; ElSayed, R.; ElHelw, M. "On-board multiple target detection and tracking on camera-equipped aerial vehicles", *Robotics and Biomimetics (ROBIO)*, 2012 IEEE International Conference on, pp, 2399 – 2405.
7. Tice, Brian P. (Spring 1991). "Unmanned Aerial Vehicles – The Force Multiplier of the 1990s". *Airpower Journal*. Archived from the original on 24 July 2009.
8. https://www.dji.com/
9. Zinober, A S. I. and Owens, D. H., *"*Nonlinear and adaptive control*", Springer-Verlag,* Berlin, 2003.
10. Habibi, Aghdam, Hamed. "Guide to convolutional neural networks : a practical application to traffic-sign detection and classification". Heravi, Elnaz Jahani. Cham, Switzerland, 2017.
11. https://paperswithcode.com/dataset/pascal-voc#:~:text=PASCAL%20VOC%20(PASCAL%20Visual%20Object%20Classes%20Challenge)&text=object%20class%20annotations
12. Intermediate dynamics for engineers Newton-Euler and Lagrandian mechanics by Oliver M. OReilly Cambridge University, second edition, 2017.

13. Optic Flow and Beyond by Milena Raffi, Ralph M. Siegel (auth.), Lucia M. Vaina, Scott A. Beardsley, Simon K. Rushton et-Al, *Boston University, U.S.A. vol. 234, 2004.*
14. Tensor Algebra and Tensor Analysis for Engineers by Mikhail Itskov, RWTH Aachen University, Eilfschornsteinstr.18 52062 Aachen, Germany, 2007.
15. A Robust Control for an Aerial Robot Quad-rotor under Wind Gusts by Li Ding and Zhenwei Wang *College of Mechanical Engineering, Jiangsu University of Technology, Changzhou 213001, China*, *Department of Industrial and System Engineering, The Hong Kong Polytechnic University, 999077 Hung Hom, Hong Kong, 2018.*
16. Robust Tracking Control of a Quad-rotor UAV Based on Adaptive Mode Controller, Tianpeng Huang, Deqing Huang , Zhikai Wang, and Awais Shah, *School of Electrical Engineering, Southwest Jiaotong University, Chengdu, China, 2019.*
17. Visual Flight Control of a Quad-rotor Using, Bio-inspired Motion Detector, Lei Zhang, Tianguang Zhang, HaiyanWu, Alexander Borst and Kolja K¨uhnlenz, *Institute of Automatic Control Engineering (LSR),Technische Universit¨at M¨unchen, 80290 M¨unchen, Germany,2012.*
18. M. T. Do, Z. Man, C. Zhang, J. Jin, and H. Wang, "Robust sliding mode learning control for uncertain discrete-time MIMO systems," *IET Control Theory and Applications*, vol. 8, no. 12, pp. 1045-1053, 2014.
19. H. Wang**,** Z. Man, W. Shen, and M. T. Do, "Robust control for Steer-by-Wire systems with partially known dynamics," *IEEE Transactions on Industrial Informatics***,** 2014, accepted for publication, June 2014.
20. R. Wang and J. Liu, "Trajectory tracking control of a 6-dof quad-rotor UAV with input saturation via back-stepping," *Journal of the Franklin Institute*, vol. 355, no. 7, pp. 3288–3309, 2018.
21. S. Mobayen, "A novel global sliding mode control based on exponential reaching law for a class of underactuated systems with external disturbances," *Journal of Computational and Nonlinear Dynamics*, vol. 11, no. 2, p. 9, 2016.
22. F. Muñoz, I. Gonz´alez-Hern´andez, S. Salazar, E. S. Espinoza, and R. Lozano, "Second order sliding mode controllers for altitude control of a quad-rotor UAV: real-time implementation in outdoor environments," *Neurocomputing*, vol. 233, pp. 61–71, 2017.

23. https://www.electricaltechnology.org/2016/05/bldc-brushless-dc-motor-construction-working-principle.html
24. H. R´ıos, J. Gonz´alez-Sierra, and A. Dzul, "Robust tracking output-control for a quad-rotor: a continuous sliding-mode approach," *Journal of the Franklin Institute*, vol. 354, no. 15, pp. 6672–6691, 2017.
25. https://www.techtonics.in/a2212-1000kv-bldc-brushless-dc-motor
26. N. Fethalla, M. Saad, and H. Michalska, "Robust observer based dynamic sliding mode conroller for a quad-rotor UAV," *IEEE Access*, vol. 6, pp. 45846–45859, 2018.
27. B. Abci, G. Zheng, D. Efimov, and M. E. B. E. Najjar, "Robust altitude and attitude sliding mode controllers for quad-rotors," in *Proceedings of World Congress of the International Federation of Automatic Control*, pp. 2720–2725, Toulouse, France, July 2017.
28. http://gsp.humboldt.edu/olm/Courses/GSP_216/online/lesson1/intro-rs.html
29. https://en.wikipedia.org/wiki/Hilbert_curve
30. https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection/#:~:text=YOLO%20algorithm%20employs%20convolutional%20neural,in%20a%20single%20algorithm%20run.
31. https://medium.com/@MrBam44/yolo-object-detection-using-opencv-with-python-b6386c3d6fc1#:~:text=YOLO%20algorithm%20employs%20convolutional%20neural,in%20a%20single%20algorithm%20run.
32. https://developer.nvidia.com/embedded/jetson-nano-developer-kit
33. https://www.techtarget.com/whatis/definition/lithium-polymer-battery-LiPo#:~:text=A%20lithium%2Dpolymer%20battery%20(LiPo,a%20lithium%20ion%20polymer%20battery.
34. https://www.amazon.com/Hobbypower-Plastic-Propeller-Quadcopter-Multi-rotor/dp/B00Q2JFNV4
35. https://robu.in/product/600tvl-170-degree-super-small-color-video-mini-fpv-camera-audio-mini-200-250-300-quadcopter/

APPENDIX I: IMPLEMENTATION

i. NVIDIA JETSON NANO 4GB

NVIDIA Jetson Nano Developer Kit is a small, powerful computer that lets you run multiple neural networks in parallel for applications like image classification, object detection, segmentation, and speech processing. All in an easy to use platform that runs in as little as 5 watts, which makes it the best fit for Quad-rotor.

Designed for use in power-limited environments, the Jetson-Nano squeezes industry-leading compute capabilities, 64-bit operating capability, and integrated advanced multi-function audio, video and image processing pipelines into a 260-pin SODIMM.

The Maxwell GPU architecture implemented several architectural enhancements designed to extract maximum performance per watt consumed. Core components of the Jetson Nano series module include:

- NVIDIA Maxwell GPU
- ARM® quad-core Cortex®-A57 CPU Complex
    - 4GB LPDDR4 memory
    - 16GB eMMC 5.1 storage
    - Gigabit Ethernet (10/100/1000 Mbps)
    - PMIC, regulators, power and voltage monitors
    - 260-pin keyed connector (exposes both high-speed and low-speed industry standard I/O)
    - On-chip temperature sensors

CPU frequency and voltage are actively managed by Tegra Power and Thermal Management Software and influenced by workload. Frequency may be throttled at higher temperatures (above a specified threshold) resulting in a behavior that reduces the CPU operating frequency. Observed chip-to-chip variance is due to NVIDIA ability to maximize performance [32].

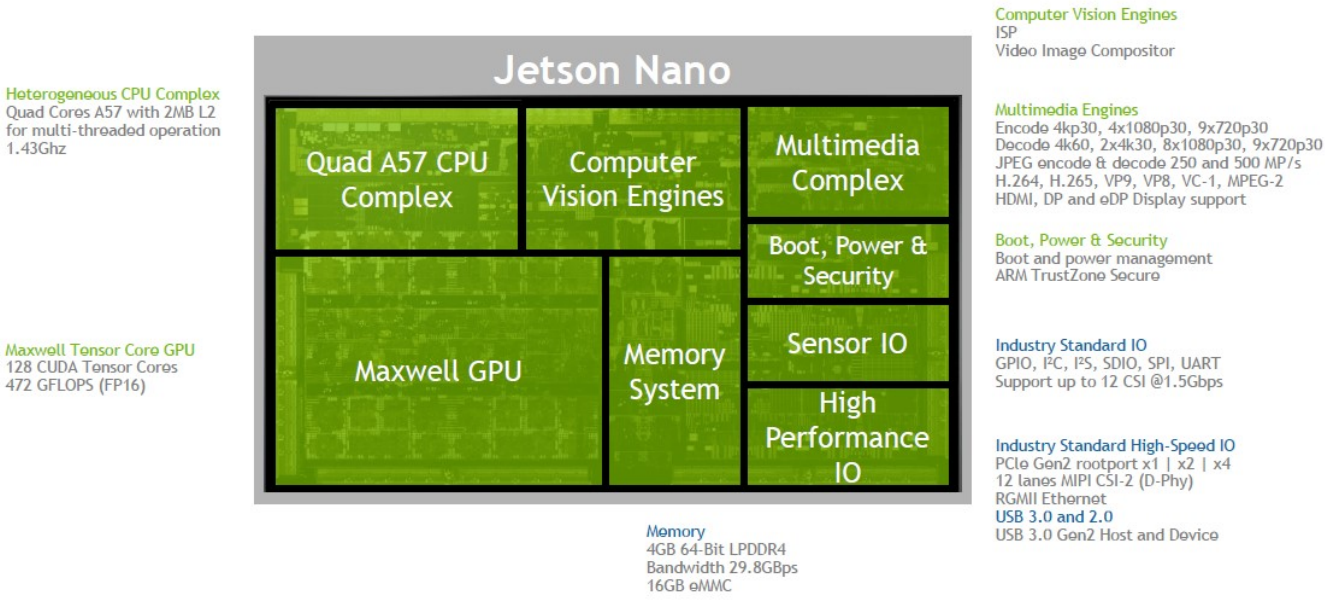


Figure A1: Jetson-Nano system Architecture

The Nvidia Jetson Nano has a 128-core Maxwell GPU at 921 MHz. Compared side by side, the Jetson Nano has a much more capable GPU than the close compotator Raspberry Pi 4. This makes the Jetson Nano more suitable for AI and ML applications, which could be a specific advantage, depending on the intended use.

The Jetson family of modules all use the same NVIDIA CUDA-X™ software, and support cloud-native technologies like containerization and orchestration to build, deploy, and manage AI at the edge. CUDA is programming language for general purpose computing on graphics processor units (GPUs).

ii. POWER ELECTRONICS

In order to control the amount of current being fed into the motor, it is important to use a motor controller which prevents the motor from breaking or burning, and it would also protect a short circuit. The Mystery 12A Brushless Speed Controller allows up to 12 amps of current to flow. It has some safety functions like reducing power when the temperature goes above 120°C and reducing power or shutting off as a low current protection.

Figure A2: 11.1V, 12 A BLDC motor drive circuit

iii. BATTERY

Looking for a battery to use for a quad-rotor, there should be some considerations in order establishing a balance between the weight of the battery and its capacity. An initial consideration must be the weight of the UAV. Generally speaking, a good guideline is to have the total weight at around half of what the motors' thrust value is equal to. This can prevent overheat and fail of the motors, and they will be able to lift the drone off the ground and hover at approximately 50% throttle. Beyond the weight and the capacity of the battery must also be considered. LiPo 3S 11.1V 10000mAh 25C a consumer needs to know how they are categorized so that the proper battery for their needs is purchased. The S-number is the number of battery cells in series, so a 3S is three cells; a 4S is four cells, and so on. Each cell is equivalent to 1.7V, so the voltage of a 3S battery is 11.1V. The more battery cells there are in series, the higher the voltage, which makes it more efficient while lessening current flow. Adding more cells in parallel increases the capacity of the battery as it adds more amps. The mAh is the milliamps per hour discharge rate, which is essentially the battery capacity. The Free-Fly LiPo (lithium polymer) discharges at a rate of 9 amps per hour. All of the batteries were fairly similar in a weight to capacity ratio, so a 10000mAh battery was chosen as it was projected to give a flight time of about 13 minutes. Initial projections when considering smaller capacity batteries indicated that the flight time would only reach about 8 minutes, which falls short of the project goals [33].

Capacity:		11.1v,			10000mAh
Size:	130	x	20	x	74mm
Weight:					394g
Discharge		Rate:			25C

Figure A3: 11.1V, 10000mAh power supply battery

iv. PROPELLERS AND ACCESSORIES

The set of propellers is designed for KY601S foldable drone, perfectly fitting drone having high aerodynamic efficiency, safety, stability and flexibility. The plastic material is light 10 -18g and of high tension, which won't be easily damaged. Easier to install and remove and portable [34].

Figure A4: KY601S Foldable Drone propeller with accessories

v. BLDC MOTORS

Figure A5: A2212/6T 1000K_v BLDC motors[25]

vi. CAMERA

The use of high quality cameras also affects the system overall because it has additional requirements like higher transmission rate, a better processing memory, and more memory space. For this specific case the camera used is already discussed above on photo-grammetry section [35].

Figure A6: 600TVL 170 degree mini FPV AV camera

vii. MISSION CONTROL AND GRAPHIC USER INTERFACE

All the coordinate inputs are user defined. Engine start, shutdown, manual reversion and mission abort are accomplished through the user interface. It is also possible to monitor the operation on a live video streaming.

Figure A7: User interface and Mission control console

As indicated on earlier discussion, the navigation system automatically calculates the center of the segments and generates the trajectory.

viii. DATA COMMUNICATION AND REMOTE CONTROL

A central computer with a receiving and transmitting antenna will interrogate the UAV control system for live video streaming and control feedbacks, to allow the mission control center all the information regarding UAV, target positions and status of the entire system.

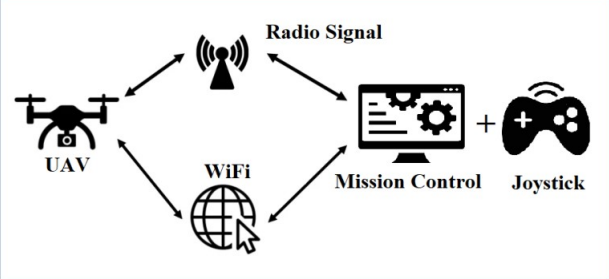


Figure A8 (a): Data Communication protocol used between UAV and Mission Control

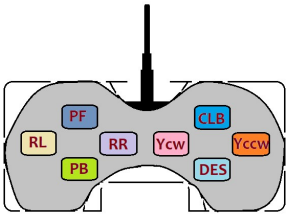


Figure A8 (b): Joystick for Manual mode control

Joystick button addressing for manual operations: By default ***0000*** means hovering (**ALT**) at last assigned altitude.

CLM(ALT)	**0001**	**PF**	**0101**
DES(ALT)	**0010**	**PB**	**0110**
RR	**0011**	**Ycw**	**0111**
RL	**0100**	**Yccw**	**1000**

Table A1: Joystick button addressing for digital communication

ix. VTX (Video Transmitter)

The Video Transmitter (VTX) is a crucial component of the FPV system, wirelessly transmitting camera images or videos. When selecting a 5.8GHz VTX, consider compact size, features, channel support and accurate transmitting frequency, in addition to long range and high output power, to ensure optimal performance and minimal interference for fellow operators.
Recommendation:

Figure A9: HDZero 1W Freestyle VTX

The HDZero Freestyle VTX offers the best range and penetration for the HDZero systems, thanks 1W transmit power. Ideal for aggressive freestyle and long range flights, it features massive heat-sinks both sides for physical protection and cooling. It supports 25mW/200mW, can be upgraded to 1W special firmware. It weights 28g and supports 2S-6S batteries.
The main factors that contribute to FPV signal quality and range are the quality and placement of the antennas and the accuracy in transmitting at the intended frequency.

x. The UAV Frame

Frame is the structural backbone of a UAV. It is the skeleton on which all other components are mounted, and provides protection to all the electronics inside the drone.
A typical FPV drone frame is made of carbon fiber plates and metal hardware consists of:

- 4x Arms
- Top and bottom plates
- Camera mount
- Standoff
- Bolts

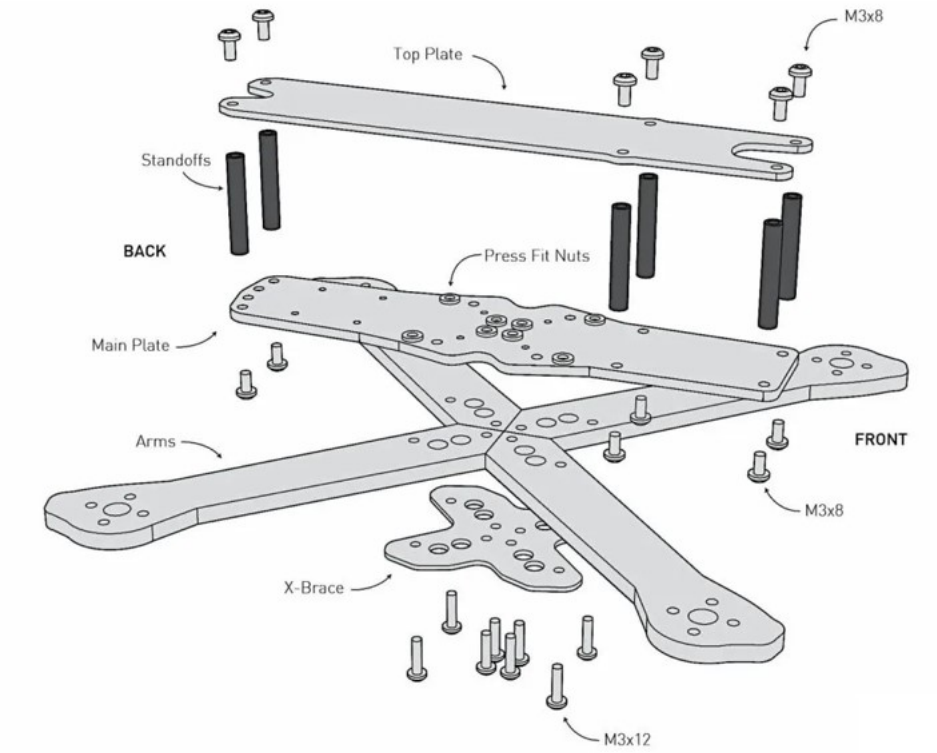


Figure A10: A typical FPV Quad-rotor drone frame

Carbon fiber is the most popular material for FPV drone frames because:

- Low cost: makes it accessible to a broad range of pilots, from beginners to professionals.
- Light weight: a lighter racing drone means faster speed, better agility, longer flight time, and less destructive inertia in a crash.
- Durable: it can withstand the high stresses and impacts that drones encounter during flights and
- Rigidity: CF has high stiffness to weight ratio. Frame rigidity is important for tuning and flight performance.
- Highly customizable: it allows manufacturers to produce frames in various shapes, sizes, and thicknesses to meet the specific needs of different pilots and flying styles.

The Disadvantages of carbon fiber are:

- Carbon fiber is electrically conductive: If you have live wires touching the frame it could cause circuit and burn out components.
- It can block or attenuate radio frequencies; antennas should be mounted outside of the frame for signal strength.

There could be a substantial weight and cost difference when using different metal alloy for bolts/nuts/hardware. Steel is the cheapest and strongest option but also the heaviest. Aluminum is lightest, but also the softest and tends to bend in a crash, and strip easily when too much torque is Titanium is light weight and strong but expensive.

Construction

A drone frame has two main parts: the body and the arms. The body houses and protects your electronic components including flight controller, battery, VTX etc. Typically it consists of a bottom plate, top plate and some standoffs in between to hold them together.
The arms are for installing the motors on. The shape and thickness play a huge role in the durability frame, because the arms tend to be the first place to break in a crash.
Quad-rotor frame size (wheelbase) is the diagonal motor to motor distance measured in millimeters. Dictates the sizes of propeller you can run on it, therefore the frame size often are referred to the maximum propeller size it can support, instead of its wheelbase.
As the motors are mounted on the very end of the arms, the further away they are from the centre of the larger the moment of inertia would be. It introduces a tendency to resist angular acceleration and deceleration and makes your quad-rotor feel more sluggish and less responsive. Therefore you want to use biggest propellers a frame can support to maximize its performance. Here is a table showing roughly what maximum prop size different quad-rotor frame sizes can support:

Frame Size	Prop Size
280mm+	7″
220-250mm	6″
180-220mm	5″
150-180mm	4″
120-150mm	3″
90-120mm	2″

Table A2: Frame to prop size comparison

***If KY601S Foldable Drone propeller having 4 inch length is selected as discussed earlier, the yellow highlighted row will be the perfect frame dimension.*
A typical frame is made up of various parts: 4 arms, 2-3 top/bottom plates, hold together by standoffs bolts.
The arm layout refers to how the arms of the FPV drone frame extend from the body. It might look like the motor layouts are similar (they are all rectangular after all), but it does have an effect on control flight performance. There are some other reasons too when choosing one particular motor layout than another.

H frame

On an “H” frame, the front and back arms are perpendicular to the middle plate, forming an “H” shape. Leads to a long and roomy body section for installing your electronics comfortably, makes it very easy HD camera and battery can both sit on the top plate, the weight of the quad is more spread out. For reason, it might feel less agile than X frames due to higher moment of inertia on the pitch axis.

In recently years the hobby has mostly moved to the X frame designs.

True-X Frames

On a “True-X” frame all the arms intersect at the centre forming an “X” shape, with equal distance between the motors to the center of the frame. True-X frames tend to have a more balanced performance as motors are contributing equally.

Some argue it has little to no difference to an H frame in terms of flight characteristics, since the mass distribution is the same, while others claim there is a difference to how the thrusts are applied on the due to the way the arms are connected to the body (think about leverage). Also the ways vibrations transferred to the FC through the arms are different.

Stretch X Frames

Similar to the “True X” frame, but with the front and back arms stretched farther apart than the side idea is to reduce air disturbance that the rear propellers get from the front propellers, and it might also handling at high speeds and in tight turns.

Dead-cat

This layout features front and back arms that are angled inward and the rear arms that are longer and outward. This creates a unique shape that resembles a dead cat, hence the name.

The dead-cat layout is often used in cinematic rigs as the propellers are out of the camera’s view and ruin the beautiful shot.

Box Frames

A box frame is pretty much an enclosed X Frame. This basically creates a tougher frame that is less has broken arms. However the extra material creates more drag and weight. Not the best choice for performance but definitely helps improve strength of the arms.

Plus Frame

Instead of flying forward like an “X”, a Plus Frame flies forward like a “+” sign. There may be some benefits with regards to motor turbulence though, because the side motors on a plus frame will always be spinning clean air. One of the downsides is the front motor and propeller can easily get in the view of the camera. It is not a common frame type apart from the fact that it looks unique and eye-catching.

Durability

Arm thickness: In general, a thicker arm is less likely to break because more material provides a wider distribute stress over in a crash.

Stress points: avoid arm design that involves hard angles as it's much easier to break in a crash, curves round cutouts are much better for strength. The carbon fiber plates are stronger when cut along the weaves and weaker when it's 45-degrees to the weaves.

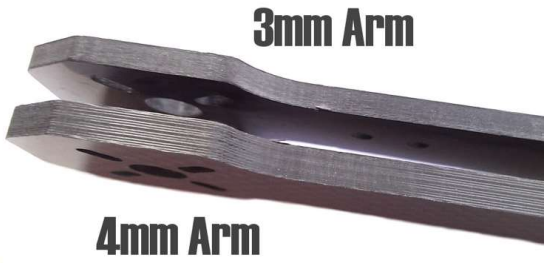


Figure A11: carbon fiber plate thickness comparison

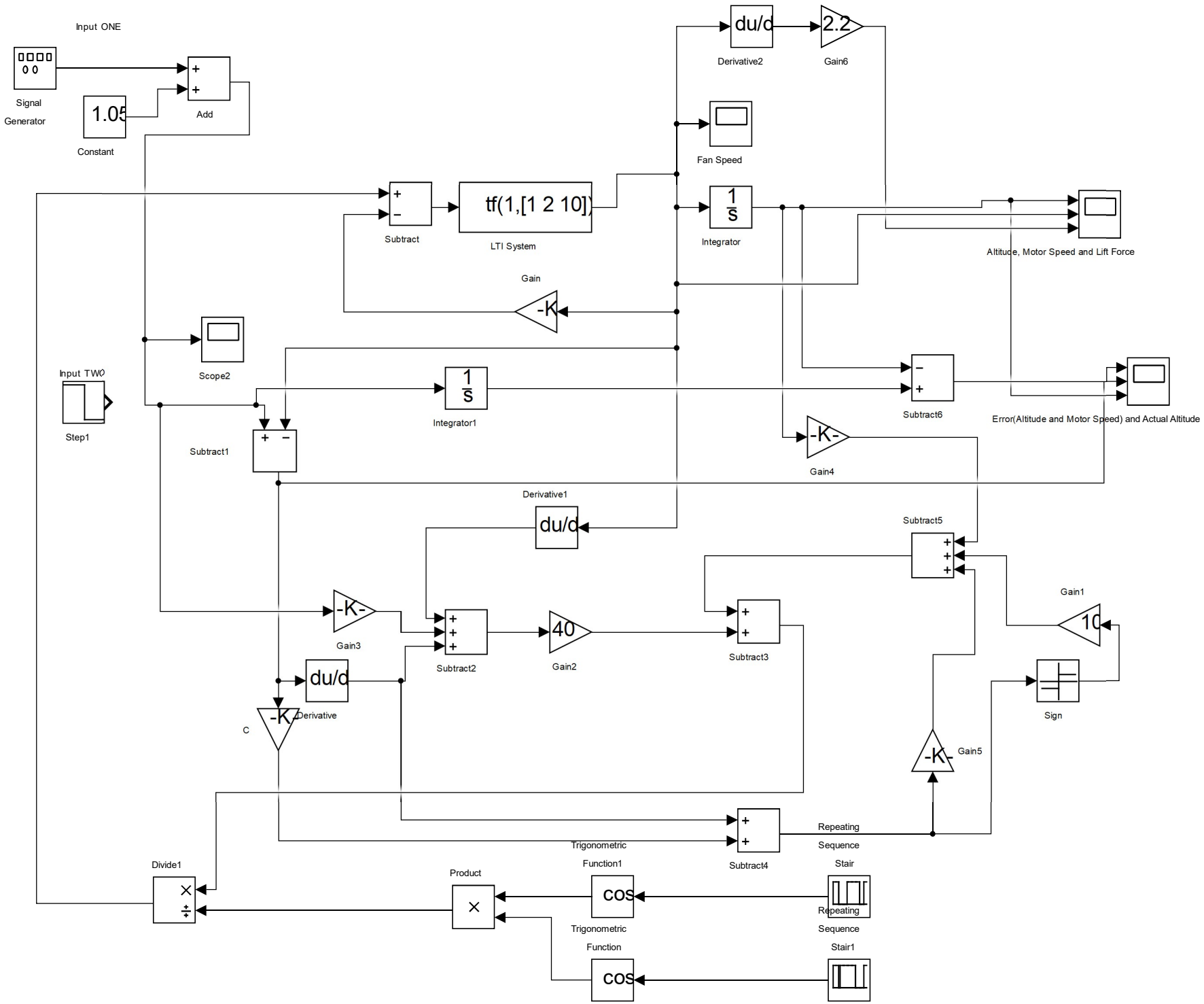
Thicker carbon fiber means better strength, rigidity and sturdiness, but it also gets heavier. Durability matters most on the arms, as they take most of the strain in a crash. This makes it common top, centre and bottom plates cut from thinner 3mm or even 2mm. Any thinner than this will break too easily and probably too flexible for decent flight characteristics.

Frames are designed for specific application and flying styles, usually stated in the product description. Flying styles are: freestyle, racing, long ranges and cinematic

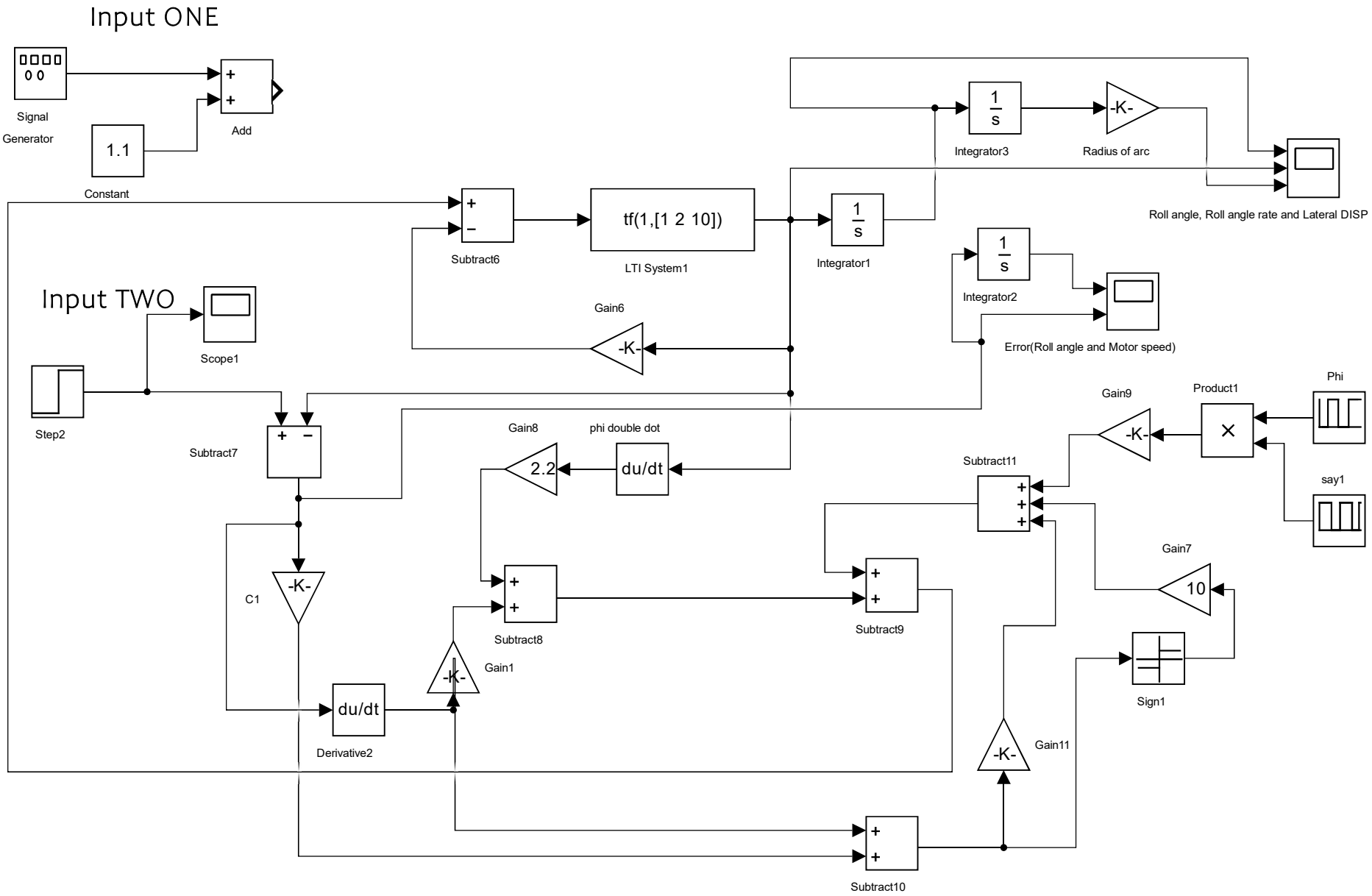
With that being said, there's no hard rule what frame you must use for what purpose. It can be totally a freestyle frame, or freestyle with a racing frame.

For starting out, it is recommended using a freestyle frame as they are versatile, generally easier thanks to the roomy body, and they are relatively durable against crashes.

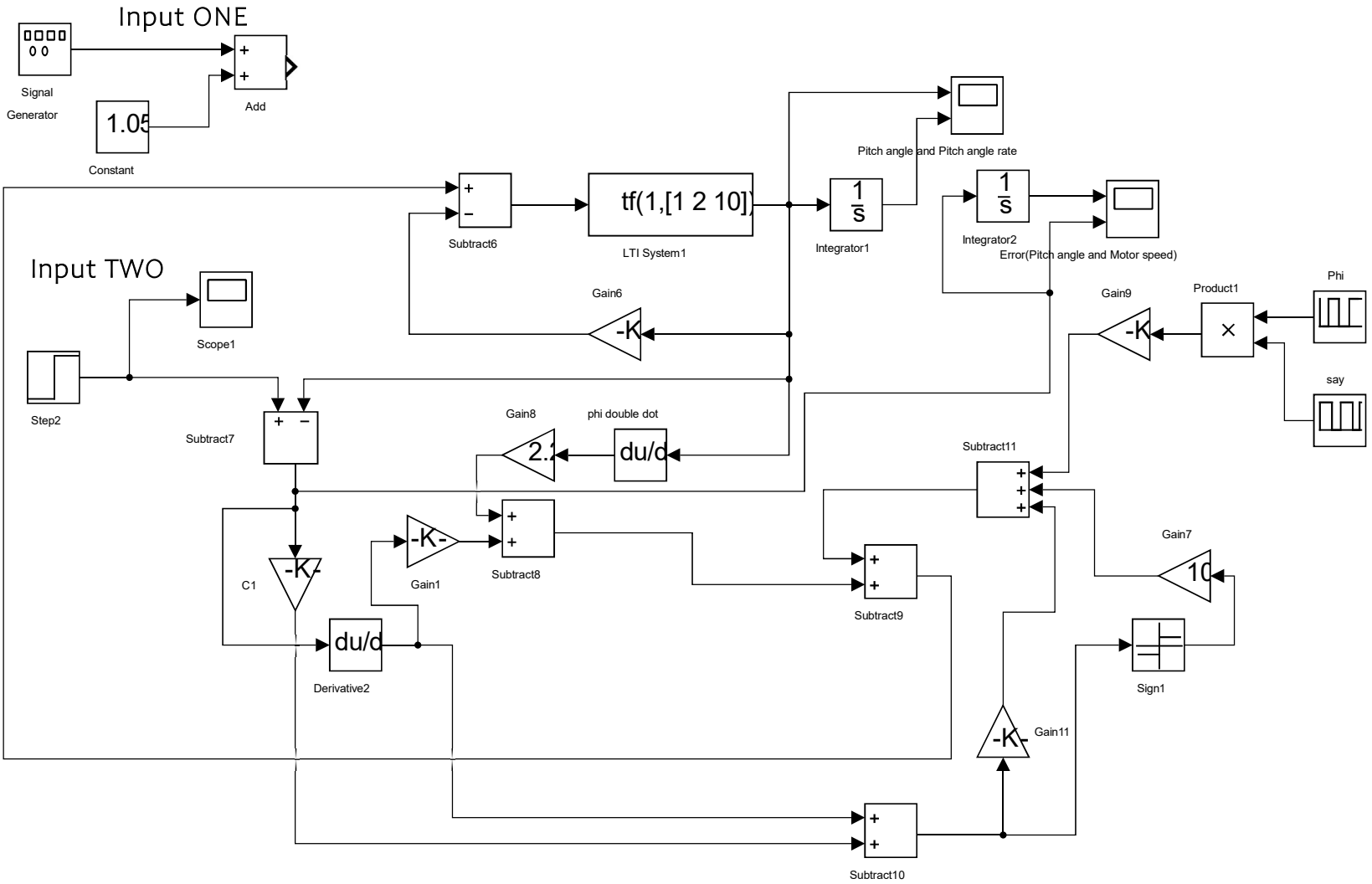
APPENDIX II ALTITUDE CONTROLLER



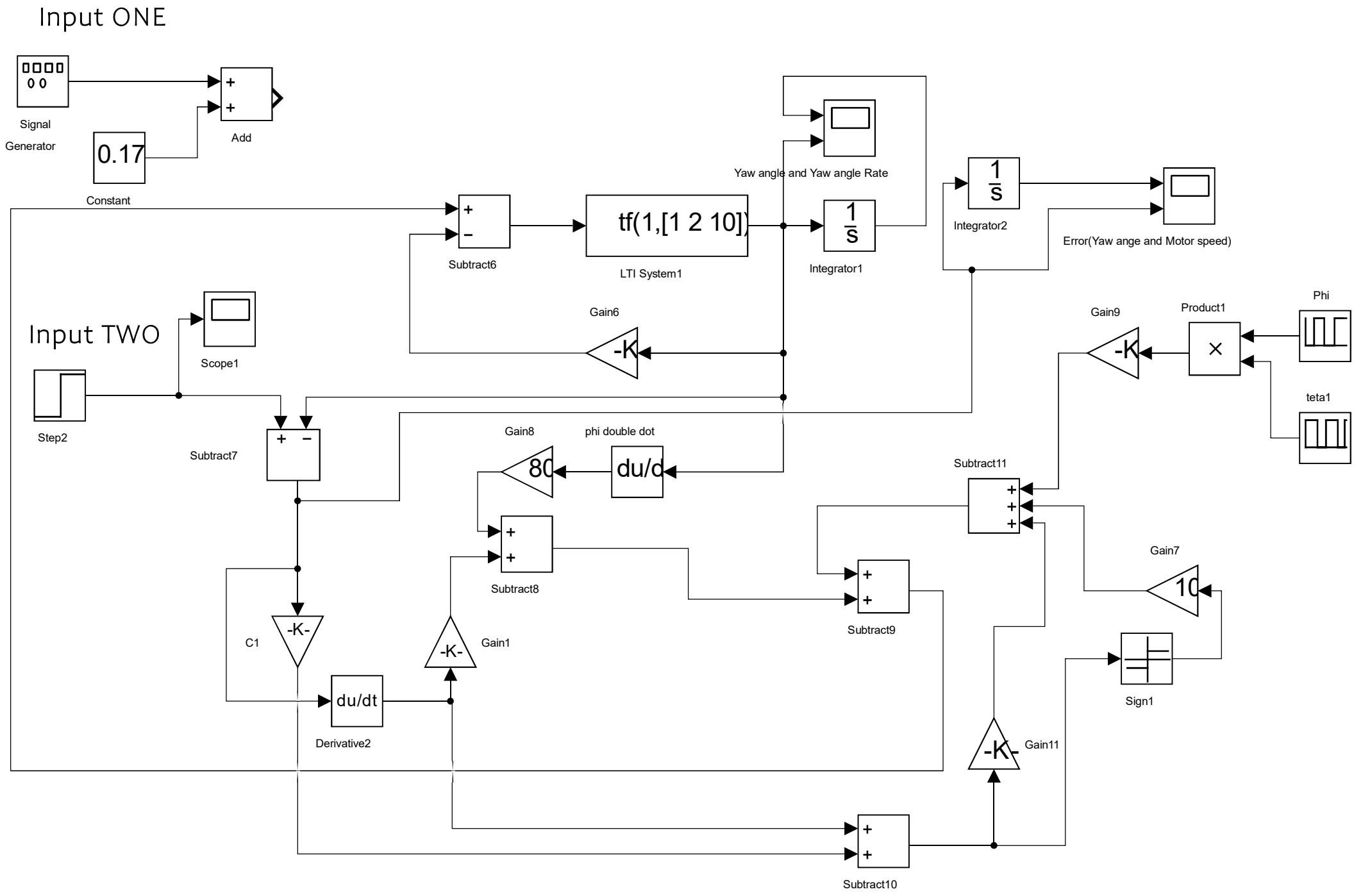
APPENDIX III **ROLL CONTROLLER**



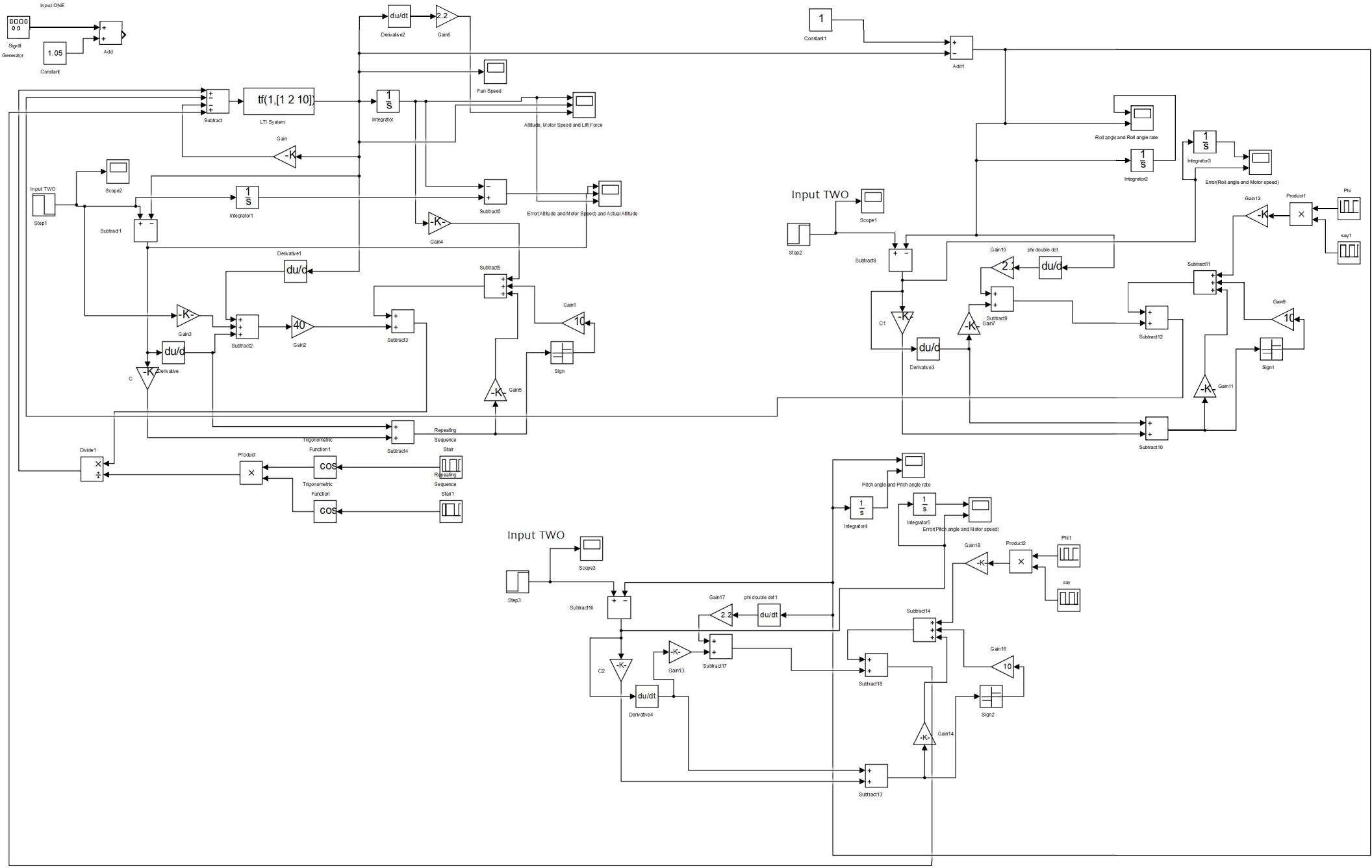
APPENDIX IV **PITCH CONTROLLER**



APPENDIX V YAW CONTROLLER



APPENDIX VI **OVERALL FLIGHT CONTROLLER**



APPENDIX VII: TARGET DETECTION, IDENTIFICATION AND TRACKING *python* CODES

In computer vision, target detection and identification typically involves using algorithms to process images or video feeds to identify and locate objects or individuals. Here the deep learning-based approaches YOLO (You Only Look Once) or Faster R-CNN (Region-based Convolutional Neural Networks) algorithm is used for this task.

Once objects or individuals of interest are detected, they can be further analyzed and identified based on user defined various features, such as shape, color, size, and texture.

Target Detections

```
import cv2
import numpy as np

# Load YOLO weights and configuration files
net = cv2.dnn.readNet("yolov3.weights", "yolov3.cfg")

# Load object classes
classes = []
with open("coco.names", "r") as f:
    classes = [line.strip() for line in f.readlines()]

# Set threshold values for object detection
conf_threshold = 0.5
nms_threshold = 0.4

# Load image
img = cv2.imread("image.jpg")
height, width, _ = img.shape

# Create a blob from the input image
blob = cv2.dnn.blobFromImage(img, 1/255, (416, 416), (0, 0, 0), True, crop=False)

# Set input blob for the neural network
net.setInput(blob)

# Get output layer names
output_layers = net.getUnconnectedOutLayersNames()

# Run forward pass through the network
outputs = net.forward(output_layers)
```

```
# Initialize lists to store detected objects, their confidences, and class IDs
boxes = []
confidences = []
class_ids = []

# Process each output layer
for output in outputs:
    for detection in output:
        scores = detection[5:]
        class_id = np.argmax(scores)
        confidence = scores[class_id]
        if confidence > conf_threshold:
            center_x = int(detection[0] * width)
            center_y = int(detection[1] * height)
            w = int(detection[2] * width)
            h = int(detection[3] * height)
            x = int(center_x - w / 2)
            y = int(center_y - h / 2)
            boxes.append([x, y, w, h])
            confidences.append(float(confidence))
            class_ids.append(class_id)

# Apply non-maximum suppression to eliminate overlapping boxes
indices = cv2.dnn.NMSBoxes(boxes, confidences, conf_threshold, nms_threshold)

# Draw the final boxes and labels on the image
for i in indices:
    i = i[0]
    box = boxes[i]
    x, y, w, h = box
    label = classes[class_ids[i]]
    cv2.rectangle(img, (x, y), (x+w, y+h), (0, 255, 0), 2)
    cv2.putText(img, label, (x, y-10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)

# Display the output image
cv2.imshow("Object Detection", img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Video target tracking

Once the object is detected in the first frame, its location is tracked in the subsequent frames using a tracking algorithm. During the tracking process, the object may be occluded by other objects in the video, causing the tracking algorithm to lose its track. To handle occlusion, various methods can be used, such as re-initializing the tracking algorithm, estimating the object's position and velocity, or using multiple object trackers. Once the object is tracked in the video, its identity can be established using various features such as color, shape, and texture. This helps to differentiate the object from other objects in the video and provide useful information for further analysis. Re-detection: In some cases, the object may disappear from the video frame for an extended period. In such cases, it may be necessary to re-detect the object using YOLO to resume tracking.

```
import cv2
import numpy as np

# Load YOLO weights and configuration files
net = cv2.dnn.readNet("yolov3.weights", "yolov3.cfg")

# Load object classes
classes = []
with open("coco.names", "r") as f:
    classes = [line.strip() for line in f.readlines()]

# Set threshold values for object detection
conf_threshold = 0.5
nms_threshold = 0.4

# Load or live stream video
cap = cv2.VideoCapture("video.mp4")

# Initialize tracker
tracker = cv2.TrackerCSRT_create()

# Process first frame to get initial target location
ret, frame = cap.read()
blob = cv2.dnn.blobFromImage(frame, 1/255, (416, 416), (0, 0, 0), True, crop=False)
net.setInput(blob)
output_layers = net.getUnconnectedOutLayersNames()
outputs = net.forward(output_layers)
boxes = []
confidences = []
class_ids = []
for output in outputs:
```

```
    for detection in output:
        scores = detection[5:]
        class_id = np.argmax(scores)
        confidence = scores[class_id]
        if confidence > conf_threshold and classes[class_id] == "person":
            center_x = int(detection[0] * frame.shape[1])
            center_y = int(detection[1] * frame.shape[0])
            w = int(detection[2] * frame.shape[1])
            h = int(detection[3] * frame.shape[0])
            x = int(center_x - w / 2)
            y = int(center_y - h / 2)
            boxes.append((x, y, w, h))
            confidences.append(float(confidence))
            class_ids.append(class_id)
indices = cv2.dnn.NMSBoxes(boxes, confidences, conf_threshold, nms_threshold)
if len(indices) > 0:
    i = indices[0][0]
    box = boxes[i]
    tracker.init(frame, tuple(box))

# Process subsequent frames and track target
while True:
    ret, frame = cap.read()
    if not ret:
        break
    success, box = tracker.update(frame)
    if success:
        # Draw tracked object box
        (x, y, w, h) = [int(v) for v in box]
        cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
    else:
        # If tracking failed, try to detect target again
        blob = cv2.dnn.blobFromImage(frame, 1/255, (416, 416), (0,        0, 0), True,
crop=False)
        net.setInput(blob)
        outputs = net.forward(output_layers)
        boxes = []
        confidences = []
        class_ids = []
        for output in outputs:
            for detection in output:
                scores = detection[5:]
                class_id = np.argmax(scores)
                confidence = scores[class_id]
                if confidence > conf_threshold and classes[class_id] == "person":
                    center_x = int(detection[0] * frame.shape[1])
```

```
            center_y = int(detection[1] * frame.shape[0])
            w = int(detection[2] * frame.shape[1])
            h = int(detection[3] * frame.shape[0])
            x = int(center_x - w / 2)
            y = int(center_y - h / 2)
            boxes.append((x,y,w,h))
```

Note: This code is using YOLOv3 weights and configuration files, as well as the COCO dataset class names file. Assuiming OpenCV and numpy libraries are installed.

TRACKING A TOY CAR

```
import cv2
import numpy as np
import time

# Initialize webcam
cap = cv2.VideoCapture(0)

# Define color range for small yellow car in HSV color space
lower_yellow = np.array([20, 100, 100])
upper_yellow = np.array([30, 255, 255])

# Initialize variables for tracking small yellow car
prev_x, prev_y, prev_w, prev_h = 0, 0, 0, 0

while True:
    # Capture frame-by-frame
    ret, frame = cap.read()

    # Convert frame to HSV color space
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
   # Create a mask to filter out all colors except the small
yellow car
    mask = cv2.inRange(hsv, lower_yellow, upper_yellow)

    # Apply morphological operations to the mask
    kernel = np.ones((5, 5), np.uint8)
    mask = cv2.erode(mask, kernel, iterations=2)
    mask = cv2.dilate(mask, kernel, iterations=2)

    # Find contours of the small yellow car
    contours, hierarchy = cv2.findContours(mask,
cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

    # Draw bounding box around the small yellow car
    if contours:
        # Find the largest contour
        max_contour = max(contours, key=cv2.contourArea)
        x, y, w, h = cv2.boundingRect(max_contour)
```

```
        # Update tracking variables
        prev_x, prev_y, prev_w, prev_h = x, y, w, h

        # Draw bounding box around the car
      cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

    else:
      # Use previous bounding box coordinates if no contours found
        x, y, w, h = prev_x, prev_y, prev_w, prev_h
      cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

    # Display the resulting frame
    cv2.imshow('frame', frame)

    # Press 'q' to exit
```

DECLARATION

I hereby declare that this thesis is an original report of my research, has been written by me and has not been submitted for any previous degree. The simulation is almost entirely my own work; the collaborative contributions have been indicated clearly and acknowledged. Due references have been provided on all supporting literatures and resources.

I am aware of and understand the university's policy on plagiarism and I certify that this thesis is my own work, expect where indicated by referencing, and the work presented in it has not been submitted in support of another degree or qualification from this or any other university or institute of learning.

_____

HENOK GUTA WOLDEHANNA

MARCH 2023