ADDIS ABABA UNIVERSITY

ADDIS ABABA INSTITUTE OF TECHNOLOGY

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

Adaptive Super Twisting Sliding Mode Controller Design of Quadcopter for Wheat Disease Detection

A thesis submitted to School of Graduate Studies, Addis Ababa Institute of Technology, Addis Ababa University in partial fulfillment of the requirement for the Degree of Master of Science in Electrical Engineering (Control Engineering)

By

Nardos Belay Abera

Advisor

Lebsework Negash Lemma (PhD)

November 29, 2023

Addis Ababa, Ethiopia

Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

Adaptive Super Twisting Sliding Mode Controller of Quadcopter for Wheat Disease Detection

By: Nardos Belay Abera

APPROVED BY BOARD OF EXAMINERS

Name	Signature	Date
Bisrat Derebssa (PhD)		
(School Dean)		
Lebsework Negash (PhD)		
(Advisor)		
Mesifin Tilahun (MSc)		
(Internal Examiner)		
Mengesha Mamo (PhD)		
(External Examiner)		

Declaration

I, the undersigned, declared that this MSc thesis is my original work, has not been presented for fulfillment of a degree in this or any other University and all sources and materials used for the thesis have been duly acknowledged.

Name	**Signature**

Acknowledgment

I would love to specific my heartfelt gratitude and sincerest appreciation to the Almighty God for His infinite blessings throughout this thesis journey. Without His divine intervention and grace, I would not have been able to overcome the challenges and achieve the successful completion of this research. I would also like to extend my deep appreciation to St. Mary, the Mother of Jesus, for her intercession and constant prayers.Besides God, I would like to express my gratitude to my thesis advisor Dr.Lebsework for his invaluable guidance, mentorship, and constructive feedback. Their expertise and dedication have shaped this thesis into a meaningful contribution to the field of study.

I would also like to express my special regards for my family and Friends who supported me throughout my life.

Abstract

Brown wheat rust is a fungal disease that can cause huge destruction in wheat production and quality. Collecting accurate large scale crop data and detecting these diseases based on certain standards through visual inspection is labor intensive, time consuming, and prone to human error. This paper focuses on the design of adaptive super twisting sliding mode controller of a quadcopter for detection of brown wheat rust disease. First, the dynamics of the system was understood then the Newton-quaternion approach was used to model the dynamic system and verified in simulink. Then, the adaptive super twisting sliding mode controller was developed for attitude and position trajectory tracking of a quadrotor. Controller design involves tuning the parameters of the supertwising sliding mode controller using adaptation laws. Comparison of conventional sliding mode controller with the adaptive super twisting sliding mode controller was analyzed. The effectiveness of the proposed control scheme has been verified by developing simulation results for quadcopter in MATLAB/SIMULINK software. The results show high tracking accuracy, chattering reduction, and disturbance rejection capability of the proposed controller. For the task of brown wheat rust detection, transfer learning technique was applied using the state of the art ResNet152v2 model to perform feature extraction for the convolutional neural network architecture. The trained model achieved an accuracy level of 93.28% in the training phase and 92% in the test set.

Keywords: Brown rust, Quadcopter, Adaptive super twisting sliding mode control, ResNet152v2, Convolutional neural network.

Contents

List of Figures

List of Tables

List of Acronyms

ASTMC	Adaptive Super twisting Sliding Mode Controller
COG	Center of Gravity
CW	Clockwise
CCW	Counterclockwise
CNN	Convolutional Neural Network
DOF	Degrees of Freedom
HFOV	Horizontal Field of View
GPS	Global Positioning System
ITAE	Integral Time Absolute Error
LQR	Linear Quadratic Regulator
SMC	Sliding Mode Control
STA	Super Twisting Algorithm
STSMC	Super Twisting Sliding Mode Controller
UAV	Unmanned Aerial Vehicles

Chapter 1

Introduction

1.1 Background

Agriculture is one of the basic needs of human existence as it is directly related to food production. Because food production has a direct connection to basic human requirements and the world's population is growing faster than ever. This means that food supplies must be expanded quickly using existing technologies that minimize labor-intensive tasks, maximize productivity, and reduce environmental impact on food production. Wheat is one of the world's most important staple foods, providing vital nutrients to people worldwide. In Africa, wheat is also a strategic product that generates agricultural income. However, the productivity and quality of the wheat crop is threatened by various diseases from that one is brown wheat rust that can seriously affect the yield and quality of the grain. Brown wheat rust is caused by fungal spores that infects the leaves of wheat plants and it can limit photosynthetic activity, resulting in decreased growth, poor grain quality, and poorer yields. It can spreads via wind distribution and infected plant material. The spores can travel large distances in the wind. Wind currents can transport fungal spores over large distances. This allowed the fungus to spread not only inside a field but also across fields and even regions. Infected plant material during manual inspection of disease can transmit brown wheat rust. Effective and correct detection of wheat rust is crucial for rapid and targeted treatment of the disease. Traditional techniques for detecting wheat rust, such as visual monitoring and physical inspection, have limitations in accuracy, efficiency and lead to spreed of the disease due to physical contact. These techniques often depend on subjective interpretation and may

fail to detect diseases at an early stage, resulting in delayed interventions and greater yield losses [1]. Over the years, significant efforts have been made to develop reliable methods for detecting wheat diseases. Farmers are able to implement strategies for control and take appropriate action when they receive a diagnosis early on. Technological developments have created new opportunities for the effective and automated identification of wheat diseases, particularly in the areas of computer vision and image processing. Precision agriculture is a term used to describe this new field.

Precision agriculture, which is also called smart farming or precision farming, is a kind of agricultural management that uses technology and data analysis to increase the efficiency, sustainability, and productivity of agricultural practices. Using the most recent technological advancements, precision agriculture reduces the use of energy, water, fertilizer, pesticides, and other inputs while increasing crop yields, improving farmer profitability, and minimizing adverse environmental effects. Farmers may carefully monitor and manage the variability in their fields through the use of precision agriculture, including the soil's properties, moisture levels, nutrient content, weed density, and crop health. This enables farmers to use inputs like fertilizers and irrigation only where and when necessary, improving crop performance and better allocating resources. In general, precision farming assists farmers in making data driven decisions and maximizing their farming, reduce costs, increase yields, and minimize negative impacts on the environment, leading to more sustainable and efficient agricultural systems [2].

The use of image recognition for detecting brown wheat rust allow for early detection and classification of wheat diseases, this approach attempts to give farmers the tools they need to act quickly and appropriately. This system can save the time and effort needed for disease monitoring while simultaneously increasing the precision of diagnosis by automating the disease detection process. The findings of this investigation will aid in facilitating the development of a useful and practical system for the early diagnosis of wheat diseases in agricultural settings. Accurate on-site monitoring is a challenging issue when it comes to large-scale operations; hence, this study primarily focuses on the utilization of unmanned aerial vehicles for data collection and detection using the resulting dataset.

Unmanned aerial vehicles (UAVs) have made significant improvements in recent years, with capabilities that include aerial photography, package delivery, and military surveillance. The

controller, which directs and regulates the UAV's flight, is one of the most important parts of the device. The controller is essential in providing stable and accurate flying, which protects the UAV's effectiveness. An in depth understanding of the vehicle's dynamics, control theory, and advanced control algorithms is necessary for the difficult task of designing an efficient controller for UAVs [3].

Precision farming is one of the quadcopter's application areas that this thesis focuses on. By using the quadcopter image platform to capture images of brown wheat rust, more images can potentially be made available to farmers. By keeping an eye on plant health and identifying plant sections that need special care, this aids in image processing. The accuracy of the image taken from the air requires the development of an effective controller. The external wind disturbance is also crucial to the effectiveness and application of the quadrotor in precision agriculture[4] [5].

1.2 Statement of Problem

For wheat crop management and protection, the quick and precise diagnosis of diseases affecting wheat crops is crucial. However, the present techniques for diagnosing diseases in wheat are usually labor-intensive, time-consuming, and prone to error. For farmers and the entire agricultural sector, these difficulties lead to a slower response time, a higher risk of disease transmission, and perhaps lower yields. The issue is the lack of an automated and reliable approach for identifying wheat diseases that can precisely identify and distinguish different diseases impacting wheat crops. As a result, there is a heavy reliance on expert visual inspection; however, not all regions or farming communities may have access to or possess the necessary skills in this area. Furthermore, it is challenging to recognize diseases in their early stages, when treatments are most successful, due to the shortcomings of conventional detection methods. Additionally, the inability to carry out immediate actions due to the lack of real time disease monitoring methods affects total crop production and quality [6]. Consequently, farmers may resort to excessive use of pesticides, resulting in environmental pollution, increased production costs, and potential health hazards. Therefore, there is a pressing need for the development of an accurate, automated, and efficient wheat disease detection system that can identify the specific diseases affecting wheat crops, provide real time monitoring, and enable timely interventions. Such a system would significantly enhance

crop management practices, minimize yield losses, and support sustainable agriculture.

Unmanned aerial vehicle (UAV) control poses a number of difficulties that require effective solutions. Four independently operated propellers build up the quadrocopter. The quadcopter's mobility is dependent on variations in the rotors' speed. To achieve the desired trajectory, create six degrees of freedom by combining three translational motions and three rotational motions to produce the desired trajectory. The quadcopter is underactuated, and nonlinear by nature. Due to its under-actuated properties, the quadcopter is difficult to control directly. It's challenging to regulate all six of these states using only four inputs. Sliding mode control is one of the robust and nonlinear control but it has chattering to reduce the chattering effect super twisting controller proposed and the super twisting controller have parameter tuning problem.

1.3 Objective

1.3.1 General Objective

The main objective of this paper is to design and simulate an adaptive super twisting sliding mode control system for a quadrotor, that is appropriate for monitoring applications with regard to brown wheat rust detection.

1.3.2 Specific Objectives

The specific objectives of this thesis are:

- Modeling a quadcopter system while fully taking into account its nonlinear nature.
- Generate a possible trajectory that the quadcopter could be fed.
- Design an effective ASTSM controller for agricultural drone trajectory tracking.
- Assessing the planned controller's performance in terms of stabilizing the trajectory in the presence of disturbances.
- Complete system simulation using the simulink simulation platform and the MATLAB computational environment.

- Development, testing and analysis of the performance of image recognition.

1.4 Contribution of the Thesis

The adaptive super twisting sliding mode controller can make significant contributions to quadcopter technology for wheat disease detection. Here are some potential contributions:

1. **Enhanced robustness:** In spite of uncertainties and disturbances, the adaptive super twisting sliding mode controller has been developed to provide reliable control of the quadcopter. This means that even in difficult environmental conditions, the quadcopter will be able to maintain its stability and properly carry out its missions.

2. **Fast tracking capabilities:** Due to the fast response time of the adaptive super twisting sliding mode controller, the quadcopter can immediately respond to environmental changes and modify its control actions as necessary.

3. **Gain tuning:** The gain tuner function in an adaptive super twisting sliding mode controller helps to optimize the controller's performance by continuously adjusting the control gains based on the system's conditions in real-time.

4. **Singularity free attitude representation:** The modeling of quadcopter was done using newton-quaternion method.

5. **Brown wheat rust detection:** Help in the early detection and prevention of wheat diseases leading to improved agriculture.

Overall, the ASTSMC can contribute significantly to quadcopter technology for wheat disease detection. These advancements can help in the early detection and prevention of wheat diseases, leading to improved crop yields and general agricultural productivity.

1.5 Scope

The scope of this thesis is the dynamics of quad-copter is studied and a mathematical model was formulated using the Newton-quaternion formulation. Based on the model, a controller is designed. The controller governs the quad-copter to track the reference and use less amount of control effort. The gains of the adaptive super twisting sliding mode controller are tuned by an adaptive scheme for effective stabilization and trajectory tracking. Convolutional Neural Network (CNN) in python programming language is used for image recognition, while the modeling and controller design are built in the MATLAB/Simulink environment. Google colab for python programming is used to simulate the proposed techniques.

However, it's essential to acknowledge a limitation of this thesis. It doesn't include the implementation of a real quadcopter prototype, while covering a lot of theoretical and simulation based investigation. Instead, the research mainly focuses on modeling, control design, and simulation, providing insightful information about these areas in relation to quadcopter control and wheat diseases detection.

1.6 Thesis Outline(Organization)

The complete work is divided into seven separate chapters, including an introductory chapter. More details on the subsequent chapters can be found here. Chapter two of the study of the work of previous researchers and a description of their merits and shortcomings. The third chapter deals with the method of obtaining a nonlinear dynamic modeling of a quadcopter and its verification under Matlab/Simulink. Chapter four on control strategies of the dynamic nonlinear model of the quadcopter, i.e STSMC and adaptive controller. The fifth chapter introduces the details of the image recognition problem and the problem formulation method, as well as the data preparation step. Chapter six provides simulation results and a detailed analysis of the proposed system with image recognition results. In the final chapter, the seventh, directions for future research are summarized and recommended. References used in this thesis are presented in numerical order. Additional concepts that are relevant to this thesis included in appendix.

Chapter 2

Literature Review

This chapter provides literature reviews on wheat diseases detection, quaternion based controller and controllers of UAV.

2.1 Wheat Diseases Detection

Unmanned aerial vehicles (UAVs) have become popular in a variety of fields, including precision agriculture. Drones have the benefits of rapid deployment, high-resolution images, and data collection capabilities when it comes to monitoring plants. Finding plant diseases as early as possible to prevent crop losses is the main issue inside the agricultural zone. Agricultural experts look at the characteristics of diseases using manual recording structures. Many attempts had been made to advance the science of plant disease identification in agriculture in order to solve such issues. The use of image processing methods and approaches in the diagnosis of plant diseases has been the focus of a lot of study. Plant diseases present a serious threat to food safety in [7] [8]. In [9], computer vision was compared with conventional agricultural practices, and computer vision system showed significantly improved accuracy in agricultural systems. Precision agriculture depends extensively on remote sensing methods for the identification of wheat disease. One of the crops that is prone to devastating diseases is wheat [10]. In [11], machine learning applications in wheat rust detection has been discussed. In [12], soil sensor with satellite was used to detect fungal plant disease (powdery mildew, rust, and leaf blight) but this paper had the limitation that the satellites' low resolution data made it difficult to map the severity of the plant disease. They also had maneuverability restrictions and were expensive [13].

2.2 Quaternion Based Controllers

There is a considerable amount of study on quaternion based attitude stabilization. A precise model of a quadcopter's dynamic system is necessary for controlling it. In [14], compared to Euler angles, quaternions offer a fascinating choice for describing the rotations of rigid bodies because they have no discontinuities, gimbals, or locking mechanisms, and mathematical simplicity. In [15], in order to derive control commands for the UAV motor and prevent the gimbal lock that Euler angle based controllers experience, quaternion state feedback has been proposed. Due to the lack of trigonometric parameters, the quaternion implementation also reduces the total complexity of state estimation. In [16], new control laws that are specified using quaternions has been proposed and it assure that the quadrotor will remain stable in all states.

2.3 Quadcopter Control

In [17], new algorithms for the linear quadratic regulator (LQR) controller of a quadcopter were proposed. On the basis of the LQR's pigeon-inspired optimization (PIO) tuning parameters, this study has been proposed. However, because they do not fully represent the vehicle's dynamic attitude behavior, the modeling approaches employed for these linear controllers have an inherent stability issue. The task of controlling the quadcopter is difficult due to the quadcopter's extremely nonlinear dynamics and its design configuration on launch, making flight control of the quadcopter a challenging problem. Sliding mode control (SMC) is in reality considered among one of the non-linear control method, which controls the section trajectory in the direction of a predefined sliding surface and slide on it. Sliding mode control design techniques are often the first choice for systems subject to external disturbances [18]. In [19], sliding mode controller was designed, and the controller was used in a quadcopter simulation. With some uncertainty, the PID sliding surface is taken into account for vertical takeoff and landing aircraft. The chattering effect is the biggest issue with utilizing the sliding mode controller. In [20], apply SMC to attitude control problem with external disturbance. The quadcopter can be stabilized using this method, and it can be moved to any desired position with the appropriate switching function even if it generates disturbance. However, there is a major chattering issue, and the controller gain was manually adjusted.

There are several methods for reducing chattering while using sliding mode control. In [21], the discontinuous control of SMC was changed into the continuous "saturation" or "sigmoid" features. However, this method no longer makes the sliding system trajectories to the sliding surface because sigmoid function is the approximation of so it can reduce the accuracy of the control action. In [22], higher-order sliding mode control techniques was used, which makes sliding surface and its next derivatives to be reset to zero. However, the main problem for higher order sliding mode controllers (HOSMC) is the usage of the time derivatives of a sliding surface. One method to avoid the chattering problem with out usage of the time derivatives of a sliding surface is the popular super twisting algorithm [23]. Aerospace and robotics control systems have all successfully used the reliable nonlinear control technique known as supertwisting sliding mode control. It solves the problem of robustness against uncertainties and disturbances, making it suitable for controlling and navigating drones in difficult environmental conditions. When creating the super twist control law, more control gain results from manual gain tuning [24]. The situations in which quadcopters frequently operate are dynamic, with shifting weather and unidentified disturbances. Adaptive control techniques can be incorporated to tune the gain of the STSMC online to enhance the quadcopter control system's performance and robustness.

Adaptive control is a kind of nonlinear conventional control system that has the ability to change parameters in real time while operating in a dynamic environment. In [25], adaptive control is broadly classified as direct adaptive control in which its parameters are adjusted directly and indirect adaptive control in which parameter estimation and control are performed simultaneously. Currently, adaptive control is primarily employed as a conventional control approach for quadcopter flight. This is made possible by the controller's parameters being able to be changed in a way that allows for automatic adaptation to modifications in flight dynamics. Adaptive control enhances system stability and accuracy by allowing real-time adjusting of STSM control parameters based on the UAV's current operating conditions.

In this thesis, a singularity free adaptive super twisting sliding mode controller of quadcopter is going to applied for the problem of wheat disease detection.

Chapter 3

Mathematical Modeling of Quadcopter

The process of developing and utilizing mathematical techniques and ideas to address issues in the actual world is known as mathematical modeling. To evaluate a system's behavior, understand it, and draw conclusions, one must create a mathematical model of the system. Basic forms for mathematical models include equations, graphs, and computer simulations. Creating mathematical models and equations to describe the dynamics and behavior of a quadcopter system is a component of mathematical modeling for quadcopter controller design. Engineers can use this modeling in order to understand how the quadcopter will respond to various inputs and disturbances. Typically, the mathematical modeling method includes:

I **Kinematics:** The process of computing the quadcopter's position, orientation, velocity, and acceleration from the inputs from its motors and control inputs.

II **Dynamics:** Using physical principles, such as Newton's laws of motion and the forces and torques acting on the system, to describe how the quadcopter moves.

III **Control Systems:** Creating control methods and algorithms that stabilize the quadcopter and allow it to fly in predefined directions or complete specific tasks.

Developing accurate mathematical models helps to simulate the behavior of the quadcopter and use this information to design and optimize the controller to achieve desired performance and stability. These models are also used for testing and validating the controller before implementing it on the actual quadcopter.

3.1 Modeling

3.1.1 General Model Assumptions

Various assumptions are made in the literature to simplify the modeling of the quadcoper. Universal modeling assumptions taken from various literature [26], [27] and adopted in these papers are presented here as follows:

- The propellers of quadcopter is considered rigid.
- The center of gravity of the vehicle coincides with the origin of the body's coordinate system.
- The quadcopter has a symmetric structure and the inertia matrix around this symmetry is diagonal.
- Thrust is proportional to the square of rotor speed.
- Torque is proportional to the square of rotor speed.
- The physical structure of the quadcopter is considered to be a rigid frame built with four propellers.

3.1.2 Quadcopter Flight Configurations

The arrangement of the four rotors on a quadcopter drone is referred to as the quadcopter flight configuration. Four rotors are commonly installed on a quadcopter in a symmetrical design. Different configurations are created to achieve particular performance and flight characteristics. The two most popular quadcopter configurations are: the plus configuration, where the quadcopter is powered by a single rotor, and the cross configuration, where the quadcopter is powered by both rotors.

1. **The quadcopter in cross-configuration**: In this configuration, the motors and propellers are arranged in the shape of an "X" on the frame. Cross configuration in pitch or yaw mode in order to achieve an upward pitch moment, imagine a pitch mode where the two front rotors accelerate and the two rear rotors decelerate. The torque generated by rotating one of the two forward accelerator rotors in a clockwise direction

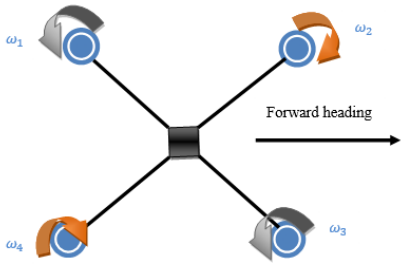


Figure 3.1: Cross configuration

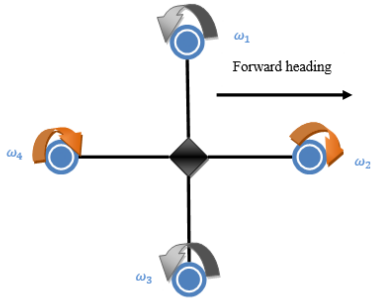


Figure 3.2: Plus configuration

and the other counterclockwise cancels out. The rear rotors must be stopped in the same way. Therefore, pitch control in the cross configuration does not result in a net yaw moment on the quadcopter. To create a right-hand roll moment, the two right rotors slow while the two left rotors accelerate. When the two left rotors revolve in opposite directions, their torques cancel each other out. The quadcopter's roll control in cross configuration does not result in any net yaw moment; therefore, the same holds true for the deceleration of the two right rotors.

2. **The quadcopter in the Plus configuration:** Motor and propeller arranged in shape of '+' and in this case, pitch control mode produces a pitching moment by accelerating the one front rotor and decelerating the single rear rotor. The increase in counterclockwise forward rotor torque does not equal the decrease in counterclockwise tail rotor torque, resulting in a net yaw torque for the plus configuration quadcopter, which must be compensated for by a yaw control input. The single right rotor is decelerated, and the single left rotor is accelerated in the roll control mode to produce a right roll moment in the quadcopter in the plus configuration. The torque generated by the spinning right rotor is greater than the torque generated by the spinning left rotor, but not by an equal amount. Therefore, the quadcopter in the plus configuration experiences an excessive yaw torque when in pitch control mode and roll control mode, which needs to be compensated by the yaw control input. Yaw mode does not provide a pitch or roll moment for the plus and cross quadcopters.

The comparison of the two quadcopter configurations mentioned above is subjective because it depends on the particular use case and needs. However, there are some circumstances in which the cross configuration is better than plus configuration from that one is stability [28].

The equally placed motors and all of its rotors are engaged in operation cross configurations provides greater stability than the plus configuration. Since in case of plus configuration the camera is covered by the quadcopter's frame and cross configuration provides greater stability for this thesis cross configuration is chosen.

3.2 Principle of Quadcopter Operation

Quadcopter work is similar to other types of flight. To enable flight, the engine starts and runs, and the propellers rotate. Four quadcopter propellers can accomplish each of these objectives in accordance with Newton's third law of motion. Air is forced downward as the propellers rotate. The quadcopter is lifted as a result of the reaction force known as lift. The quadcopter has four propellers on the four corners of the frame.

Motions in quadcopter:- A quadcopter is said to be in motion when it moves or changes its position or orientation. A quadcopter can move in a number of ways, such as:

- **Throttle(Hover:)** This motion refers to the vertical movement of the quadcopter, controlling its altitude or vertical position. It is done by simultaneously and equally increasing the speed of all four rotors.

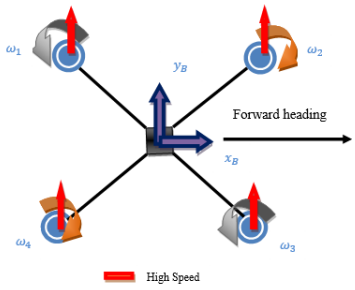


Figure 3.3: Throttle

The quadcopter can fly up or go down, hover at a particular height, or conduct upward or downward maneuvers by controlling the overall thrust of all the motors at once. The quadcopter will begin to ascend vertically when the total force of its four rotors is higher than the weight of the body. The quadcopter will go vertically downward when the total force of its four rotors is less than the force of gravity of the body.

- **Roll:** This motion describes the quadcopter tilting from side to side while rotating

along its longitudinal axis. The quadcopter can roll, change direction, and do aerial maneuvers by adjusting the speed of the motors on the opposite sides which means by decreasing (increasing) the speed of the two propellers on the left and increasing (decreasing) the speed of the remaining propellers on the right, the roll torque can be produced. During a roll command directing the vehicle to navigate in the body reference frame's y-axis.

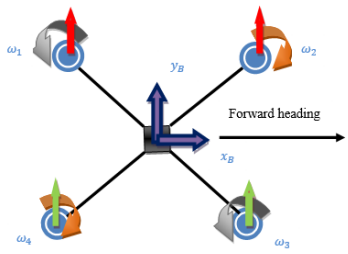


Figure 3.4: Roll

- **pitch:** This movement describes the quadcopter rotating around its lateral axis and tilting forward or backward. The quadcopter can pitch, change its inclination, or conduct actions like flying forward or backward by varying the speed of the motors on the front or back.

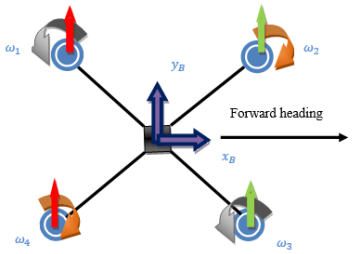


Figure 3.5: Pitch

In order to have pitch movement the two front propellers' speeds are decreased (increased) to create movement in this case, while the speed of the rear propellers is increased (decreased). As a result, the quad-rotor will move in the x direction due to torque generated along the body reference frame's y axis.

- **yaw:** This motion is the left- or right-rotating rotation of the quadcopter around its vertical axis. The quadcopter can yaw and control its heading by varying the speed of the motors on its opposite sides, as well as turn in different directions which means

by increasing (decreasing) each of the opposing pairs of propellers moving in counter-clockwise (clockwise) directions, the yaw movement is produced. The result is that the vehicle rotates around the z axis of the body reference frame as a result of the imbalance in the total torque.

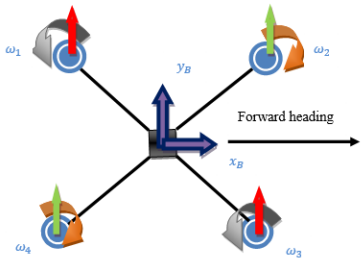


Figure 3.6: Yaw

3.3 Reference Frames Alignment

A reference body is a conceptual body of coordinates that can be used to describe the positioning and motion of objects.

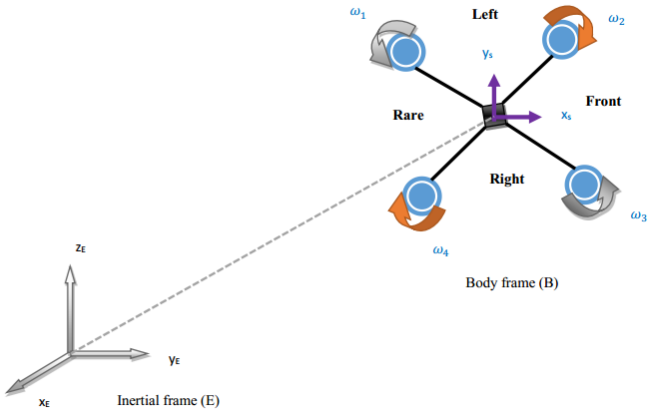


Figure 3.7: The inertial and body coordinates

The coordinate system's direction is determined using the right-hand rule.

- Thumb: Positive direction of x-axis.
- First finger: Positive direction of y-axis.
- Middle Finger: Positive direction of z-axis.

Reference frames used to measure distances, positions, velocities, and other physical characteristics of objects, it serves as a point of reference. Depending on the observer's point of view, various reference frames can be employed to explain the same real-world situation. In quadcopters, reference frame alignment is the procedure used to make sure that the coordinate systems of all the sensors, control systems, and flight controllers are correctly aligned. For accurate control and stable hovering, this alignment is essential. A quadcopter needs the following important parts and sensors to be in alignment [29]:

1. **IMU (Inertial Measurement Unit):** The IMU, which is made up of accelerometers and gyroscopes, collects data on the orientation, changes in motion of the quadcopter and measure linear and angular velocity with respect to body frame. To correctly read and react to the quadcopter's movement, the reference frames of the IMU and the flight controller must be in alignment.

2. **GPS (Global Positioning System):** In order to facilitate autonomous flight, position holding, navigation, some quadcopters come with GPS receivers and measures position with respect to inertial frame. To precisely identify the location and altitude of the quadcopter, the reference frames of the IMU and flight controller must be lined up with those of the GPS receiver.

3. **Newton laws:** Newton laws are valid in inertial frame only.

For an accurate combination of sensors, stable hovering, and accurate flight control, proper reference frame alignment is essential. It ensures safe and effective flight by assisting the quadcopter by properly and precisely interpreting control inputs and outside environmental conditions.

Earth reference frame and body reference frame :- It is typical to employ both the earth reference frame and the body reference frame while navigating or operating a quadcopter. While the body frame enables a local perspective for control and stability adjustments, the earth frame serves as a global reference for position and orientation. The rotation matrix, which serves as the method of transformation needed to obtain the system equations according to a single frame of reference, it is covered in detail in the next section.

1. **Earth (Inertial) reference frame:** refers to a set of fixed coordinates that is station-

ary with respect to the earth's surface. It is used to specify the position and orientation of the quadcopter with respect to the surface of the earth.

- Position vector $[x, y, z]^T$

2. **Body reference frame:** Is a coordinate system that is attached to the quadcopter's body. It rotates and moves along with the quadcopter, enabling measurements in relation to the quadcopter itself.

- Linear velocity ($V^B = [u, v, w]^T$)
- Angular velocity ($\omega^B = [p, q, r]^T$)
- Force (F^B)
- Torque τ^B

3.4 Quaternion

Irish mathematician Sir William Rowan Hamilton created quaternions and the procedures for using them in 1843. A quaternion is a four-dimensional extension of the concept of complex numbers in mathematics. It can be shown as a scalar (real number) and a three-dimensional vector combined. The formal definition of a quaternion is $q = q_0 + q_1 i + q_2 j + q_3 k$, where q_0, q_1, q_2 and q_3 are real numbers, and i, j, and k are the quaternion units. These units satisfy specific multiplication rules, such as $i^2 = j^2 = k^2 = -1$, and $ij = k, jk = i$ and $ki = j$ among others [30]. Due to their capacity to describe three-dimensional rotations, quaternions are frequently utilized in the mathematical modeling of quadcopters. They offer an easy method to carry out computations regarding the quadcopter's rotation and orientation. Quaternions can be used in a quadcopter mathematical model to represent the quadcopter's orientation and rotation in three-dimensional space. For the purposes of control and stability, this is crucial. A four-dimensional vector known as a quaternion can be described as $q = [q_0, q_1, q_2, q_3]$, where q_0 is the scalar component and the others are the vector components.

The procedures below can be used to create a mathematical model for a quadcopter using quaternions:

1. Determine the equations, such as the equations of motion, torque, and thrust, that explain the dynamics of the quadcopter.

2. The rotational transformations and calculations must take the quaternion into account.

3. Run the mathematical model via simulation to examine the quadcopter's behavior under various flight conditions.

It is made simpler to carry out computations including rotations, transformations, and control inputs by employing quaternions in the mathematical model. Additionally, it enables a more accurate representation of the orientation of the quadcopter in three dimensions, improving control and stability.

Quaternion relative to Euler Quaternion relative Euler angles are extremely important for quadcopters because they provide a singularity free of rotations and are free from gimbal lock.

Gimbal lock: is a phenomenon that happens in mathematical modeling when Euler angles are used to represent an object's orientation in three dimensions. Euler angles are used to represent an object's orientation by using three rotations around three perpendicular axes (such as pitch, yaw, and roll). In this representation, gimbal lock happens when two of the axes line up. As a result, the object loses one degree of freedom, and its orientation cannot be uniquely determined using Euler angles. The gimbal lock prevents rotations around these two aligned axes from having separate effects. As a result, the model may exhibit unexpected behavior, such as the object appearing to flip or acting abruptly [31].

Alternative representations, such as quaternions or rotation matrices, can be employed to avoid gimbal lock because they are not affected by this problem. A four-dimensional vector made up of a scalar component and a three dimensional vector component is used to express the rotation when utilizing quaternions. These parts are utilized to carry out rotation operations and encode the rotation information. Quaternion multiplication is used for the rotation operations instead of conventional euler angles or matrix representations in order to eliminate gimbal lock utilizing quaternions. This removes the chance of gimbal lock and permits continuous rotation. In a quadcopter, gimbal lock is the loss of one degree of freedom in the rotating axis of the gimbal system, often brought on by the alignment or

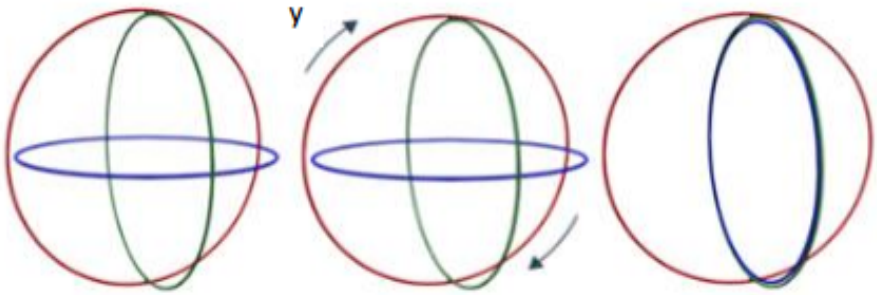


Figure 3.8: Occurrence of gimbal lock when the y axis rotates 90° and the rotation scheme lose one degree of freedom because the x and z become aligned

approaching alignment of the pitch and roll angles. This causes the quadcopter's gimbal mechanism to abruptly change and lose its ability to precisely regulate the rotating axis, which leads to unpredictable or unstable motions. This may result in problems like unstable camera movements or accidental quadcopter rotations. To prevent gimbal lock, in this thesis quaternion based modelling was used.

The properties of quaternions include:

1. **The conjugate and size of a quaternion:** The conjugate property of quaternions states that the conjugate of a quaternion is obtained by changing the sign of its vector part. In other words, the real part of the quaternion remains the same, but the signs of the imaginary parts i, j, and k are flipped.

$$q = q_0 + q_1 i + q_2 j + q_3 k \tag{3.1}$$

$$q^{'} = q_0 - q_1 i - q_2 j - q_3 k \tag{3.2}$$

2. **The magnitude of a quaternion:** Is the square root of the sum of the squares of its components and $\otimes$ is quaternion multiplication symbol. It can be calculated as shown below:-

$$|| \ q \ || = \sqrt{q \otimes \ q^{'}} \tag{3.3}$$

$$= \sqrt{q_0^2 + q_1^2 + {q_2}^2 + {q_3}^2} \tag{3.4}$$

3. **Unit quaternion property of quaternion:** Is when the magnitude of quaternion is

one:-

$$|| q || = 1 \tag{3.5}$$

4. **Quaternion multiplication is not commutative:** When multiplying two quaternions, the order of multiplication matters. This property is expressed as:-

$$q1 \otimes q2 \neq q2 \otimes q1 \tag{3.6}$$

5. **Quaternions are non-associative:** When performing multiple quaternion multiplications, the grouping of terms matters. This means that:-

$$(q_1 \otimes q_2) \otimes q_3 = q_1 \otimes (q_2 \otimes q_3) \tag{3.7}$$

6. **Quaternion inverse:** The inverse of a non-zero quaternion exists and the inverse of a quaternion is same as the multiplication inverse (or $\frac{1}{q}$)
and can be computed as:-

$$q^{-1} = \frac{q^{'}}{(q * q^{'})}$$

If a quaternion has length 1, q is called a unit quaternion and its conjugate become the inverse. q^{-1}=$q^{'}$.

Quaternion representation: A quaternion can be represented as a linear combination of a scalar and a vector with four components. The general form of a quaternion is expressed as: $q = q_0 + q_1 i + q_2 j + q_3 k$ where:- q is the quaternion q_0 is the scalar component q_1, q_2 and q_3 are the vector components , i, j, and k are imaginary units that satisfy certain properties. Therefore, the quaternion q can be represented as a linear combination of its components as: $q = (q_0, (q_1, q_2, q_3))$. In this representation, the scalar component q_0 is separate from the vector component (q_1, q_2, q_3). The vector component is a 3-dimensional vector that represents the direction of rotation or orientation in 3D space. This notation allows for easy manipulation of the quaternion using vector operations.

- As a linear combination ,$q = q_0 + q_1 i + q_2 j + q_3 k$
- As a vector with four coefficients [q_0, q_1, q_2, q_3]
- As a scalar (s, v)

Where : $s = (q_0)$ and , $v = (q_1 , q_2 , q_3)$

3.4.1 Quaternions as Rotary Operators in 3D Space

A given quaternion relates to a 3D spatial orientation. A body in 3D space is rotated when a quaternion's four values are changed by a pair. The concept of rotation using quaternions is based on the fact that a point or coordinate frame can be moved from any initial orientation to any final orientation by performing a single rigid rotation about an axis of a specific unit-length, denoted by the notation $u = u_x + u_y + u_z$, rotated by an angle θ. A pure quaternion is described as a vector in three-dimensional (3D) space by the following definition: $v = 0 + v_1 i + v_2 j + v_3 k$. The next quaternion multiplication, which consists of the right and left multiplication of quaternions, whose explanation is shown in Figure 3.9, can then be used to find the vector's rotated version v'. Where q^* is quaternion transpose.

Figure 3.9: Quaternion as a rotary operator

$$v' = q^* v q = q v q^* \tag{3.8}$$

It is clear that in order to produce a single rotation, the operation necessitates both left and right multiplying the vector by a quaternion q. This is due to the fact that multiplying a vector by a quaternion q rotates the vector only by half of its original angle and visually changes it in 3D space. Therefore, to continue rotating the vector up to the appropriate angle by correcting its changed form, a further multiplication by the inverse quaternion is required. Quaternions can be subject to the same generalized Euler-Rodrigues identity as two-dimensional complex numbers. As a result, the following axis angle representation of the quaternion q can be obtained:

$$e^{\theta u} = (cos\theta + usin\theta) \tag{3.9}$$

$$q = e^{\frac{\theta}{2}u} = (cos\frac{\theta}{2} + usin\frac{\theta}{2}) \tag{3.10}$$

Where u = $u_x + u_y + u_z$ is an arbitrary, unit-length axis of rotation in three dimensional space and θ is the angle of rotation about the specified axis of rotation, u.

3.4.2 Quaternion Rotation Matrix

A quadcopter's orientation or attitude in three-dimensional space is represented by a rotation matrix in the mathematical modeling of a quadcopter. A rotation matrix is a square matrix that represents a rotation in three-dimensional space. It usually has the letter R after it and is 3x3 in size. The rotation matrix in the context of a quadcopter is used to define the rotation of the quadcopter's body frame with respect to the inertial frame. This enables us to determine the quadcopter's position and attitude. The quadcopter's attitude can be represented as a quaternion or as a set of euler angles, from which the rotation matrix can be determined. Vectors from the body frame to the inertial frame can be transformed using the rotation matrix once it has been established. Overall, the rotation matrix is essential for correct estimation and control of the quadcopter's attitude and location. It makes it possible to translate coordinates and vectors between the body and inertial frames, which is essential for the mathematical modeling of quadcopters.

The rotation of the quadcopter is represented by a 3x3 matrix called the quaternion rotation matrix. A series of mathematical formulas are used to create it from the quaternion. Quaternion multiplication can be utilized as a rotation if it is a unit quaternion. However, the transformation is made up of two conventional operations and is conjugate rather than quaternary multiplication. The vector is rotated from one frame to the next as shown on Figure 3.9. The details of the rotation matrix representation can be seen in Appendix A.2, and the rotation matrix can be written as shown below.

$$R_E^B = \begin{bmatrix} q_0^2 + q_1^2 + q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{3.11}$$

3.5 Quadcopter Dynamics

Unmanned aerial vehicles (UAVs) with four rotors are known as quadcopters, and the study of their motion and behavior is known as quadcopter dynamics. Designing controller algorithms to stabilize and control the quadcopter requires a thorough understanding of its dynamics.

The forces and moments exerted on a quadcopter can be utilized to analyze its dynamics. Six degrees of freedom three translational and three rotational can be used to describe the motion of a quadcopter. The quadcopter's x, y, and z coordinates, as well as its u, v, and w velocities, are included in the translational motion. The orientation of the quadcopter, as well as the angular velocities (p, q, and r) and motor specifications, are included in the rotational motion.

Transnational dynamics :- Force affect transnational motion of quadcopter using Newton second law, the forces acting on a quadcopter can be calculated as shown bellow.

Trust Force: The term refers to the upward force produced by the quadcopter's propellers that enables it to take off and maintain flight. The trust force develops when the propellers rotate, creating a pressure differential between the blades' top and bottom surfaces. Air flows downward as a result of the pressure differential, opposing the quadcopter's gravitational pull with an equal and opposite upward push. Trust Force is expressed in body-fixed frame $\sum F_T^B = F_1 + F_2 + F_3 + F_4$.

Gravitational Force: Refers to the downward force exerted on the quadcopter due to the earth's gravity. All objects that possess mass, including quadcopters, are subject to this fundamental force. The gravitational force's strength can be determined using Newton's law of universal gravitation. A quadcopter can't take off into space; the earth's gravitational pull keeps it stationary. It is critical to the balance and stability of the quadcopter when in flight. It only happens in the z direction and expresses itself most naturally in the inertial reference frame of the earth F_g^E ,

$$F_g^E = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \tag{3.12}$$

and using Newton 2nd law: $F = m\vec{a}$ the transitional forces can be described:

$$\sum F^E = F_T^E + F_g^E = m\vec{a} \tag{3.13}$$

In a quadcopter dynamic, the relationship between trust force and angular velocity can be explained by the idea of torque. As the rotational equivalent of force, torque is what causes a rotating body's angular velocity to change. A quadcopter's motors provide trust force, which produces torque that causes the aircraft to rotate. Each motor's trust force is quadratically

related proportional to the angular velocity squared ($F_i = k_f \omega_i^2$). Here, k_f is a constant that depends on a variety of variables, including the torque proportionality constant, the back EMF, the area the propeller sweeps, and the density of the surrounding air. This indicates that the trust force produced by the motors will increase quadratically as the angular velocity increases. On the other hand, a reduction in angular velocity will lead to a reduction in trust force. However, it is important to note that there is a threshold where the motors will not be able to produce enough trust force to sustain flight, resulting in a loss of control. The relationship between trust force and angular velocity is integral to controlling the stability and maneuverability of a quadcopter. Since two reference frames are employed transport theorem has to be employed [32]. The Coriolis terms embeds the information of relative velocity between the two reference frames.

$$F = \frac{dP^B}{dt} + \omega^B \times P^B$$

$$F = m \begin{bmatrix} \dot{u} + q\omega - vr \\ \dot{v} + p\omega - p\omega \\ \dot{\omega} + pv - qu \end{bmatrix}$$

Trust force has quadratic relation with angular velocity i.e

$$F_i = k_f w_i^2 \tag{3.14}$$

$$F_T^B = \begin{bmatrix} 0 \\ 0 \\ F_1 + F_2 + F_3 + F_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ k_f w_1^2 + k_f w_2^2 + k_f w_3^2 + k_f w_4^2 \end{bmatrix} \tag{3.15}$$

Use the rotation matrix R to transfer trust force from the body frame to the earth frame. In a quadcopter dynamic, a rotation matrix is primarily used to transfer trust force from the body frame to the earth frame. This is required because the body frame, a reference frame associated with the quadcopter itself, is typically used to explain the quadcopter's dynamics. It is crucial to change the trust force from the body frame to the earth frame. By offering a mathematical description of the interaction between the two reference frames, the rotation matrix aids in this transformation. The trust force can be described in terms of the coordinates and directions of the Earth frame using the rotation matrix. The trust force can

then be modified in ways that correspond with the Earth frame, making it easier to control and navigate the quadcopter as a result. Overall, the rotation matrix is essential to the dynamics of a quadcopter because it allows the trust force to be transferred from the body frame to the earth frame, improving the quadcopter's ability to be controlled and navigated.

$$F_T^E = R_B^E * F_T^B = \begin{bmatrix} 2(q_1q_3 - q_0q_2)F_T^B \\ 2(q_2q_3 + q_0q_1)F_T^B \\ 2(2(q_0^2 + q_3^2) - 1)F_T^B \end{bmatrix} \tag{3.16}$$

using Newton 2nd law:$F = m\vec{a}$ the transitional forces with respect to earth frame can be described:

$$\begin{bmatrix} m\ddot{x} \\ m\ddot{y} \\ m\ddot{z} \end{bmatrix} = \begin{bmatrix} 2(q_1q_3 - q_0q_2)F_T^B \\ 2(q_2q_3 + q_0q_1)F_T^B \\ 2(2(q_0^2 + q_3^2) - 1)F_T^B \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \tag{3.17}$$

Equation of dynamics of transitional motion with respect to earth frame is described:

$$\ddot{x} = 2(q_1q_3 - q_0q_2)\frac{U_1}{m} \tag{3.18}$$

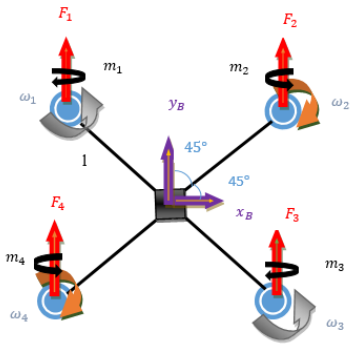
$$\ddot{y} = 2(q_2q_3 + q_0q_1)\frac{U_1}{m} \tag{3.19}$$

$$\ddot{z} = 2(2(q_0^2 + q_3^2) - 1)\frac{U_1}{m} - g \tag{3.20}$$

Rotational dynamics: Quadcopter rotational dynamics is the study of the rotating motion and stability of these quadcopters. For control of a quadcopter's stability, maneuverability, and overall flying performance, it is essential to understand the rotational dynamics of the quadcopters. In quadcopters, torque, moments of inertia, angular momentum, and gyroscopic effect are the four main components of rotational dynamics.

Torque: Torque is a key factor in a quadcopter's rotational dynamics. An object rotates around an axis as a result of the rotating force known as torque. The propellers of a quadcopter are what produce torque. A quadcopter lifts off the ground while in flight due to the thrust produced by the propellers. This thrust force produces torque on the quadcopter body in addition to the upward motion. In order to generate a balanced rotation, a quadcopter has four propellers, each of which spins in the opposite direction. While the other two propellers spin in the opposite direction, there are four total. As the propellers spin, they create a torque due to Newtons third law of motion which states that for every action, there is an equal

and opposite reaction. The quadcopter rotates about its center of mass due to the torque produced by the propellers. The quadcopter uses a mix of changing the speed of each propeller individually and using additional control devices like gyroscopes or accelerometers to regulate rotation. $\sum \tau = (r_i \times F_i)$ The relationship between reaction moments and angular velocity is quadratic $\tau = d\omega_i^2$. The constant d determined based on the number of blades, diameter, pitch, material, and air viscosity of the propellers. Total moment for each coordinate:



$F = k_f\omega_i^2$ and $M_i = d\omega_i^2$ (Reaction moment), For this thesis CW direction is negative and CCW direction is positive.

$$\tau_x^B = -F_1 lcos(45°) + F_2 lcos(45°) + F_3 lcos(45°) - F_4 lcos(45°)$$
$$\tau_y^B = F_1 lsin(45°) + F_2 lsin(45°) - F_3 lsin(45°) - F_4 lsin(45°)$$
$$\tau_z^B = -M_1 + M_2 + M_3 - M_4$$

Where:

- k_f indicates the thrust factor and d indicates the drift factor.
- l = center of quadrotor to center of propeller distance.

Let $\tau_x^B = U_2$, $\tau_y^B = U_3, \tau_z^B = U_4$.

$$U_2 = \frac{\sqrt{2}}{2} lk_f(\omega_1^2 + \omega_2^2 - \omega_3^2 - \omega_4^2) \tag{3.21}$$
$$U_3 = \frac{\sqrt{2}}{2} lk_f(\omega_1^2 - \omega_2^2 - \omega_3^2 + \omega_4^2) \tag{3.22}$$
$$U_4 = -M_1 + M_2 + M_3 - M_4 = -d\omega_1^2 + d\omega_2^2 - d\omega_3^2 + d\omega_4^2 \tag{3.23}$$

The torque equation in the rotation dynamics of a quadcopter is given by : $\tau = I * \alpha$. Where:

- τ is the torque applied to the quadcopter.
- I is the moment of inertia of the quadcopter.
- α is the angular acceleration of the quadcopter.

The quadcopter's rotation can be controlled, and desired flight maneuvers can be accomplished by varying the torque applied to it. This is crucial for balancing the quadcopter during flight and carrying out complex maneuvers. Newton's 2nd law (law of torque balance) states that in an inertial frame of reference, the rate of change of angular momentum L of an arbitrary part of a continuous body is equal to the sum of the torques M effect on this part. It is calculated considering the net torque and the reaction force.

$$\sum \tau_i = \frac{dL}{dt} = I\alpha = \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} + M_{gyro} \tag{3.24}$$

$$\sum \tau_i = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} Roll\,moment \\ Pitch\,moment \\ Yaw\,moment \end{bmatrix} \tag{3.25}$$

Where: I is Inertia matrix, α Angular acceleration ,M_{gyro} Gyroscopic effect.

The gyroscopic effect: The propellers' rotation causes the gyroscopic effect in a quadcopter. The gyroscopic effect is caused by the torque, or rotational force, that the propellers produce as they spin. The phenomenon known as the gyroscopic effect occurs when a rotating object, such as a quadcopter's propellers, resists orientation changes. This means that when the quadcopter tries to move or change its position, the gyroscopic effect causes the rotating propellers to try and maintain their original orientation. For example, if a quadcopter pitches forward to move ahead, the propellers attempt to maintain the original orientation because of the gyroscopic effect, which opposes the forward pitch. This causes a gyroscopic effect, a

destabilizing force, to arise, tilting the quadcopter sideways.

$$M_{gyro} = -\sum_{k=1}^{4} J_{tp}[\begin{pmatrix} p \\ q \\ r \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}](-1)^K w_k \tag{3.26}$$

J_{tp} is the total rotational moment of inertia around the propeller axis.

$$M^B_{gyro} = -\sum_{k=1}^{4} Jtp \begin{bmatrix} p \\ -q \\ r \end{bmatrix} (-1)^k \omega_k = Jtp \begin{bmatrix} q & -q & q & -q \\ -p & p & -p & p \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} \tag{3.27}$$

$$M^B_{gyro} = Jtp \begin{bmatrix} q\omega_1 - q\omega_2 + q\omega_3 - q\omega_4 \\ -p\omega_1 + p\omega_2 - p\omega_3 + p\omega_4 \\ 0 \end{bmatrix}$$

$$\omega_T = -\omega_1 + \omega_2 - \omega_3 + \omega_4 \tag{3.28}$$

$$M^B_{gyro} = Jtp \begin{bmatrix} q\omega_T \\ -p\omega_T \\ 0 \end{bmatrix} \tag{3.29}$$

Angular momentum(L): Is a vector quantity that measures the number of rotational movements an object possesses. Its definition is that it is the cross product of an object's moment of inertia, which measures how resistant it is to rotational motion, and its angular velocity, which measures how quickly it is rotating. An example of this is a quadcopter, which rotates around its center of mass as a result of the rotational force produced by each individual propeller. The quadcopter's angular momentum can explain this rotation. A quadcopter's angular momentum can be influenced by a number of variables, including the propellers' rotational speed and direction, the quadcopter's internal mass distribution, and any external torques operating on it : $L = I\omega$ and $I = mr^2$. Where I is inertia tensor , r is radius from rotation axis, ω rotational motion

$$\omega = \begin{bmatrix} p\, q\, r \end{bmatrix}^T \tag{3.30}$$

Moment of Inertia : Represents how much the body resists the angular acceleration. In quadcopter rotational dynamics, the moment of inertia refers to the strength of an object's resistance to changes in its rotational motion. As a result, when the moment of inertia is high, it is challenging to rotate the mass. It is particular to the way mass is distributed around the quadcopter's center of mass. In simpler terms, it measures how difficult it is to change a quadcopter's rotating movement over a certain distance. The quadcopter's moment of inertia is affected by the mass of each component and how far they are from the center of mass. Inertia tensor of rigid body's is expressed as:-

$$I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \tag{3.31}$$

Where I_{xx} : Is the moment of inertia about the x-axis when the body rotates around the x-axis and I_{xy} : specifies the moment of inertia around the y-axis when the object rotates about the x-axis. For objects like a standard quadcopter (with geometric symmetry) has $I_{xy} = I_{xz} = I_{yz} = 0$.

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{3.32}$$

Angular momentum for the body frame can be calculated as constant inertia tensor times the angular velocity because in the inertial frame the direction of the body will keep changing which means that a constant inertial tensor cannot be used. It has to be calculated every time.

$$\frac{dL}{dt} = \frac{dL}{dt}^{E} + \omega \times L \tag{3.33}$$

The equation that determines the motion relative to the body frame is given by :

$$\tau^{B} = \begin{bmatrix} I_{xx}\dot{p} \\ I_{yy}\dot{q} \\ I_{zz}\dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_{xx}p \\ I_{yy}q \\ I_{zz}r \end{bmatrix} = \begin{bmatrix} I_{xx}\dot{p} \\ I_{yy}\dot{q} \\ I_{zz}\dot{r} \end{bmatrix} + \begin{bmatrix} rqI_{zz} - rqI_{yy} \\ rpI_{xx} - rpI_{zz} \\ pqI_{yy} - pqI_{xx} \end{bmatrix} \tag{3.34}$$

The total torque applied in the body frame is given by:

$$\sum \tau_i = \frac{dL}{dt} = I\alpha = \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} + M_{gyro} = \frac{dL}{dt}^E + \omega \times L$$

$$\begin{bmatrix} I_{xx}\dot{p} + rqI_{zz} - rqI_{yy} \\ I_{yy}\dot{q} + rpI_{xx} - rpI_{zz} \\ I_{zz}\dot{r} + pqI_{yy} - pqI_{xx} \end{bmatrix} = \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} + Jtp \begin{bmatrix} q\omega_T \\ -p\omega_T \\ 0 \end{bmatrix}$$

The dynamic equation of the rotation motion with respect to body reference is written as:

$$\dot{p} = \frac{U_2}{I_x x} - \frac{qr(I_{zz} - I_y y)}{I_x x} + \frac{J_{tp}\omega_T q}{I_x x} \tag{3.35}$$

$$\dot{q} = \frac{U_3}{I_{yy}} - \frac{rp(I_{xx} - I_{zz})}{I_{yy}} + \frac{J_{tp}\omega_T p}{I_{xx}} \tag{3.36}$$

$$\dot{q} = \frac{U_4}{I_{zz}} - \frac{pq(I_{yy} - I_{xx})}{I_{zz}} \tag{3.37}$$

For the above equation to find the earth reference frame expression, quaternion differential equation has to be solved. To solve rate of change of quaternion the following steps should be followed.

$$q = e^{\frac{\theta}{2}u} = (cos\frac{\theta}{2} + usin\frac{\theta}{2})$$

$$\dot{q} = e^{\frac{\theta}{2}u}\frac{u}{2}\dot{\theta}$$

$$\dot{q} = \frac{1}{2}e^{\frac{\theta}{2}u}\dot{\theta}u$$

From this we can see that $e^{\frac{\theta}{2}u}$ is the axis angle representation of a quaternion and $\dot{\theta}$ represents the angular velocity ω.

$$\dot{q} = \frac{1}{2}q \otimes \begin{bmatrix} 0 \\ w \end{bmatrix} \tag{3.38}$$

Then

$$\dot{q} = \frac{1}{2}\begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_o \end{bmatrix}\begin{bmatrix} 0 \\ p \\ q \\ r \end{bmatrix} \tag{3.39}$$

$$\dot{q_0} = \frac{1}{2}(pq_1 + qq_2 + rq_3) \tag{3.40}$$

$$\dot{q_1} = \frac{1}{2}(pq_0 + rq_2 - qq_3) \tag{3.41}$$

$$\dot{q_2} = \frac{1}{2}(qq_0 + pq_3 - rq_1) \tag{3.42}$$

$$\dot{q_3} = \frac{1}{2}(rq_0 + qq_1 - pq_2) \tag{3.43}$$

To represent quadcopter in state space typically includes the 12 state variables and this 12 state variables describe the position, velocity, orientation, and angular velocity of the quadcopter in three dimensions. More detail present on Appendix E. The BLDC motor with 2000 RPM is selected and desired output in the range of [-0.5Nm 0.5Nm]. The transnational and rotational dynamic equation of quad-copter modeling based on newton-quaternion summarized as follows.

$$\begin{cases} \ddot{x} & = 2(q_1q_3 - q_0q_2)U_1/m \\ \ddot{y} & = 2(q_2q_3 + q_0q_1)U_1/m \\ \ddot{z} & = 2(2(q_0^2 + q_3^2) - 1)U_1/m \\ \dot{p} & = \frac{U2}{I_{xx}} - \frac{qr(I_{zz}-I_{yy})}{I_{xx}} + \frac{Jtp\omega_T q}{I_{xx}} \\ \dot{q} & = \frac{U3}{I_{yy}} - \frac{rp(I_{xx}-I_{zz})}{I_{yy}} + \frac{Jtp\omega_T p}{I_{xx}} \\ \dot{q} & = \frac{U4}{I_{zz}} - \frac{pq(I_{yy}-I_{xx})}{I_{zz}} \\ \dot{q_0} & = \frac{1}{2}(pq_1 + qq_2 + rq_3) \\ \dot{q_1} & = \frac{1}{2}(pq_0 + rq_2 - qq_3) \\ \dot{q_2} & = \frac{1}{2}(qq_0 + pq_3 - rq_1) \\ \dot{q_3} & = \frac{1}{2}(rq_0 + qq_1 - pq_2) \end{cases} \tag{3.44}$$

The speeds of the propeller formulated from the control inputs by an inverse relationship as:-

$$\begin{cases} w_1 & = \sqrt{U1/4k_f + U2/4k_fl + U3/4k_fl - U4/4d} \\ w_2 & = \sqrt{U1/4k_f - U2/4k_fl + U3/4k_fl + U4/4d} \\ w_3 & = \sqrt{U1/4k_f - U2/4kf_l - U3/4k_fl - U4/4d} \\ w_4 & = \sqrt{U1/4k_f + U2/4k_fl - U3/4k_fl + U4/4d} \end{cases} \tag{3.45}$$

3.6 Model Verification

Model verification in quadcopter refers to the process of ensuring that the mathematical model used to describe the dynamics and behavior of the quadcopter accurately represents its real-world performance. The quadcopter's behavior is tested then simulated using the model, which is then implemented in the simulation environment MATLAB/Simulink.

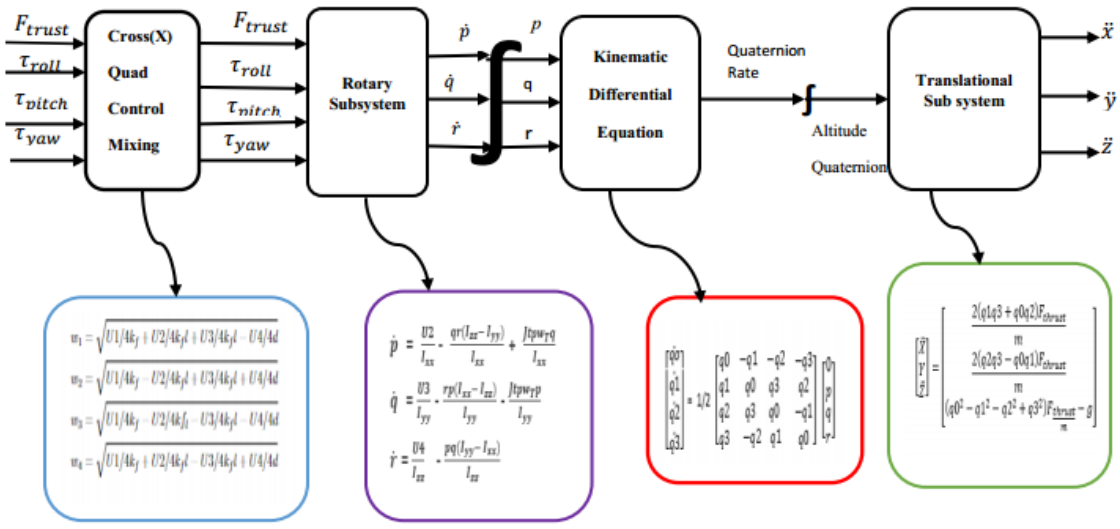


Figure 3.10: Overall mathimatical model of quadcopter

Plant Parameters	Symbol	Unit	Values
k_f	Trust factor	Ns^2	$54.2 * 10^-6$
d	Drag factor	Nms^2	$1.1 * 10^-6$
g	Gravitational Acceleration	m/s^2	9.81
l	Arm length of the quad-rotor	m	0.24
m	Mass of quad-rotor	Kg	1
I_{xx}	Trust factor	Nms^2	$8.1 * 10^-3$
I_{yy}	Trust factor	Nms^2	$8.1 * 10^-3$
I_{zz}	Trust factor	Nms^2	$14.23 * 10^-3$

Table 3.1: Plant Parameter [33]

Hover:- Applying hover control with U_1 equal to the mass times gravity ($U_1 = mg = 9.81N$) causes the quadrotor to move in the vertical Z direction with no attitudinal change, then the quadcopter stays on same height (5m) above the ground. Then the speed is become $\omega_1 = \omega_2 = \omega_3 = \omega_4$.

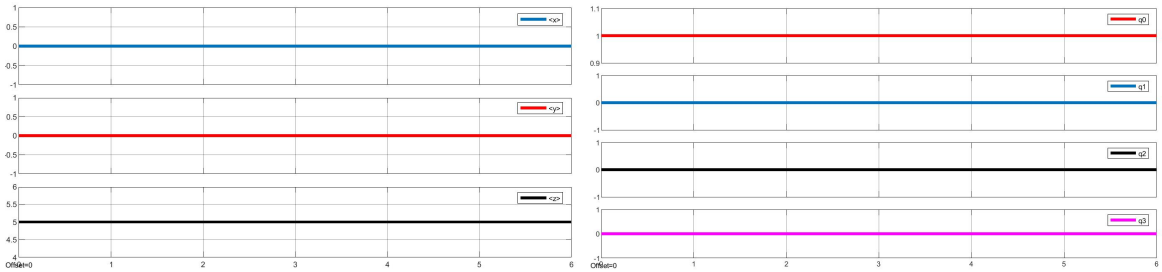


Figure 3.11: Model verification for hovering state output

The MATLAB simulink simulation results for model verification in the quadcopter's hovering state were successful and showed the generated model's accuracy in hovering state. Controlling the quadcopter to keep it in a fixed location in the air was required for the modeling of the hovering state. For applications like aerial photography, where stability and exact location are essential, hovering state is essential.

Roll command:- With a positive forward roll command ($U_2 = 0.1Nm$), the quadcopter will tilt around the x axis (the rotation's quaternion representation is $q_1 = cos(\theta/2) + sin(\theta/2) * i$, where theta is the roll angle and i is the unit vector along the x-axis), causing it to navigate across transitionally in the y direction. When roll command is applied the quadcopter moves towards y-axis and the quadcopter moves down ward along negative z-axis. when (ω_1 and ω_2) > (ω_3 and ω_4) quad-copter moves to the right or moves to negative y-axis. Thus, the form of the altitude quaternion will become: $\mathrm{q} = \begin{bmatrix} q_0 & q_1 & 0 & 0 \end{bmatrix}$.

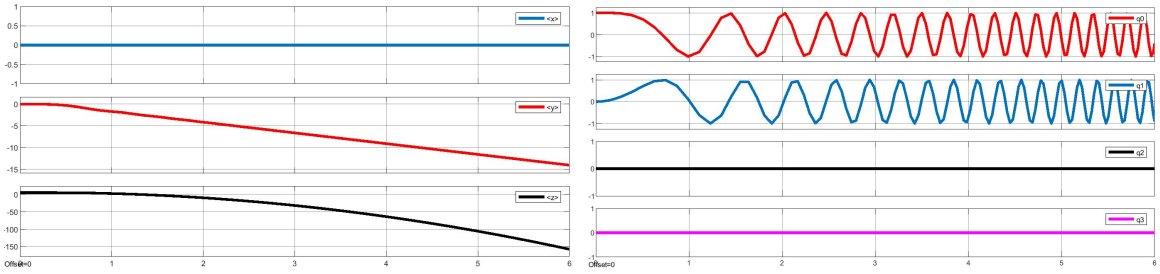


Figure 3.12: Model verification for roll output for both rotational and translational

Note: From the simulation result the quad-copter goes downward in z-direction because of force decomposition. $U_{1y} = U_{new}\ sin\theta$ and $U_1 = U_{new} cos\theta$, where: U_1 = mg(Hovering). The sine wave graph depicts the change in sinusoidal value over time. The sine graph commonly oscillates between -1 and 1, reflecting sine function values ranging from $-\pi/2$ to $\pi/2$. This shows the quadcopter's rotation angle around the x-axis as it varies over time. The sine

graph can be used to determine how the quadcopter rotates around the x-axis.

Pitch command:- With a positive pitch command ($U_2 = 0.1Nm$) applied to the quadcopter, it will tilt around the y axis, causing it to navigate transnationally in the x direction. the quad-copter goes downward in z-direction because of force decomposition $U_x = U_{new}sin\theta$ and $U_1 = U_{new}cos\theta$, where: U_1 =Mg(Hovering) when (ω_1 and ω_4) > (ω_2 and ω_3) quad-copter moves to the forward or moves to positive x-axis. Thus, the form of the altitude quaternion will become: q = $\begin{bmatrix} q_0 & 0 & q_2 & 0 \end{bmatrix}$

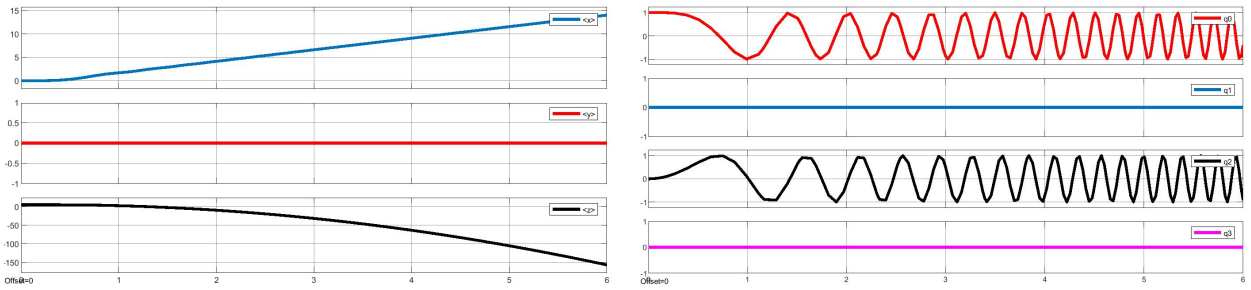


Figure 3.13: Model verification for pitch state output for both rotational and transnational

Yaw:- The positive yaw command given to the quad-rotor shall not change the translational position of the quad-rotor without the expected change in attitude around x and y. When (ω_1 and ω_3) < (ω_2 and ω_4) quad-copter rotates CCW. Thus, The form of the altitude quaternion will become: q = $\begin{bmatrix} q_0 & 0 & 0 & q_3 \end{bmatrix}$

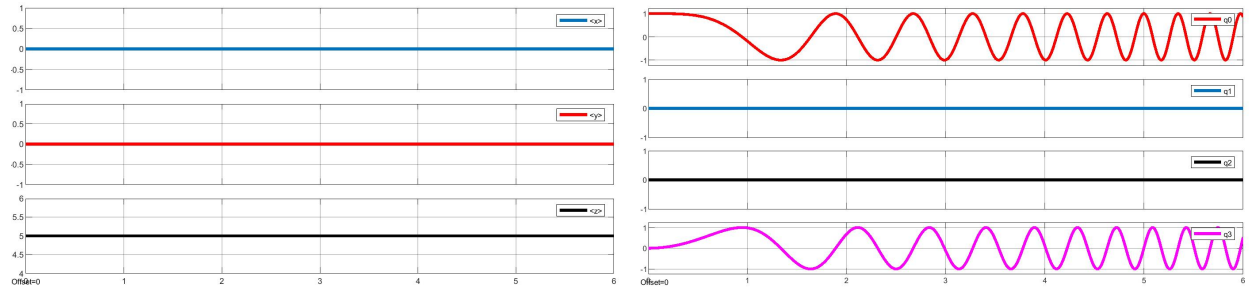


Figure 3.14: Model verification for yaw state output for both rotational and translational

The positions are not affected by yaw movement. The quadcopter rotates CW or CCW based on the command.

Chapter 4

Controller Design

4.1 Overview

Controller design for a quadcopter refers to the process of developing a control system that enables the quadcopter to achieve desired flight trajectory and stability. Designing algorithms and fine-tuning parameters is required to manage the quadcopter's motion and attitude by adjusting its motors or propellers. A quadcopter's controller design is essential for maintaining steady and precise flight performance, which enables the quadcopter to carry out a variety of activities, including aerial photography, surveillance, search and rescue missions.

4.1.1 Adaptive Super Twisting Sliding Mode controller

The first step to design ASTSMC is designing Sliding Mode Control. SMC enables the design of controllers for both linear and nonlinear processes. The concept of the SMC technique is to force the command signal to move to the sliding surface and stay on that surface after the command signal has been reached. In order to guarantee that the system states stay on the sliding surface, the control law is then created. The steps below can be used to create a sliding mode controller for a quadcopter:

1. **Choose a sliding surface:** Sliding surface is a mathematical expression that represents the desired behavior of the system. It should be selected so that the desired control objectives are met when the system states are at the surface. The difference between the desired attitude and the actual attitude, for instance, could be used to determine

the sliding surface for attitude control.

2. **Design the control law:** The sliding function and system dynamics are used to create the control law. The system should stay on the sliding surface because it should be able to force the sliding function to zero.

The chattering, which can harm the actuator, is the biggest issue with this method in sliding mode. A discontinuous control action $U_{dis} = ksign(\sigma)$ best describes this control method. Because it causes poor control precision, excessive wear on moving mechanical parts, and significant heat loss in electrical circuits, the chattering issue is a problem [34]. Controlling the sliding mode in a super-twisted manner is one method for minimizing chattering.

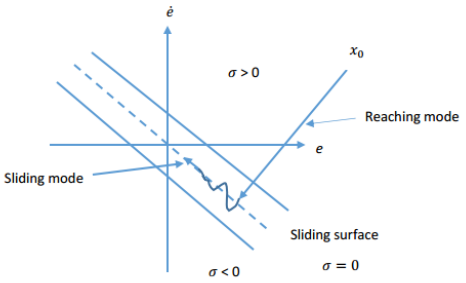


Figure 4.1: Sliding surface

Super twisting sliding mode control (STSMC) is a robust control technique that is commonly used for the control of quadcopters. STSMC is built to manage uncertainties, disturbances, and external perturbations. It is possible to design an STSMC for quadcopters by following the procedures below:

1. **System Modeling:** Begin by modeling the dynamics of the quadcopter. Obtaining the equations that characterize the quadcopter's motion and behavior is part of this process.

2. **Sliding Mode Controller Design:** Implement the super twisting sliding mode controller. The two major components of this controller are a reaching law and a sliding gain.

   Reaching Law: The reaching law is responsible for driving the state of the system onto the sliding surface. It helps in the system's rapid convergence to the sliding surface.

   Sliding Gain: The sliding gain modifies the control input to the system in order to

keep it on the sliding surface. It is a function of the system states and is designed such that the sliding surface is attractive and the system remains stable.

The main disadvantage of the STW control algorithm is that it has gain tuning problem. This gain is difficult to estimate in real-world situations. When developing the STW control law, overestimating the gain limit will lead to larger control gains. Chattering reduction is one of the advantages of super twisting sliding mode control over conventional sliding mode control for quadcopters.

1. **Reduced chattering:** In a sliding mode controller, chattering is the quick switching of control actions that causes high-frequency oscillations in the control signal. This may result in problems including mechanical stress, poor performance, and system damage. This chattering problem is addressed by the supertwisting sliding mode controller by incorporating a continuous nonlinear function during the sliding action. Lower chattering is the outcome of this function's help in reducing switching dynamics while improving slide motion. The supertwisted sliding mode controller improves system dynamics and reduces the chattering effect by introducing a continuous nonlinear function $U_{st} = -\alpha\sqrt{|\ \sigma\ |}sign(\sigma) - \int \frac{\beta}{2} sign(\sigma)$. Where $\alpha\,, \beta$ are super twisting controller gain. It provides a smoother control action while maintaining the robustness and fast response of the conventional sliding mode controller. Overall, the super twisting sliding mode controller improves system performance by removing high-frequency switching and reducing the chattering effect by providing a continuous nonlinear function.

This area of control design focuses on improving the control system's performance and robustness in dynamic and uncertain environments. When using adaptive control, the controller constantly assesses how the system reacts to various inputs. Based on this information, control effectiveness and responsiveness to system dynamics change.

The adaptive super-twisting sliding mode controller is a type of sliding mode control (SMC) that is capable of adjusting and adapting to varying system dynamics. The adaptive super-twisting sliding mode controller's main objective is to stabilize the quadcopter by precisely monitoring the desired trajectory. It accomplishes this by designing a sliding surface that guarantees the desired trajectory.

The adaptive part of the controller comes into play to continuously update the controller

gains depending on the parameter estimates. The conventional sliding mode control method has been modified to create the super-twisting algorithm. It is able to adapt to disturbances and has a higher convergence rate. The super twisting algorithm lessens the chattering effect that is frequently associated with sliding mode control, and the sliding mode controller uses a discontinuous control rule to move the system dynamics toward the sliding surface.

The adaptive super twisting sliding mode controller combines the benefits of the adaptive controller and the super twisting sliding mode controller to improve the controller's performance. STWC substantially reduced chattering's effect. Using an adaptive super twisting sliding mode controller design for a quadcopter has the following benefits:

1. **Robustness:** The adaptive super twisting sliding mode controller offers stable control performance by controlling uncertainties and disturbances in the quadcopter's dynamics and environment. This qualifies it for handling a variety of operational circumstances and outside disturbances that might conflict with the quadcopter's flying.

2. **Fast and accurate response:** The controller design allows for fast response times and accurate tracking of the desired trajectory or control commands. This is crucial for maintaining flight speed and stability, which allows for fine control of the quadcopter's position, speed, and attitude.

3. **Chattering reduction:** Sliding mode controllers are known for chattering, which is a phenomenon where the control signal rapidly switches between different values near the sliding surface. The super twisting algorithm reduces chattering significantly, leading to smoother control signals and smoother motion of the quadcopter.

4. **Gain tuning:** the gain tuner function in an adaptive super twisting sliding mode controller helps to optimize the controller's performance by continuously adjusting the control gains.

Overall, the main contribution of ASTSMC is the application of the STWC scheme with a gain adjustment law without the overestimate gain of control. The value of the gain is difficult to determine and often leads to overestimated values. One way to counteract this is to introduce dynamic control gain adaptation. To do this, the adaptive law must be applied.

4.2 Controller Architecture

The controller architecture for designing a controller for a quadcopter refers to the structure or framework that governs how the various components and subsystems of the controller are organized and interact with each other. A quadcopter has six degrees of freedom (6DOF), which refers to the six separate axes along which it may move and spin. These six degrees of freedom are translation along the x-axis (forward/backward movement), translation along the y-axis (left/right movement), z-axis translation (up/down movement), x-axis rotation (roll), rotation about the y-axis (pitch) and rotation about the z-axis (yaw). Quadcoter has only four input and this makes the quadcopter under-actuated and in order to make fully actuated their must be decoupling.

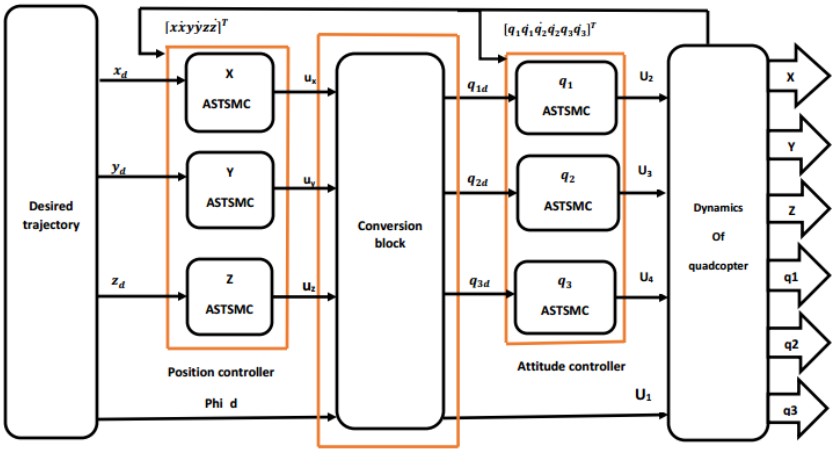


Figure 4.2: Controller architecture

In the context of quadcopter controller design, decoupling refers to the process of separating control of each degree of freedom. In other words, decoupling permits autonomous control of each quadcopter axis while reducing coupling between them so the overall controller is developed by decoupling position and attitude control and designing an ASTSMC for both position controller designed which is then used to identify the desired attitude reference to be fed to the quad-rotor system and the desired quaternion required to get the vehicle to the desired attitude.

4.2.1 Position Controller Design

Designing a control system that enables a quadcopter to precisely retain its intended position in space is known as position controller design. The desired quaternion values for the altitude

controllers are found using the virtual controllers (v_x, v_y, v_z) which are found using the position controllers. As a result, the ASTSMC has a two part design. Designing the sliding surface in the first step will ensure that the sliding motion complies with design requirements. The second is to select a control law that attracts the system state to the switching surface. The next step is design of sliding manifold to design STSMC.

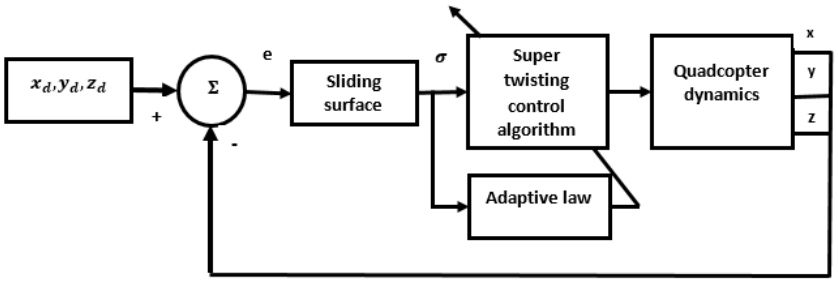


Figure 4.3: ASTSMC of quadcopter position controller architecture

Design sliding manifold: The following procedures should be taken to design a sliding manifold:

1. Determine the quadcopter's desired trajectory.
2. Design a sliding manifold, a function that measures the difference between the quadcopter's actual and desired states.
3. Design control law.

Designing the sliding surface in accordance with the dynamics of the error is the first stage in minimizing the difference between the desired and the actual value, and Stoline and Li proposed the general form of the following equation in [35]:

$$\sigma = (\lambda + \frac{d}{dt})^p e \tag{4.1}$$

where :- e is the tracking error, defined as for x-axis $e(x) = x_d - x$ (the error between the desired trajectory and the actual trajectory in the x-axis), λ is a positive constant that interprets the dynamics of the surface and $p = n - 1$, n: relative degree between output and input.

Design equivalent controller: The equivalent controller is a continuous part of the controller. At the time, the system is in the sliding phase, U_{equ} ensure that they stay there, thus, the system will slide along the sliding surface toward the origin. The U_{equ} controller can be considered to be equivalent control and can be calculated by deriving the sliding surface to zero $\dot{\sigma} = 0$.

Design super twisting sliding mode controller

Super twisting sliding mode controller is given by the equation below:

$$U_{st} = -\alpha\sqrt{\mid \sigma \mid}sign(\sigma) - v \tag{4.2}$$

$$\dot{v} = \frac{\beta}{2}sign(\sigma) \tag{4.3}$$

The total control law (U) calculated by summing the equivalent and supertwisting control law :

$$U = U_{equ} + U_{st} \tag{4.4}$$

Where:- U_{equ} is equivalent controller and U_{st} is super twisting controller. Position error and its derivative can be described as follow:

$$\begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix} = \begin{bmatrix} x_d - x \\ y_d - y \\ z_d - z \end{bmatrix}, \begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_z \end{bmatrix} = \begin{bmatrix} \dot{x}_d - \dot{x} \\ \dot{y}_d - \dot{y} \\ \dot{z}_d - \dot{z} \end{bmatrix}, \begin{bmatrix} \ddot{e}_x \\ \ddot{e}_y \\ \ddot{e}_z \end{bmatrix} = \begin{bmatrix} \ddot{x}_d - \ddot{x} \\ \ddot{y}_d - \ddot{y} \\ \ddot{z}_d - \ddot{z} \end{bmatrix} \tag{4.5}$$

Based on equation 4.1 position controller sliding manifold can be designed :

$$\begin{cases} \sigma_x = \lambda_x e_x + \dot{e_x} \\ \sigma_y = \lambda_x e_y + \dot{e_y} \\ \sigma_z = \lambda_x e_z + \dot{e_z} \end{cases} \tag{4.6}$$

After selecting sliding surface the next step is design equivalent control law and this law ensure the system remains in sliding mode during operation. Equivalent control law can be designed by making $\dot{\sigma_x}, \dot{\sigma_y}, \dot{\sigma_z} = 0$. Equivalent controller law along x-axis is:

$$\begin{aligned} \dot{\sigma_x} &= \lambda_x \dot{e_x} + \ddot{e_x} \\ &= \lambda_x \dot{e_x} + (\ddot{x_d} - \ddot{x}) \end{aligned}$$

From virtual control law we have $\frac{U_{xequ}}{m} = \ddot{x}$ the derivation of virtual controller more discussed

on appendix C:

$$U_{xequ} = \lambda_x \dot{e_x} + \ddot{x_d} \tag{4.7}$$

Then the total STSMC along x-axis is the sum of the equivalent control law equ 4.7 and super twisting sliding mode control signal stated bellow:

$$\begin{aligned} U_x &= U_{xequ} + U_{xst} \\ U_{xst} &= -\alpha_x \sqrt{\mid \sigma_x \mid} sign(\sigma_x) - v_x \\ \dot{v_x} &= \frac{\beta_x}{2} sign(\sigma_x) \\ U_x &= \lambda_x \dot{e_x} + \ddot{x_d} - \alpha_x \sqrt{\mid \sigma_x \mid} sign(\sigma_x) - v_x \end{aligned} \tag{4.8}$$

Where, λ_x is positive definite equivalent control gain and α_x, β_x are super twisting sliding mode control gains which are selected through adaptation law.

Equivalent controller law along y-axis is:

$$\begin{aligned} \dot{\sigma_y} &= \lambda_y \dot{e_y} + \ddot{e_y} \\ &= \lambda_y \dot{e_y} + (\ddot{y_d} - \ddot{y}) \end{aligned}$$

From virtual control law we have $\frac{U_{yequ}}{m} = \ddot{y}$ then:

$$U_{yequ} = \lambda_y \dot{e_y} + \ddot{y_d} \tag{4.9}$$

Then the total STSMC along y-axis is the sum of the equvalent controller equ 4.9 and super twisting control signal stated bellow:

$$\begin{aligned} U_y &= U_{yequ} + U_{yst} \\ U_{yst} &= -\alpha_y \sqrt{\mid \sigma_y \mid} sign(\sigma_y) - v_y \\ \dot{v_y} &= \frac{\beta_y}{2} sig(\sigma_y) \\ U_y &= \lambda_y \dot{e_y} + \ddot{y_d} - \alpha_y \sqrt{\mid \sigma_y \mid} sign(\sigma_y) - v_y \end{aligned} \tag{4.10}$$

Where, λ_y is positive constant equivalent control gain and α_y, β_y are super twisting sliding mode control gains selected through adaptation law.

Equivalent controller law along z-axis is:

$$\begin{aligned} \dot{\sigma_z} &= \lambda_z \dot{e_z} + \ddot{e_z} \\ &= \lambda_z \dot{e_z} + (\ddot{y_z} - \ddot{z}) \end{aligned}$$

From virtual control law we have $\frac{U_{zequ}}{m} = \ddot{z}$ then:

$$U_{zequ} = \lambda_z \dot{e_z} + \ddot{z_d} \tag{4.11}$$

Then the total STSMC along z-axis is the sum of the equivalent controller equ 4.11 and super twisting sliding mode control signal stated bellow:

$$\begin{aligned} U_z &= U_{zequ} + U_{zst} \\ U_z st &= -\alpha_z \sqrt{\mid \sigma_z \mid} sig(\sigma_z) + v_z \\ \dot{v_z} &= \frac{\beta_z}{2} sig(\sigma_z) \\ U_z &= \lambda_z \dot{e_z} + \ddot{z_d} - \alpha_z \sqrt{\mid \sigma_z \mid} sig(\sigma_z) - v_z \end{aligned} \tag{4.12}$$

Where, λ_z is positive constant equivalent control gain and α_z, β_z are super twisting sliding mode control gains selected through adaptation law. Then we have virtual controller (U_x, U_y, U_z) which means that the three common input will control motion along x, y and z and these virtual control are given by equation.

$$\begin{cases} U_x &= \lambda_x \dot{e_x} + \ddot{x_d} - \alpha_x \sqrt{\mid \sigma_x \mid} sign(\sigma_x) - v_x \\ U_y &= \lambda_y \dot{e_y} + \ddot{y_d} - \alpha_y \sqrt{\mid \sigma_y \mid} sign(\sigma_y) - v_y \\ U_z &= \lambda_z \dot{e_z} + \ddot{z_d} - \alpha_z \sqrt{\mid \sigma_z \mid} sign(\sigma_z) - v_z \end{cases} \tag{4.13}$$

These virtual control used to determine desired quaternion value. when $x\,, y$ and z approach x_d , y_d, and z_d as time reaches the sliding surface $\sigma_x = 0, \sigma_y = 0, \sigma_z = 0$ and the design controller derives the actual value to the desired position.

4.3 Decoupling

The main purpose of decoupling in the controller design of a quadcopter is to separate the control loops for the different axes of motion (roll, pitch, and yaw) in order to reduce the interference and coupling effects between them. Each axis dynamics in a quadcopter are interrelated, therefore adjusting the control input for one axis can change how the other axes move. This connection may result in instability, poor control performance, and errors in the desired motion. Each axis can be individually controlled with its own controller by decoupling the control loops. The motion of the quadcopter may now be controlled with greater accuracy and effectiveness. Additionally, it helps in obtaining desired motion profiles, such as keeping the flight level or conducting complex maneuvers. Implementing control

methods and approaches that can successfully separate the control loops for each motion axis involves decoupling in the controller design. Quaternion orientation can be formula as a rotation θ about some axis $\hat{n_d}$ and by utilizing Euler-Rodrigues formula defining the vehicle orientation in terms of the flat outputs become:

$$q_d = \begin{bmatrix} \cos(\frac{\theta_d}{2}) \\ \hat{n_d}\sin(\frac{\theta_d}{2}) \end{bmatrix} \tag{4.14}$$

where $\hat{n_d}$ denotes a unit vector to be used as axis of rotation and θ_d is the angle of the shortest rotation to the desired orientation.

For derivation of the orientation equations from the flat outputs, the normalized thrust vector in the body frame and the earth reference frame will be defined as F^B and F^E, respectively. The normalized body frame thrust vector $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ and the normalized earth reference frame is defined as:

$$\hat{F}^E = \frac{1}{\sqrt{U_x^2 + U_y^2 + U_z^2}} \begin{bmatrix} U_x \\ U_y \\ U_z \end{bmatrix} \tag{4.15}$$

Then, the following definition holds true for the dot and cross product of these normalized vectors.

$$\hat{F}^B \cdot \hat{F}^E = \mid \hat{F}^E \mid\mid \hat{F}^B \mid \cos\theta_d = \cos\theta_d \tag{4.16}$$

$$\hat{F}^B \times \hat{F}^E = \mid \hat{F}^E \mid\mid \hat{F}^B \mid \sin\theta_d \hat{n_d} = \sin\theta_d \hat{n_d} \tag{4.17}$$

$$\hat{n_d} = \frac{\hat{F}^B \times \hat{F}^E}{\sin\theta_d} = \frac{\hat{F}^B \times \hat{F}^E}{\sqrt{1-\cos^2\theta_d}} = \frac{\hat{F}^B \times \hat{F}^E}{\sqrt{1-(\hat{F}^B\hat{F}^E)^2}} \tag{4.18}$$

Equation for the cosine and sine of the half angle of rotation can be derived from cosine and sine values found with inner and cross product. This derivation uses the half angle

trigonometric identity.

$$\cos(\frac{\theta_d}{2}) = \sqrt{1/2(1+\cos\theta_d)} = \sqrt{1/2(1+\cos\theta_d)} = \sqrt{1+\hat{F}^B\hat{F}^E} \tag{4.19}$$

$$\sin(\frac{\theta_d}{2}) = \sqrt{1/2(1+\cos\theta_d)} = \sqrt{1/2(1-\cos\theta_d)} = \sqrt{1-\hat{F}^B\hat{F}^E} \tag{4.20}$$

The desired quaternion value with out yaw become:

$$\hat{q_d} = \frac{1}{\sqrt{2(1+\hat{F}^B\hat{F}^B)}} \begin{bmatrix} 1+\hat{F}^B\hat{F}^E \\ \hat{F}^B \times \hat{F}^E \end{bmatrix} \tag{4.21}$$

The above desired quaternion value corrected for yaw.

$$q_d = \hat{q_d} \times \begin{bmatrix} \cos(\frac{\psi_d}{2}) \\ 0 \\ 0 \\ \sin(\frac{\psi_d}{2}) \end{bmatrix} \tag{4.22}$$

After substituting the value of $\hat{F}^B and\, \hat{F}^E$ we can get the desired quaternion value as follow:

$$U_1 = \sqrt{U_x^2 + U_y^2 + (U_z + mg)^2} \tag{4.23}$$

$$q_{0d} = \frac{1}{2}\sqrt{\frac{U_z+mg}{U_1} - 2sin^2(\psi_d/2) + 1} \tag{4.24}$$

$$q_{1d} = sin(\psi_d/2)U_x + \frac{1}{\sqrt{2}}U_y\sqrt{\frac{U_z+mg}{U_1} - 2sin^2(\psi_d/2) + 1} \tag{4.25}$$

$$q_{2d} = \frac{sin(\psi_d/2)U_y - \frac{1}{\sqrt{2}}U_x\sqrt{\frac{U_z+mg}{U_1} - 2sin^2(\psi_d/2) + 1}}{U_z + mg + U_1} \tag{4.26}$$

$$q_{3d} = sin(\frac{\psi_d}{2}) \tag{4.27}$$

NOTE: The desired quaternion value can also be calculated by rearranging the virtual control inputs.

$$\begin{cases} U_x &= 2(q_{1d}q_{3d} - q_{0d}q_{2d})U_1 \\ U_y &= 2(q_{2d}q_{3d} + q_{0d}q_{1d})U_1 \\ U_z &= (2(q_{0d}^2 + q_{3d}^2) - 1)U_1 - mg \end{cases}$$

The desired angular velocity and U_1 calculation is more discussed on the appendix D and appendix C respectively.

4.3.1 Altitude Controller Design

To design ASTSMC stabilizes the tracking errors of the attitude subsystem by generating the input control signals(U_2, U_3, U_4). The tracking errors are defined:

$$\begin{bmatrix} e_{q1} \\ e_{q2} \\ e_{q3} \end{bmatrix} = \begin{bmatrix} q_{1d} - q_1 \\ q_{2d} - q_2 \\ q_{3d} - q_3 \end{bmatrix} \tag{4.28}$$

Like position control design to design altitude controller first step is designing of sliding:

$$\begin{cases} \sigma_{q1} & = \lambda_{q1} e_{q1} + \dot{e_{q1}} \\ \sigma_{q2} & = \lambda_{q2} e_{q2} + \dot{e_{q2}} \\ \sigma_{q3} & = \lambda_{q3} q_3 + \dot{e_{q3}} \end{cases} \tag{4.29}$$

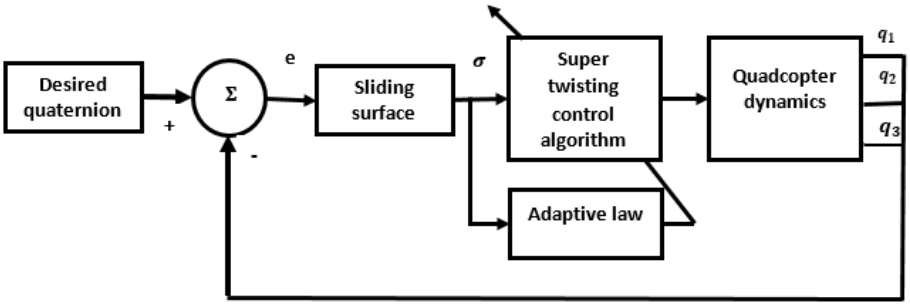


Figure 4.4: ASTSMC of quadcopter altitude controller architecture

After sliding manifold is designed the next step is of designing equivalent control law by making $\dot{\sigma}_q = 0$.

Design ASTSMC for the q_1 control variable U_2

Equivalent control law for q_1 :

$$\begin{aligned} \dot{\sigma_{q1}} &= \lambda_{q1} \dot{e_{q1}} + \ddot{e_{q1}} \\ 0 &= \lambda_{q1} \dot{e_{q1}} + (\ddot{q_{1d}} - \ddot{q_1}) \end{aligned} \tag{4.30}$$

From dynamic equation substitute the value of $\ddot{q_1}$ and the above equation become:

$$0 = (\ddot{q1}_d - (\frac{I_{yy} - I_{zz}}{I_{xx}}\dot{q_1}\dot{q_3} - \frac{J_{tp}}{I_{xx}}\dot{q_1}\omega + \frac{U_2}{I_{xx}})) + \lambda_y \dot{e_q}1$$

$$U_{2equ} = I_{xx}[\lambda_{q1}\dot{e_{q1}} + \ddot{q_{1d}} - (\frac{I_{yy} - I_{zz}}{I_{xx}}\dot{q_1}\dot{q_3} - \frac{J_t}{I_{xx}}\dot{q_1}\omega)] \tag{4.31}$$

The next step is designing of super twisting control law :

$$U_2 = U_{2equ} + U_{2st}$$

$$U_{2st} = -\alpha_{q1}\sqrt{\mid \sigma_{q1} \mid}sign(\sigma_x) - v_{q1}$$

$$\dot{v_{q1}} = \frac{\beta_{q1}}{2}sign(\sigma_{q1})$$

Then the total STSMC is the sum of the equivalent controller 4.31 and super twisting sliding mode control signal stated bellow:

$$U_2 = I_{xx}[\lambda_{q1}\dot{e_{q1}} + \ddot{q_{1d}} - (\frac{I_{yy} - I_{zz}}{I_{xx}}\dot{q_1}\dot{q_3} - \frac{J_{tp}}{I_{xx}}\dot{q_1}\omega)] - \alpha_{q1}\sqrt{\mid \sigma_{q1} \mid}sig(\sigma_{q1}) - v_{q1} \tag{4.32}$$

Designing of ASTSMC for U_3

Equivalent control law for q_2 :

$$\dot{\sigma_{q2}} = \lambda_{q2}\dot{e_{q2}} + \ddot{e_{q2}}$$

$$0 = \lambda_{q2}\dot{e_{q2}} + (\ddot{q_{2d}} - \ddot{q_2}) \tag{4.33}$$

From dynamic equation substitute the value of $\ddot{q_2}$ and the above equation become:

$$0 = (\ddot{q_{2d}} - (\frac{I_{zz} - I_{xx}}{I_{yy}}\dot{q_2}\dot{q_3} + \frac{J_{tp}}{I_{yy}}\dot{q_2}\omega + \frac{U_3}{I_{yy}})) + \lambda_{q2}\dot{e_{q2}} \tag{4.34}$$

$$U_{3equ} = I_{yy}[\lambda_{q2}\dot{e_{q2}} + \ddot{q_{2d}} - (\frac{I_{zz} - I_{xx}}{I_{yy}}\dot{q_2}\dot{q_3} + \frac{J_{tp}}{I_{yy}}\dot{q_2}\omega)] \tag{4.35}$$

The next step is designing of super twisting control law :

$$U_3 = U_{3equ} + U_{3st}$$

$$U_{3st} = -\alpha_{q3}\sqrt{\mid \sigma_{q3} \mid}sig(\sigma_{q3}) - v_{q3}$$

$$\dot{v_{q2}} = \frac{\beta_{q1}}{2}sin(\sigma_{q2})$$

Then the total STSMC is the sum of the equivalent controller equ 4.35 and super twisting control law:

$$U_3 = I_{yy}[\lambda_{q2}\dot{e_{q2}} + \ddot{q_{2d}} - (\frac{I_{zz} - I_{xx}}{I_{yy}}\dot{q_2}\dot{q_3} + \frac{J_{tp}}{I_{yy}}\dot{q_2}\omega)] - \alpha_{q2}\sqrt{\mid \sigma_{q2} \mid}sign(\sigma_{q2}) - v_{q2} \tag{4.36}$$

Designing of ASTSMC for U_4

Equivalent control law for q_3 :

$$\begin{aligned} \dot{\sigma_{q3}} &= \lambda_{q3}\dot{e_{q3}} + \ddot{e_{q3}} \\ 0 &= \lambda_{q3}\dot{e_{q3}} + (\ddot{q_{3d}} - \ddot{q_3}) \end{aligned} \tag{4.37}$$

From dynamic equation substitute the value of $\ddot{q_3}$ and the above equation become:

$$0 = (\ddot{q_{3d}} - (\frac{I_x - I_y}{I_z}\dot{q_1}\dot{q_2} + \frac{U_4}{I_z})) \tag{4.38}$$

$$U_{4equ} = I_z[\lambda_{q3}\dot{e_{q3}} + \ddot{q_{3d}} - (\frac{I_x - I_y}{I_z}\dot{q_1}\dot{q_2}] \tag{4.39}$$

The next step is designing of super twisting control law :

$$\begin{aligned} U_4 &= U_{4equ} + U_{4st} \\ U_{4st} &= -\alpha_{q3}\sqrt{\mid \sigma_{q3} \mid}sign(\sigma_{q3}) - v_{q3} \\ \dot{v_{q3}} &= \frac{\beta_{q3}}{2}sign(\sigma_{q3}) \end{aligned}$$

Then the total STSMC is the sum of the equivalent controller equ 4.39 and super twisting control law as stated bellow:

$$U_4 = I_{zz}[\lambda_{q3}\dot{e_{q3}} + \ddot{q_{3d}} - (\frac{I_{xx} - I_{yy}}{I_{zz}}\dot{q_1}\dot{q_2})] - \alpha_{q3}\sqrt{\mid \sigma_{q3} \mid}sign(\sigma_{q3}) - v_{q3} \tag{4.40}$$

Where, λ_{q1} ,λ_{q2} ,λ_{q3} is positive constant equivalent control gain and α_{q1}, β_{q1} ,α_{q2}, β_{q2} ,α_{q3}, β_{q3} are super twisting sliding mode control gains selected through adaptation law. Stability analysis for STSMC done based on lypanove stability analysis and more discussed on Appendix F.

4.4 Adaptive STSMC Design

Control structure of ASTSMC

The following STW control is considered (Levant, 1993)

$$\begin{aligned} U_{st} &= -\alpha|\sigma|^{\frac{1}{2}}sign(\sigma) + v \\ \dot{v} &= -\frac{\beta}{2}sign(\sigma) \end{aligned}$$

Where the adaptive gains:-

$$\alpha = \alpha(\sigma, \dot{\sigma}, t) \tag{4.41}$$

$$\beta = \beta(\sigma, \dot{\sigma}, t) \tag{4.42}$$

The adaptation law are to be defined:-

$$\dot{\alpha} = \begin{cases} \omega_1 \frac{\sqrt{\gamma}_1}{\sqrt{2}} sign(|\sigma| - \mu), & \text{if } \alpha > \alpha_m \\ \eta, & \text{if } \alpha \leq \alpha_m \end{cases}$$

$$\beta = 2\epsilon\alpha \tag{4.43}$$

where $\epsilon, \gamma_1, \omega, \eta$ are arbitrary positive constants, and the parameter α_m is an arbitrary small positive constant. This method use continuous control without overestimating the control gain to bring the sliding variable σ and its derivative $\dot{\sigma}$ to zero in finite time. In order to do stability analysis of system the following Lyapunov function candidate is introduced.

$$v(\alpha, \beta) = v_0 + \frac{1}{2\gamma_1}(\alpha - \alpha^*)^2 + \frac{1}{2\gamma_2}(\beta - \beta^*)^2 \tag{4.44}$$

Where v_0 is Lyapunov function for $(\sigma, \dot{\sigma})$,α^*, β^* are maximum possible values of α, β respectively. The derivative of the Lyapunov function is obtained as:

$$\dot{v}(\sigma, \alpha, \beta) \leq -\eta_0 \sqrt{v(\sigma, \alpha, \beta)} - |\epsilon_\alpha| \left(\frac{1}{\gamma_1}\dot{\alpha} - \frac{\omega_1}{\sqrt{2\gamma_1}} \right) - |\epsilon_\beta| \left(\frac{1}{\gamma_2}\dot{\beta} - \frac{\omega_2}{\sqrt{2\gamma_2}} \right) \tag{4.45}$$

It gives

$$\dot{v}(\sigma, \alpha, \beta) \leq -\eta_0 \sqrt{v(\sigma, \alpha, \beta)} + \zeta \tag{4.46}$$

With

$$\zeta = -|\epsilon_\alpha| \left(\frac{1}{\gamma_1}\dot{\alpha} - \frac{\omega_1}{\sqrt{2\gamma_1}} \right) - |\epsilon_\beta| \left(\frac{1}{\gamma_2}\dot{\beta} - \frac{\omega_2}{\sqrt{2\gamma_2}} \right) \tag{4.47}$$

Where $\epsilon_\alpha = \alpha - \alpha^*$ and $\epsilon_\beta = \beta - \beta^*$. It can be seen that the finite time convergence of the system is guaranteed [36]. Design parameter of ASTSMC and SMC.

parameter	Values
ω_1	200
α_m	0.01
η	0.01
γ_1	2
μ	0.7

Table 4.1: Design Parameter of adaptaion law

parameter	value
$\lambda_x, \lambda_y, \lambda_z, \lambda_{q1}, \lambda_{q2}, \lambda_{q3}$	10
$\alpha_x, \alpha_y, \alpha_z, \alpha_{q1}, \alpha_{q2}, \alpha_{q3}$	0.01
$\beta_x, \beta_y, \beta_z, \beta_{q1}, \beta_{q2}, \beta_{q3},$	0.002

Table 4.2: Design parameter for both SMC and ASTSMC

Position and altitude ASTSMC of quadcopter summarized as follow :

$$\begin{cases} U_x &= 2(q_{1d}q_{3d} - q_{0d}q_{2d})U_1 \\ U_y &= 2(q_{2d}q_{3d} + q_{0d}q_{1d})U_1 \\ U_z &= (2(q_{0d}^2 + q_{3d}^2) - 1)U_1 - mg \\ U_1 &= \sqrt{U_x^2 + U_y^2 + (U_z + mg)^2} \\ U_2 &= I_{xx}[\lambda_{q1}\dot{e_{q1}} + \ddot{q_{1d}} - (\frac{I_y - I_z}{I_{xx}}\dot{q_1}\dot{q_3} - \frac{J_t}{I_x}\dot{q_1}\omega_T)] - \alpha_{q1}\sqrt{\mid \sigma_{q1} \mid}sign(\sigma_{q1}) + v_{q1} \\ U_3 &= I_{yy}[\lambda_{q2}\dot{e_{q2}} + \ddot{q_{2d}} - (\frac{I_{zz} - I_{xx}}{I_{yy}}\dot{q_2}\dot{q_3} + \frac{J_t}{I_{yy}}\dot{q_2}\omega_T)] - \alpha_{q2}\sqrt{\mid \sigma_{q2} \mid}sign(\sigma_{q2}) + v_{q2} \\ U_4 &= I_{zz}[\lambda_{q3}\dot{e_{q3}} + \ddot{q_{3d}} - (\frac{I_{xx} - I_{yy}}{I_{zz}}\dot{q_1}\dot{q_2})] - \alpha_{q3}\sqrt{\mid \sigma_{q3} \mid}sign(\sigma_{q3}) + v_{q3} \\ \dot{v} &= -\frac{\beta}{2}sign(\sigma) \end{cases} \tag{4.48}$$

Chapter 5

Image Recognition

The study of image recognition, also known as image classification, is a subfield of artificial intelligence that focuses on how well computers recognize and interpret visual data from videos and images. It involves creating algorithms and methods that enable computer systems to recognize specific objects, patterns, or capabilities in images or videos. The goal of image recognition is to replicate the human visual system's capacity to recognize and extract meaningful information from visual data.

Brown rust, also known as leaf rust, is caused by the fungus Puccinia triticina and manifests as small, reddish-brown pustules on the leaves, stems, and grains of infected wheat plants. These pustules contain spores that spread through wind and rain, causing the disease to spread quickly in favorable conditions. Brown rust can reduce grain yield and quality, as well as weaken the overall health and vigor of the plants. The fungus Puccinia striiformis causes yellow rust, sometimes known as stripe rust [37]. Yellowish orange pustules that often follow the veins on the leaves are the disease's outward sign. Yellow rust can spread swiftly and harm wheat harvests, just like brown rust can reduced yield and quality are some of the signs that are comparable to those of brown rust. Both rust infections can be managed using a combination of conventional techniques, such as rotating crops and planting resistant wheat types, and chemical control strategies, like fungicide sprays. Significant diseases of wheat crops include brown rust and yellow rust. Brown rust, on the other hand, is usually considered to be more severe and harmful wheat rust. Brown rust, which can spread quickly and seriously harm wheat crops, can lower yields and lower grain quality. Monitoring the

existence and severity of both diseases is crucial for farmers and researchers to employ prompt management techniques to reduce their negative effects on wheat production. This thesis involves the detection of brown wheat rust.

There are several main steps involved in image recognition for wheat disease detection:

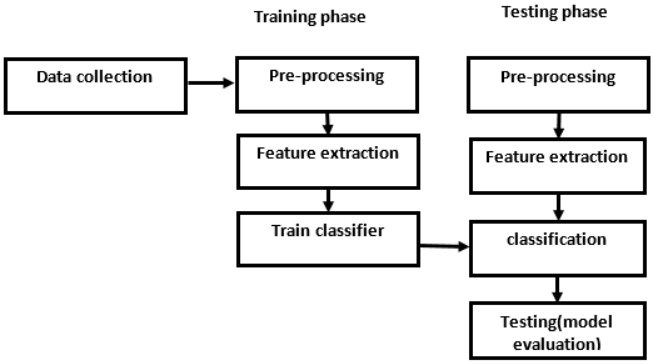


Figure 5.1: Flow chart of image recognition

5.1 Data Collection

Images of wheat plants with various disease stages have been gathered. Cameras or drones with high-resolution cameras can be used for this. The image was gathered via the online data science and machine learning community known as Kaggle. Kaggle allows users to collect and publish data sets. The main weakness in the sources mentioned earlier is that there is image duplication and python code was used to get rid of the duplicated photos. The image dataset must be divided into three data sets after it has been collected:

- **Training data set**: in wheat disease detection refers to a collection of labeled images of healthy wheat plants as well as different types of diseased wheat plants. The images are used to accurately identify and categorize various wheat diseases using machine learning or deep learning models. The dataset is labeled with the appropriate disease classifications in order to assign a label to each class and allow the model to learn the distinct features and patterns associated with each condition. The trained model can then be used to automatically detect and diagnose wheat diseases.

- **Validation dataset**: A validation dataset is another component of the dataset that is utilized during the model-building phase to fine-tune and optimize the performance of

a model. It is used to evaluate various hyperparameters and model improvements in order to increase generalization and avoid overfitting. The validation dataset helps in establishing the best possible model and the appropriate hyperparameters.

- **Test dataset**: Is used to evaluate a trained model's final performance and generalization capacity. It is an accurate measure of how well the model works on previously unknown or new data. To avoid distorted evaluation and overfitting, the testing dataset should not be used throughout the model development process. It provides an estimate of a model's performance and is typically used after model creation and optimization.

(a) Brown wheat rust (b) Healthy wheat (c) Brown wheat rust

Figure 5.2: Sample of wheat image from kaggle data set

5.2 Image Preprocessing

Image preprocessing in wheat disease detection refers to the application of various techniques and methods to enhance raw images of wheat crops affected by diseases. Before feeding them into machine learning or computer vision algorithms for disease classification and detection, the original images must first be modified and manipulated. Image preprocessing aims to optimize the input images through quality enhancement, contrast adjustment, and variability reduction. This aids in removing significant patterns and features from the photos, improving their suitability for precise disease analysis and identification. Image scaling, filtering, and segmentation are just a few of the image preprocessing methods frequently employed in the identification of wheat diseases. With the use of these techniques, diseases can be diagnosed more precisely and reliably because they can improve contrast, lessen background noise, separate distinct areas of the image, and highlight disease-specific characteristics.

5.3 Feature Extraction

The preprocessed images are used to extract important capabilities. Feature extraction is the process of detecting and extracting useful information or features from a pre-processed image. These features in the image can indicate certain patterns, edges, shapes, colors, or structures. The goal of feature extraction is to minimize image complexity by selecting or storing the most selective and useful characteristics while removing unnecessary or duplicate information. In simple terms, image pre-processing focuses on improving image quality and reducing noise before feature extraction. Feature extraction, on the other hand, is concerned with obtaining relevant and representative features from the pre-processed image for further analysis or classification. Convolutional layers in CNNs are specifically designed to perform feature extraction.

5.3.1 The Relation Between Biological and Artificial Neural Network

Neural network is processing system design inspired by the structure and function of human body. Artificial neural network replicate the human brain by creating an artificial human brain. Neurons the main functional structure of human body. Dendrites receive signals, axons transmit signals over long distances, and the soma integrates and creates signals. They work together to allow neurons in the nervous system to communicate and operate.

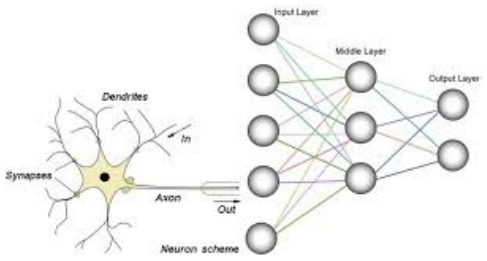


Figure 5.3: Biological and artificial neural network

Biological neuron	Artificial neuron
Cell	Neuron/node
Dendrite(synapse)	Weight
Soma	Net input
Axon	Output

Table 5.1: Neural network

5.3.2 Convolutional Neural Network

The anatomy and operation of the visual cortex in animals served as the basis for CNN design. They are made up of several layers, including pooling, convolutional, and fully connected layers. Different aspects of an image are detected using convolutional layers. To the input data, they apply filters or convolutions to detect patterns like edges and textures. By removing local patterns from the image, these techniques assist in identifying and emphasizing significant elements. Pooling layers downsample the feature maps generated by the convolutional layers. They serve to reduce computational complexity and increase the network's resistance to variations in the input data by reducing the spatial dimension while keeping the most crucial information. Fully connected layers are used at the end of the network to perform classification or regression tasks. They enable the network to understand complicated correlations between the features recognized by the convolutional layers by connecting every neuron in the previous layer to every neuron in the current layer.

A Convolution Neural Network (CNN) in wheat diseases detection refers to a type of neural network architecture that is specifically designed to detect and classify wheat diseases in images. When detecting wheat diseases, a CNN uses a wheat plant input image as input and a sequence of convolutional filters to extract relevant characteristics from the image. These filters help the network identify patterns and structures that are indicative of various diseases that may affect the wheat plant. To get a more compact representation and minimize the spatial dimensions, the collected features then pass through pooling layers. In order to identify the type and degree of severity of the wheat diseases shown in the image, classification or segmentation tasks are carried out on the output before it is fed into one or more fully connected layers. Because they can automatically learn and recognize complicated patterns

and features from large amounts of labeled training data, CNNs are very successful at detecting plant diseases. This enables them to identify and classify wheat diseases with high accuracy and efficiency, enabling farmers and researchers to engage in early disease identification and crop management.

5.4 Training the Model

Selecting a suitable machine learning or deep learning algorithm for the model's training is the first step before training. For this thesis, convolutional neural networks (CNNs) are commonly utilized in the recognition of image tasks and ResNet152v2 model is used. The convolutional neural network (CNN) model ResNet152V2 is a member of the ResNet (Residual Network) family. Deep learning models called residual networks have been found to perform remarkably well in computer vision applications, particularly image classification.

Extension of ResNet152, ResNet152V2 includes 152 layers (152 convolutional layers) placed on top of one another. V2 refers to the second version of the architecture, which introduced some improvements over the original ResNet model. The usage of residual connections is the main idea behind ResNet models. These connections enable the network to learn more effectively and achieve higher accuracy. The obtained features and accompanying disease labels are used to train a machine learning or deep learning model. In order to learn the patterns and links between the images and their respective diseases, this includes splitting the dataset into training and validation sets and feeding the data into the model.

5.5 Model Evaluation and Testing

The validation set is used to test the trained model's performance in accurately identifying wheat diseases. Evaluation may be done using metrics like accuracy, precision, recall, and F1-score. To enhance the performance of the model, hyperparameter tuning or experimenting with various designs may be used. To improve the model's accuracy, strategies like data augmentation, transfer learning, or combination methods might be used. The model can be tested on unseen image to identify wheat diseases after it has been evaluated and optimized.

Chapter 6

Simulation Result

This chapter presents the simulation results of the control system with the MATLAB/Simulink. The mathematical model of the UAV is then extremely useful for simulating and analyzing how the system behaves when the suggested control system is in use. A study simulating the use of the proposed controller for wheat disease detection by UAV was carried out to reduce trajectory errors and the occurrence of uncertainties for trajectory stabilization and monitoring effectiveness. The simulation results are then shown and reviewed to show how well the suggested quadcopter can track the given trajectory along various flight trajectory. Finally, utilizing several trajectories, including the helical trajectories, infinity trajectories, and square wave trajectories assess the validity and effectiveness of the suggested controller. The major goal of the trajectory tracking tests is to demonstrate the effectiveness of the ASTSM controllers.

6.1 Trajectory Generation

For quadcopters, the process of planning and determining a desired flight path for the quadcopter to track is referred to as trajectory generation. In order to produce a smooth and feasible trajectory, it is necessary to determine the positions at various periods in time and used to set the dynamic system's trajectory in a way that is both practical and realistic. The vehicles must pass through specific checkpoints in order to finish a specified flight mission. Planning and generating desirable pathways or trajectories for the quadcopter to follow is the objective of trajectory generation in the controller design of a quadcopter. It involves

carefully figuring out the quadcopter's desired position over a given period of time. The controller may direct the quadcopter to track a specific trajectory or carry out maneuvers by producing the proper trajectories, which can then be monitored. For a number of uses, such as aerial photography, search and rescue, and others, this is essential. The dynamics of the quadcopter, constraints, and mission objectives are all taken into consideration during trajectory generation. To provide safe and effective navigation, it takes factors such as target speed, altitude, waypoints, and the smoothness of the trajectory into account. Polynomials are utilized to calculate the trajectory tracking reference. For smooth, continuous motion with some degree of continuous derivatives, polynomials are natural choices. The order of polynomial in trajectory generation determines the level of continuity required for the trajectory at each derivative level. Third order polynomials are used in this thesis. Steps to do trajectory generation:-

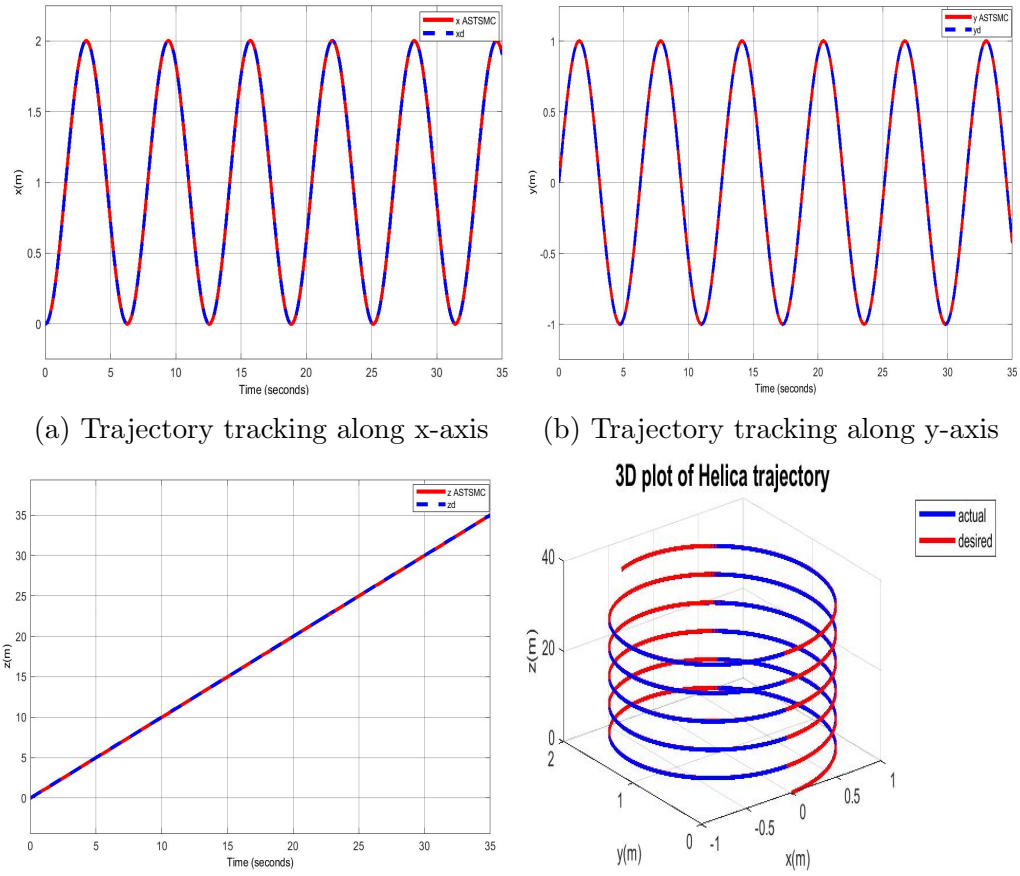
1. Design the way point: Way points refer to locations or positions that a quadcopter must visit or pass through during its flight when creating a trajectory for it. In order to create smooth trajectories that the quadcopter may travel, trajectory generation algorithms for quadcopters often calculate the way points. Polynomial techniques can be used for this. This keeps the quadcopter stable and under control while allowing it to track the desired trajectory. By defining waypoints, it will be possible for the quadcopter to autonomously navigate and carry out its mission.

2. Each waypoint's governing constraint (path constraints and velocity constraints) should be specified.

3. Finding the coefficient of each trajectory x, y and z by solving simultaneous equation for the navigation of quadcopter, the way points are defined manually, the detail for trajectory generation shown in appendix H.

6.2 Helical Input

A helical trajectory refers to a curved path with a vertical component. To determine the performance and efficiency of the quadcopter controller in tracking a specific trajectory, Matlab simulation results utilizing a helical reference input are presented. The simulation enables testing the controller's capability to precisely track a dynamic trajectory by using a helical reference input, which shows a circular or helical path for the quadcopter to track. In this thesis adaptive technique is used to obtain the optimal gain for the ASTSM controller to track the helical trajectory. x_d = 1 - cos(t), y_d =sin(t) and z_d =t. The trajectory tracking performance along the x, y and z axes are shown in Figure 6.1.



(a) Trajectory tracking along x-axis

(b) Trajectory tracking along y-axis

(c) Trajectory tracking along z-axis

(d) 3D Plot of the helical trajectory

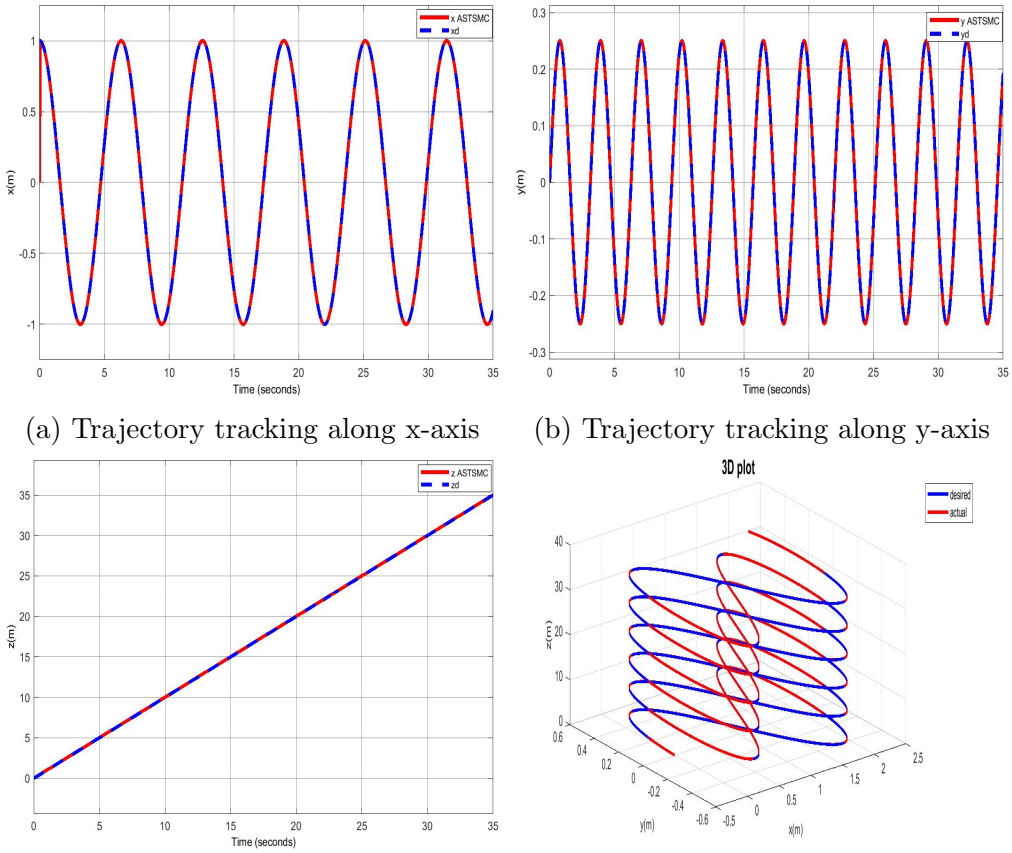
Figure 6.1: Helical trajectory tracking performance of position controller

The simulation results can provide insights into the controller's tracking performance. Additionally, it can help in improving the controller's accuracy and adjusting the controller's parameters to improve tracking performance. The result of position of the 3D Helical trace are shown in the Figure 6.1(d). The tracking positions along x, y and z-axis are presented

in Figure 6.1 from the obtained responses it is shown that the proposed ASTSMC tuned adaptaive scheme is able to guarantee accurate reference tracking.

6.3 Spiral (Infinity) Input

Another trajectory that requires a complex maneuver is the spiral trajectory. The simulation was performed to evaluate the effectiveness and efficiency of the proposed controller in achieving accurate and stable trajectory tracking. The trajectory of Spiral was used as a tracking reference, as follows. The input trajectories along x and y are sinusoidal, i.e. $x_d = 1 - cos(t)$, $y_d = \frac{sin(2t)}{2}$ and $z_d = t$.



(a) Trajectory tracking along x-axis

(b) Trajectory tracking along y-axis

(c) Trajectory tracking along z-axis

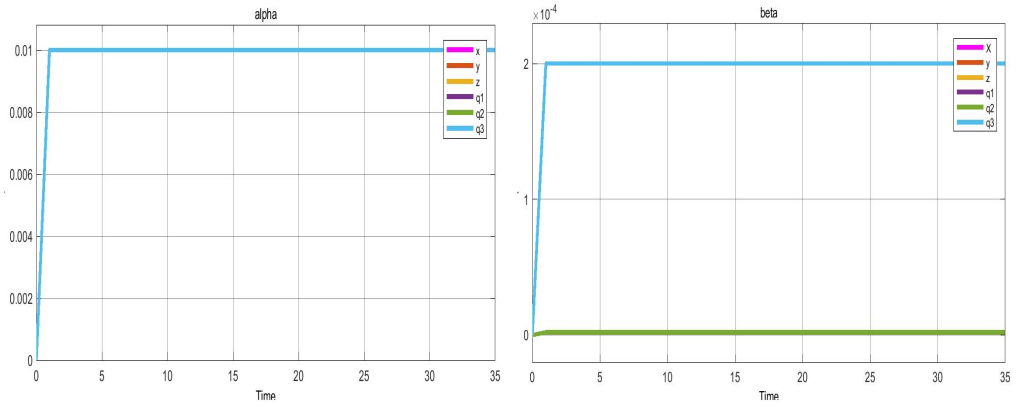
(d) 3D plot of spiral input

Figure 6.2: Trajectory tracking performance of position controller

The simulation results demonstrated the superior performance of the proposed adaptive super twisting sliding mode controller in achieving spiral trajectory tracking for the quadcopter. The quadcopter was able to accurately follow a predefined trajectory without any significant deviation. The controller demonstrated good trajectory tracking performance, validating its adaptability and robustness.

6.4 Adaptive Gain in ASTSMC

The adaptive gain in the ASTSMC is a key element in adjusting the controller's sliding surface to counteract uncertainties and system dynamics, while ensuring stability and control accuracy. The performance of the controller can be improved by adjusting the adaptive gain, enabling reliable and accurate control under various circumstances. The adaptive nature of ASTSMC offers better tracking and disturbance rejection properties than conversational sliding mode controllers, which demand predefined, fixed gains. The controller's parameters have been adjusted as (this adjustment was made to obtain high performances) based on [36] table of parameter selected discussed on Appendix I.



(a) alpha adaptation gain (b) Beta adaptation gain

Figure 6.3: Adaptation gain for square wave ASTSMC

- **Initial Increase in Adaptive Gain:** During the initial stages of the simulation, the adaptive gain dynamically adjusts based on the desired output and the actual output of the quadcopter. To lower the overall control error and guarantee the stability of the system, this adjustment process is crucial. In real-world circumstances, adaptive gains frequently rise initially to promote error convergence and quick system dynamics adaptation. The initial increase in adaptive gain helps the controller overcome the initial disturbances and uncertainties by increasing the control action, thereby lowering the error.

- **Constant Adaptive Gain:** After the initial increase, the adaptive gain reaches a constant value and remains unchanged for the duration of the simulation. This occurrence happens as a result of the controller's capability to precisely estimate and monitor the system parameters. This indicate the controller has correctly adapted to

the dynamics of the quadcopter and can continue to maintain stable flying without making further adjustments. The adaptive gain eventually reaches an ideal value that preserves stability and control precision once the system dynamics and uncertainties have been successfully captured and compensated for during the initial phase.

6.5 Square Wave Input

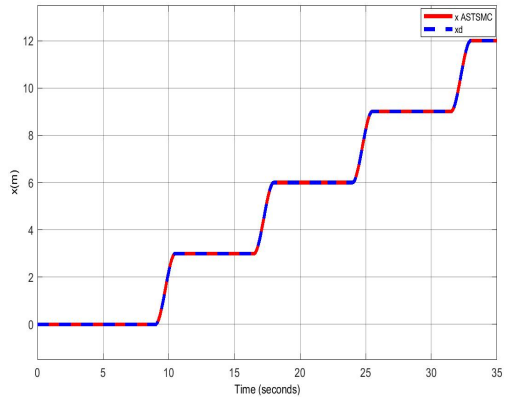
To make application based trajectory first spectral camera should be selected and its specification should be considered and for this thesis the selected multispectral camera is wheatScan Pro [38]. Since the selected camera HFOV is 26.8 °, To meet this specfication when the quadcopter is at 6m the selected horizontal coverage distance is 12m to generate square wave trajectory. For the vegetation image analysis, Ground sample distance (GSD) of 1-2 cm/pixel is recommended the discussion is on Appendix G. The capabilities of unmanned aerial vehicles (UAVs) offer the opportunity to capture the image platforms for real-time agricultural monitoring and use those trajectories to cover an entire area. A navigation speed of 2 m/s was taken to tabulate flight times.

Time	z	x	y
0	0	0	0
3	6	0	0
9	6	0	0
10.5	6	3	12
16.5	6	3	0
18	6	6	12
24	6	6	12
25.5	6	9	12
31.5	6	9	0
33	6	12	0

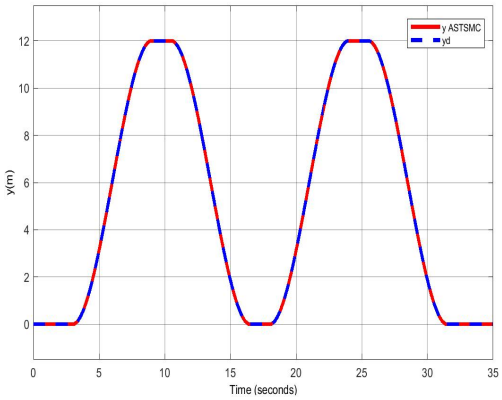
Table 6.1: Flight time table

By employing a square wave trajectory, which typically consists of alternate periods the controller is tested for its ability to track and maintain the desired position and altitude of the quadcopter. This kind of trajectory is particularly helpful for assessing the controller's ability

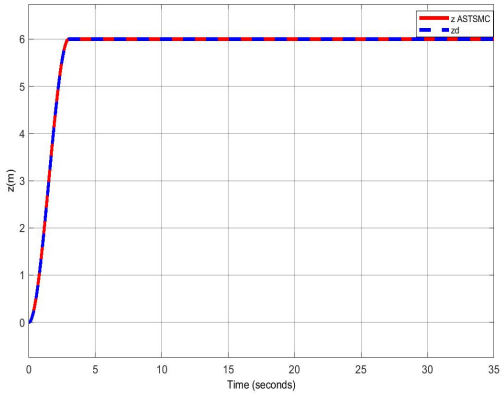
to manage changes in direction and transitions. When designing a quadcopter controller, square wave trajectory tracking is used to test and improve the controller's capacity to accurately track and maintain a predefined trajectory, providing accurate and reliable flying performance.



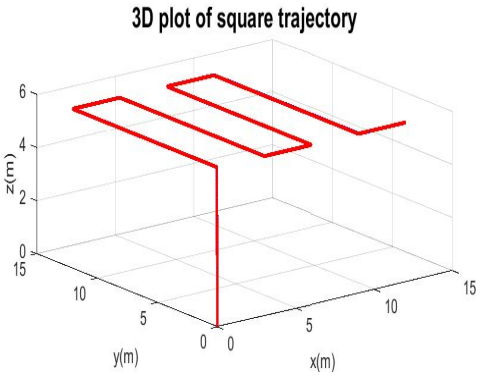
(a) Trajectory tracking along x axis



(b) Trajectory tracking along y axis



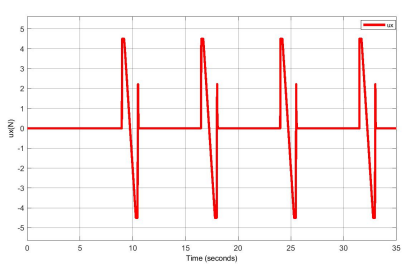
(c) Trajectory tracking along z axis



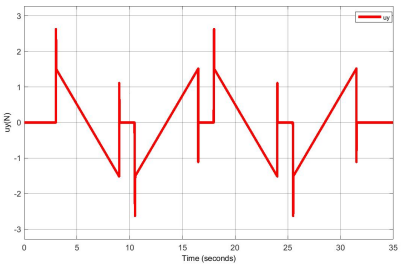
(d) 3D plot of square wave input

Figure 6.4: Square wave trajectory tracking performance of position controller

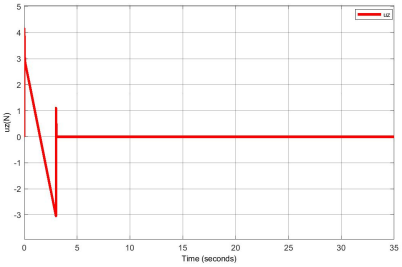
It can be noticeably observed from the above figure the ASTSM auto tuned controllers using adaptation law have good performances for positions (x, y, z) control.



(a) Virtual controller along x



(b) Virtual controller along y

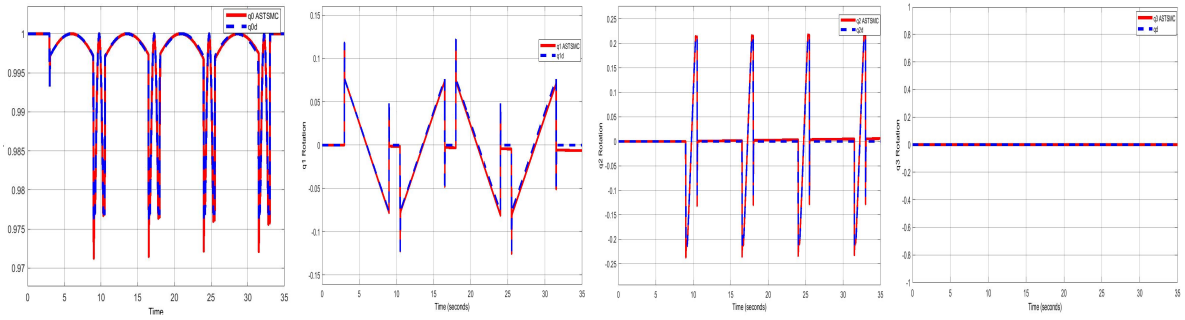


(c) Virtual controller along z

Figure 6.5: Virtual controller for position square wave trajectory

Virtual controller U_x, U_y and U_z are used to control translational dynamics along x,y,z axis.

Responses for attitudes controllers (q_0, q_1, q_2, q_3) are illustrated in figure 6.6:

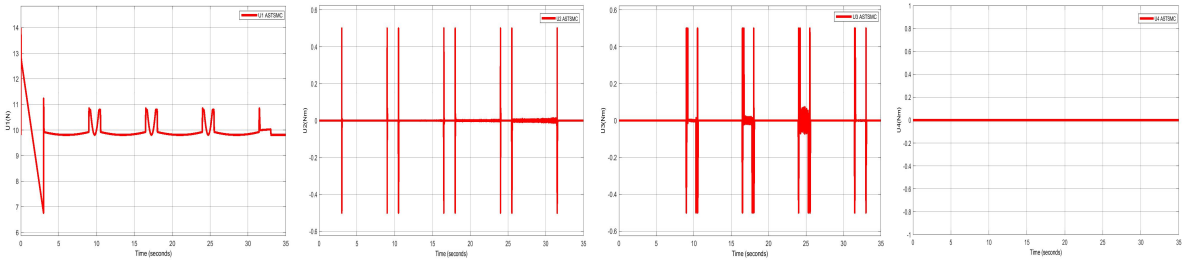


(a) q0 rotation (b) q1 rotation (c) q2 rotation (d) q3 rotation

Figure 6.6: Trajectory tracking performance of altitude controller

The simulation results demonstrate the effectiveness of the adaptive super twisting sliding mode controller in achieving square wave trajectory tracking for the quadcopter. Throughout the trajectory, the controller ensures precise tracking and smooth motion. During the simulation, the adaptive super-twisting sliding mode controller exhibits good tracking performance. The controller maintains accurate tracking throughout the trajectory and quickly adjusts to changes in the trajectory. We can see that the desired quaternion produced by the control system's virtual controller. virtual controller signals U_x, U_y, and U_z which are used to determine the desired quaternion value and attitude controller U_1 are shown in Figure 6.7. It can be noticeably observed from Figure 6.6 that the ASTSMC auto tuned controllers have good performances for positions (x, y,z) control and altitude control.

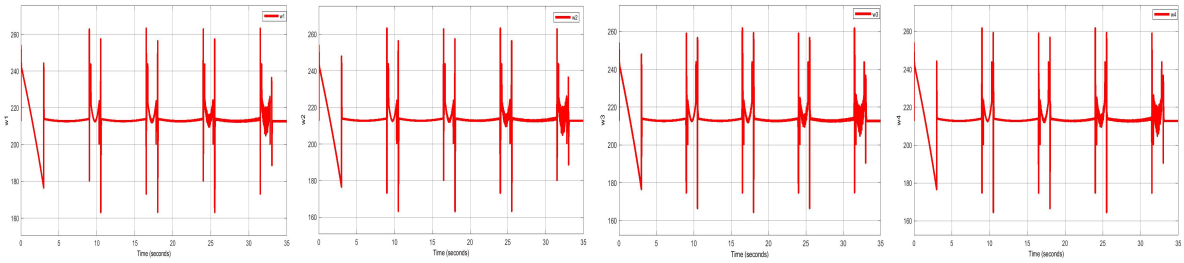
Control input for square wave trajectory generation:- The controller signal amplitude is vary based on the input trajectory. The simulation results of control input can be used to evaluate the energy efficiency of the control algorithm. By analyzing the thrust and moments exerted by the quadcopter's propellers, it can be determined if the control algorithm is optimized to minimize energy consumption. A well-optimized control algorithm will aim to minimize unnecessary control inputs, reducing power consumption and improving flight endurance.



(a) Altitude controller (b) Roll controller (c) pitch controller (d) Yaw controller

Figure 6.7: The control input for square wave trajectory generation

The simulation results demonstrate that the quadcopter is able to closely follow the square wave trajectory with minimum tracking error. To ensure that the quadcopter's position and orientation match the required trajectory, the control inputs dynamically adapt to the changing requirements of the trajectory. The simulation results show that the adaptive super twisting sliding mode control produces smooth control inputs and the amount of control effort is minimal. This indicates that the control algorithm is efficient and consumes less energy to achieve the desired trajectory tracking. The next section discusses how the adaptive super twisting sliding mode control can handle system uncertainties and disturbances, which is a key feature. The amount of control effort and energy consumed by quadcopters is a crucial additional issue to take into account. The speed command represents the desired motor speed, and it is a critical input to the controller. By testing extreme speed commands, it was determined that the motor had physical limitations in terms of its response range. Extremely high speed commands caused the motor to reach its maximum capacity, potentially leading to motor saturation or even failure. Conversely, very low speed commands might fall below the motor's minimum operational limit, leading to instability or a lack of control authority.



(a) ω_1 (b) ω_2 (c) ω_3 (d) ω_4

Figure 6.8: Speed commanded to motors for square wave

The simulation results show that as the speed command increases, the motor responds by increasing its rotational speed accordingly (assuming the controller is designed correctly). This direct relationship between the speed command and the motor speed is crucial for

achieving the desired flight behavior of the quadcopter. Speed commands that introduced gradual changes in motor speed led to smooth acceleration or deceleration of the quadcopter. The motor response featured transient periods of adjustment until the desired speed was achieved. This discovery made it clear how crucial it is to take into account the motor's response time in relation to the dynamic behavior of the quadcopter.

6.6 Operation in Disturbed Environment

A disturbance in the controller design of a quadcopter refers to any external influence or force that affects the stability or performance of the quadcopter. Wind gusts is an examples of the factors that might cause disturbances. The quadcopter's controller is designed to reduce the effects of windy environment and maintain stable flight dynamics. The effectiveness of the suggested controller is evaluated in the presence of disturbances in this section. The Figure 6.9 shows that the disturbance is added from $t = 20sec$ up to $t = 30sec$ from time $t = 20 - 25$ The applied disturbance $\tau_{dx} = \tau_{dy} = 17.6 - 2.4 * t + 0.108 * t^2 - 0.0016 * t^3$ and from $t = 25 - 30$, 0.3NM assumed to act to the roll and pitch direction.

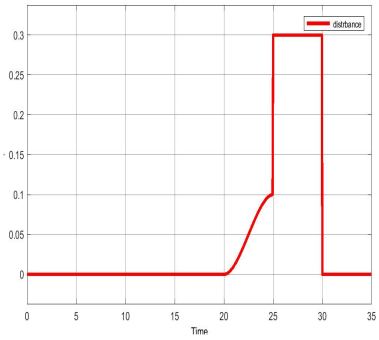
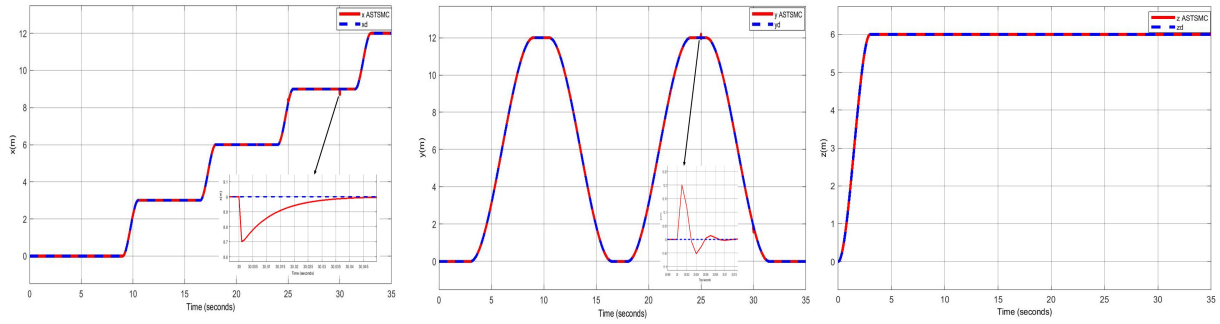


Figure 6.9: Applied disturbance

The effect of disturbance on square wave input adaptive super twisting sliding mode controller for a quadcopter:-

1. **Adaptation Delay:** The adaptation process of the controller introduce a delay in response to the disturbance. This delay can lead to temporary performance degradation until the controller adapts to the disturbance and stabilizes the system. The adaptation delay minimized to ensure accurate tracking and stability during disturbances. The following Figure 6.10 show the effect in adaptation delay and its minimization. Despite the large bounded disturbance imposed on the quadrotor system, it was found that this

was fully compensated for by the adaptive gain tuning in ASTSMC.



(a) x axis with disturbance (b) y axis with disturbance (c) z axis with disturbance

Figure 6.10: Adaptation delay

From the simulation result we can see that even if we apply higher magnitude of force as disturbance on the quadcopter, ASTSMC reject the disturbance and the errors converge to zero.

2. **Control effort increase:** The disturbance makes it more difficult for the controller to maintain stability and precisely track the specified trajectory. To account for the disturbance, the controller must continuously change the control inputs, which increases energy use.

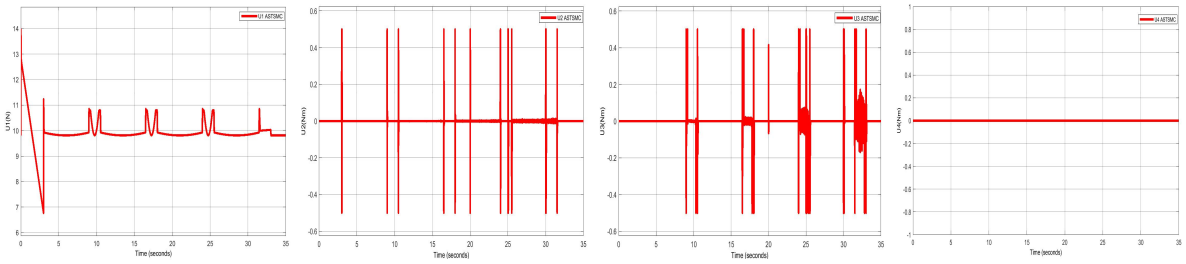


Figure 6.11: Disturbance effect on controller effort

6.7 Effect of Uncertainty

The response of the quadcopter model under various uncertain conditions can be examined to assess the controller's ability to adjust to changes in the parameters. Uncertainty tolerance is performed for the problem of trajectory tracking of the square wave input while parametric uncertainties described in Table 6.2 are imposed into the overall system. It is assumed that the inertial tensors value is five times that of design time. The ability of the suggested controller to return performance to the design level is examined.

Imposed uncertainty	Values
parametric uncertainty	$I_{xx} = 5 * I_{xn}$, $I_{xx} = 5 * I_{yn}$, $I_{zz} = 5 * I_{zn}$

Table 6.2: Imposed Uncertainties

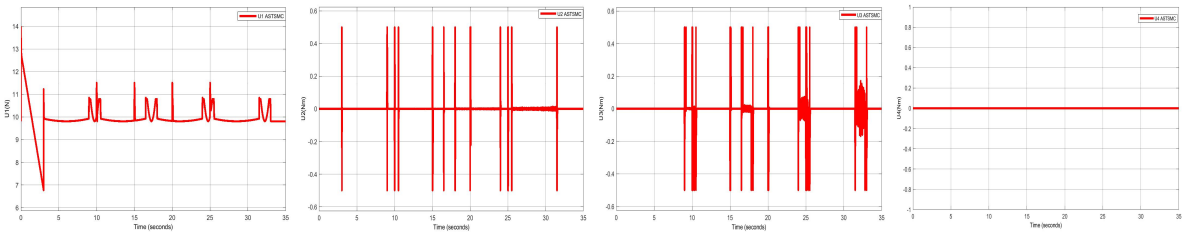
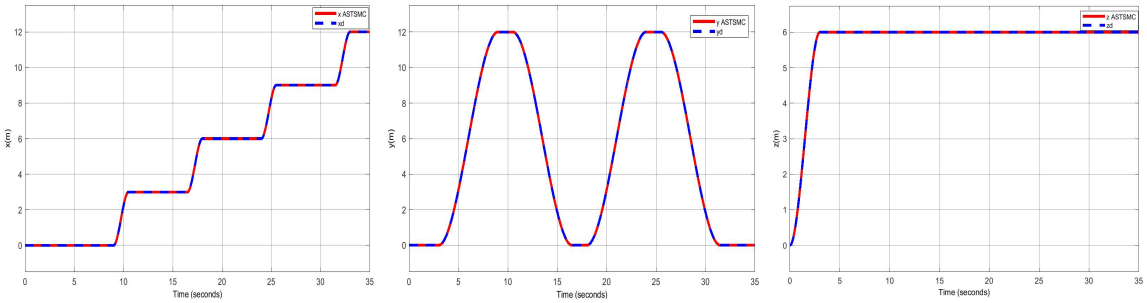


Figure 6.12: Control input with parametric uncertainty

Figure 6.12 shows the control commands required of the controller in order to maintain the design performance in spite of these parametric uncertainties.



(a) Trajectory tracking along x-axis (b) Trajectory tracking along y-axis (c) Trajectory tracking along z-axis

Figure 6.13: Square wave trajectory tracking performance with uncertainty

The simulation results can give important insights into how effectively the ASTSMC controller works and it indicates the translational response of the ASTSMC when the system is faced with parametric uncertainties.

6.8 Comparison of SMC and ASTSMC

6.8.1 Based on Chattering Reduction

For comparison based on chattering reduction helical input was considered.

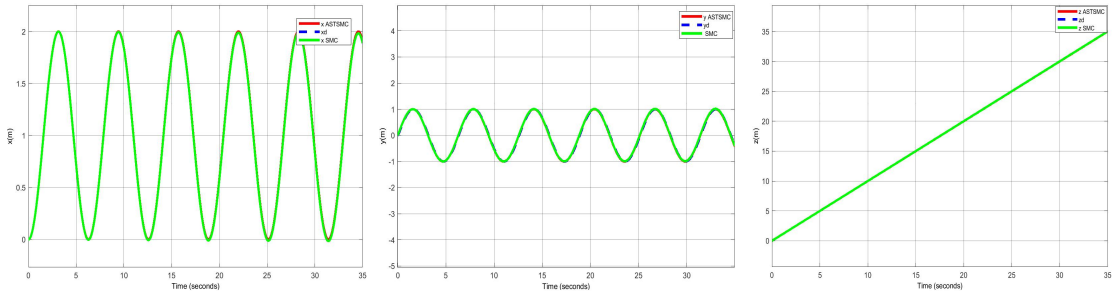


Figure 6.14: Comparison of SMC and ASTSMC

The SMC performance was found to be good since it was able to track the desired trajectory. But there was a noticeable chattering in the control signal, which can have an effect on the quadcopter's overall performance and stability. The advantages of using the STSMC to minimize chattering can be easily shown in the Figure 6.16 simulation result. The ASTSMC provides smoother control signals and improved tracking performance by effectively reducing chattering. Compared to the conventional SMC, this chattering reduction leads to improved stability and higher overall system performance. Chattering reduction for U_2 and U_3 are shown in Figure 6.16.

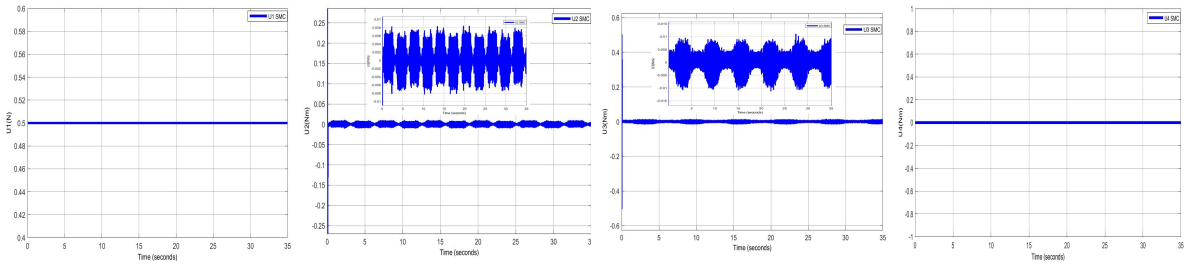


Figure 6.15: SMC control input for helical trajectory

After sliding mode controller Incorporated the chattering reduced.

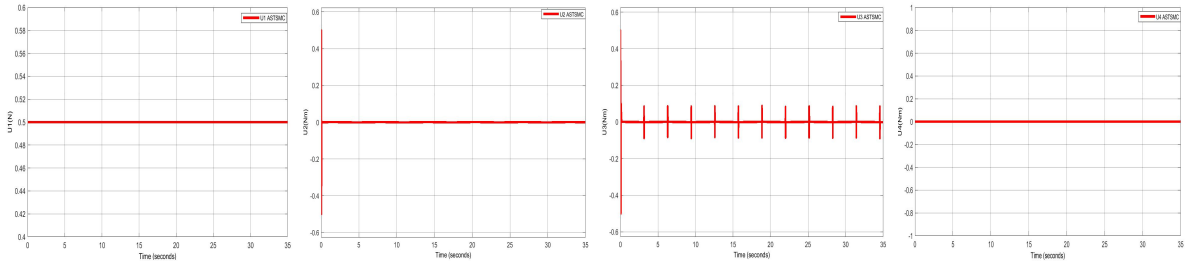


Figure 6.16: STSMC control input for helical trajectory

6.8.2 Comparison Based on Chattering Reduction and Controller Effort Reduction:

Comparison of SMC and ASTSMC done based on square wave input.

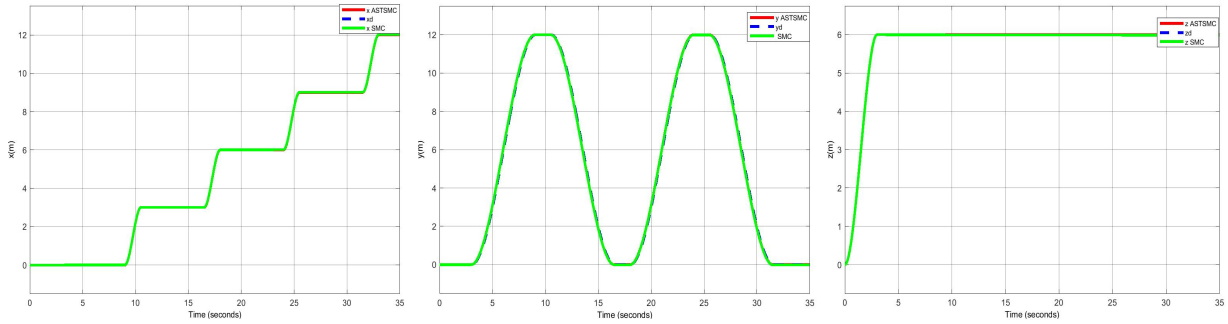


Figure 6.17: Comparison of SMC and ASTSMC

The SMC's control performance was found to be satisfactory since it was able to track the desired trajectory. But there was a noticeable chattering in U_2 and U_3 control signal.

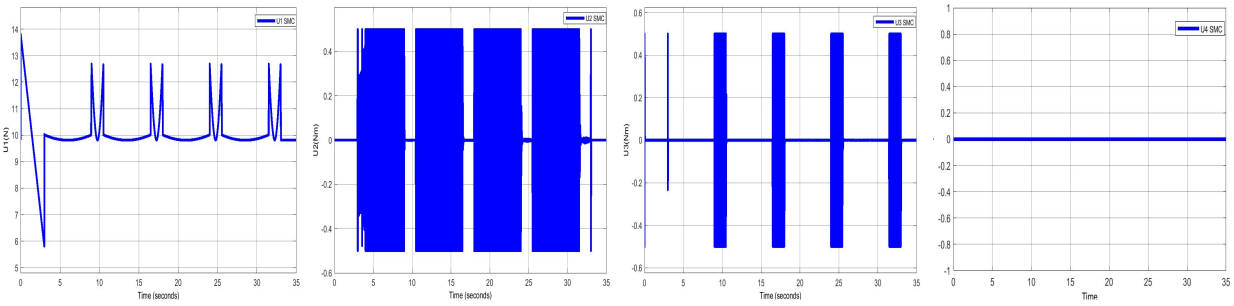


Figure 6.18: SMC control effort

The ASTSMC control effort for square wave input.

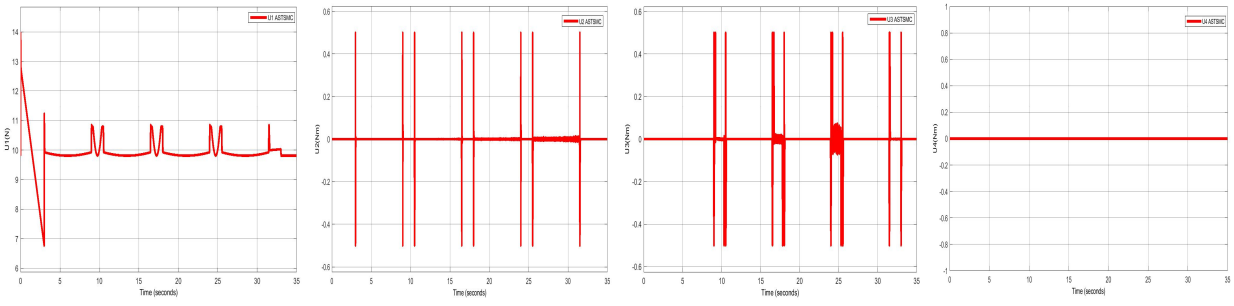


Figure 6.19: ASTSMC control effort

The advantages of using the STSMC to minimize chattering can be easily shown in the above simulation result Figure 6.19. The STSMC provides smoother control signals and

improved tracking performance by effectively reducing chattering. Chattering in sliding mode controller can increase controller effort. This is because the controller needs to continuously adjust and switch control strategies to counteract the chattering phenomenon, resulting in higher energy consumption and increased effort from the controller. In adaptive control the controller dynamically adjusts its control gains based on the system's response and variations. This allows the controller to achieve the desired control performance with less effort, as the gains are optimized for the specific system conditions and combining super twisting and adaptive controller with sliding mode controller as shown in Figure 6.19 the chattering can be minimized and the controller effort can be optimized.

6.8.3 Comparison based on tracking error

Comparstion of SMC with ASTSMC done based on helical input and the figure 6.20 shows trajectory tracking performance of the quadrotor.

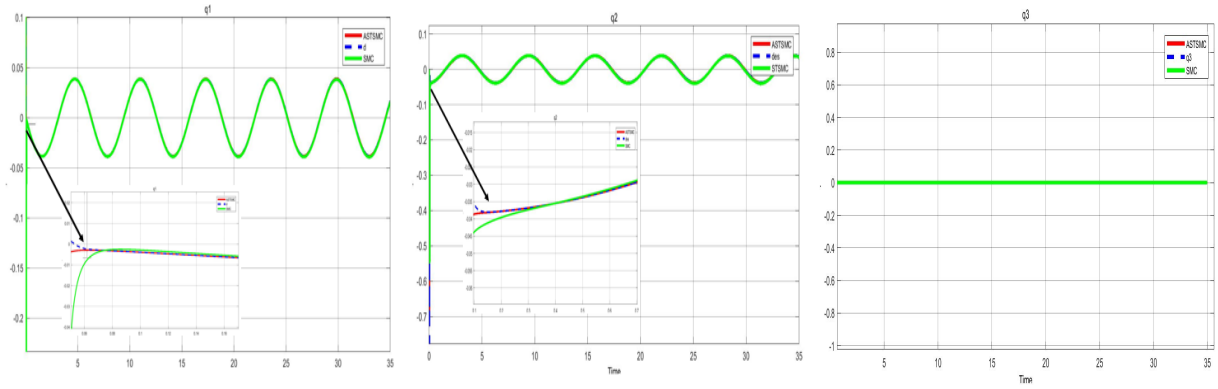


Figure 6.20: Comparison of SMC and ASTSMC based on tracking error

The above simulation results of tracking error also highlight the effectivness of ASTSMC over (SMC) in terms of tracking accuracy. The SMC exhibits tracking errors when compared to the ASTSMC. This can be attributed to the absence of adaptive control features in the SMC.

6.9 Image Recognition

6.9.1 Data preparation

The image recognition task is done using 2523 images with each in one of the two different classes. 'Brown wheat rust' and 'Healthy wheat' are the labels for the class classification.

Splitting into train, validation, and test was then completed.

- 2000 for training
- 523 for testing and validation

6.9.2 Training Phase

The training is implemented using the python programming language in the Google colab environment. The resenet152v2 architecture is adopted. The overall implementation is done and it is discussed more on Appendix K.

6.10 Training and Test Performance Analysis

Figure 6.21: Accuracy plot of wheat disease detection

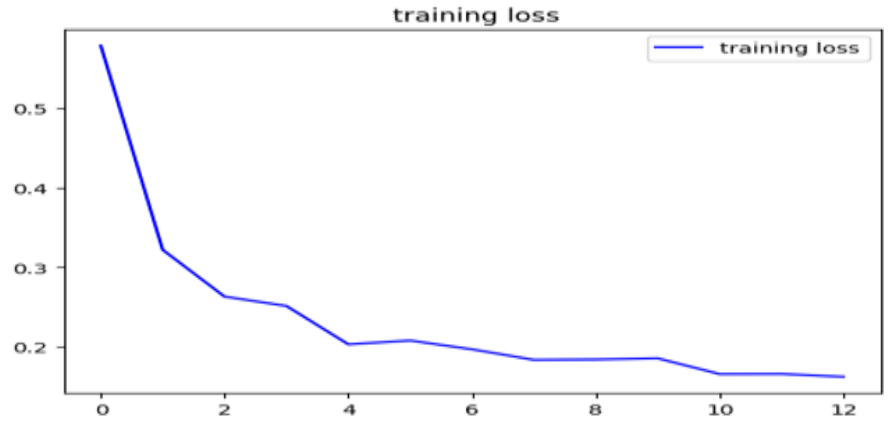


Figure 6.22: Loss plot of wheat disease detection

In the simulation results of wheat disease detection the accuracy of the image recognition model was evaluated. The accuracy of the model represents the ability of the model to

correctly classify and detect the presence of wheat diseases in the images. The results showed that the model achieved an training accuracy of 93.46% correctly identifying the wheat disease from the given images and the training loss is 0.1625 and 92.87% of test accuracy for epochs 13. This indicates that the model was able to classify the images with a relatively high level of accuracy. The Accuracy and the loss plot for epochs:13.

- Training Accuracy: 93.46%
- Test Accuracy: 92.87 %

Chapter 7

Conclusion and Recommendation

In this thesis, the Newton-quaternion formalization is used to derive the quadrotor (quadcopter) nonlinear dynamic model. The nonlinear quadrotor model has translational and rotational dynamics. The model includes the gyroscopic moment effect caused by the quadrotor's body and rotor blade. ASTSMC has been developed for the position and attitude subsystems, respectively. Different flight maneuvers such infinity, square wave, and helical trajectories are used to test the designed controller's overall performance in tracking trajectory. The simulation result for the Adaptive super twisting sliding mode controller shows good trajectory tracking where as the sliding mode controller gives a slow in the system and leads to more deviations from the given trajectory. In comparison to SMC, ASTSMC performs better in terms of eliminating tracking errors while minimizing the amount of control effort needed. Additionally, the simulation results show that the ASTSMC may provide reliable performance in the presence of severe disturbances. Even if we exert more force as a disturbance on the quadcopter's dynamics, ASTSMC resist the effects of the disturbance and provide responses that are close to the desired ones. Additionally, a 2523 picture dataset that has been split down into two classes healthy and brown wheat rust for image analysis purposes demonstrates that a retrained CNN model is accurate and has promising results in identifying brown wheat rust for precision agriculture applications. The whole control architecture is therefore practical for the real time implementation of the quadrotor platform, and the retrained CNN model can be utilized to identify brown rust.

Recommendation for Future Work

- Implementation of the quadcopter model on hardware.
- Arial image processing can further reformulate the image detection task used in this paper.

References

[1] J. Su, D. Yi, B. Su, Z. Mi, C. Liu, X. Hu, X. Xu, L. Guo, and W.-H. Chen, "Aerial visual perception in smart farming: Field study of wheat yellow rust monitoring," *IEEE transactions on industrial informatics*, vol. 17, no. 3, pp. 2242–2249, 2020.

[2] C. Zinke-Wehlmann and K. Charvát, "Introduction of smart agriculture," *Big Data in Bioeconomy: Results from the European DataBio Project*, pp. 187–190, 2021.

[3] A. Abbas, Z. Zhang, H. Zheng, M. M. Alami, A. F. Alrefaei, Q. Abbas, S. A. H. Naqvi, M. J. Rao, W. F. Mosa, Q. Abbas, *et al.*, "Drones in plant disease assessment, efficient monitoring, and detection: A way forward to smart agriculture," *Agronomy*, vol. 13, no. 6, p. 1524, 2023.

[4] U. R. Mogili and B. Deepak, "Review on application of drone systems in precision agriculture," *Procedia computer science*, vol. 133, pp. 502–509, 2019.

[5] R. Sharma, "Review on application of drone systems in precision agriculture," *Journal of Advanced Research in Electronics Engineering and Technology*, vol. 7, no. 2, pp. 5–7, 2021.

[6] J. A. Pandian, G. Geetharamani, and B. Annette, "Data augmentation on plant leaf disease image dataset using image manipulation and deep learning techniques," in *2019 IEEE 9th international conference on advanced computing (IACC)*, pp. 199–204, IEEE, 2019.

[7] P. Jiang, Y. Chen, B. Liu, D. He, and C. Liang, "Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks," *IEEE Access*, vol. 7, pp. 59069–59080, 2019.

[8] N. Parveen, A. Ali, and A. Ali, "Iot based automatic vehicle accident alert system," in

2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA), pp. 330–333, IEEE, 2020.

[9] M. S. Raval, S. Chaudhary, and J. Adinarayana, "Computer vision and machine learning in agriculture," *data science in agriculture and natural resource management*, pp. 97–126, 2022.

[10] P. K. Singh, N. C. Gahtyari, C. Roy, K. K. Roy, X. He, B. Tembo, K. Xu, P. Juliana, K. Sonder, M. R. Kabir, *et al.*, "Wheat blast: a disease spreading by intercontinental jumps and its management strategies," *Frontiers in Plant Science*, vol. 12, p. 1467, 2021.

[11] J. M. González-Camacho, L. Ornella, P. Pérez-Rodríguez, D. Gianola, S. Dreisigacker, and J. Crossa, "Applications of machine learning methods to genomic selection in breeding wheat for rust resistance," *The plant genome*, vol. 11, no. 2, p. 170104, 2018.

[12] M. Kumar, A. Kumar, and V. S. Palaparthy, "Soil sensors-based prediction system for plant diseases using exploratory data analysis and machine learning," *IEEE Sensors Journal*, vol. 21, no. 16, pp. 17455–17468, 2020.

[13] M. Ruwaimana, B. Satyanarayana, V. Otero, A. M. Muslim, M. Syafiq A, S. Ibrahim, D. Raymaekers, N. Koedam, and F. Dahdouh-Guebas, "The advantages of using drones over space-borne imagery in the mapping of mangrove forests," *PloS one*, vol. 13, no. 7, p. e0200288, 2018.

[14] H. Abaunza, P. Castillo, and R. Lozano, "Quaternion modeling and control approaches," *Handbook of Unmanned Aerial Vehicles*, pp. 1–29, 2018.

[15] R. Kumar, M. Bhargavapuri, A. M. Deshpande, S. Sridhar, K. Cohen, and M. Kumar, "Quaternion feedback based autonomous control of a quadcopter uav with thrust vectoring rotors," in *2020 american control conference (acc)*, pp. 3828–3833, IEEE, 2020.

[16] M. E. Guerrero-Sanchez, H. Abaunza, P. Castillo, R. Lozano, and C. D. García-Beltrán, "Quadrotor energy-based control laws: a unit-quaternion approach," *Journal of Intelligent & Robotic Systems*, vol. 88, pp. 347–377, 2017.

[17] Y. Sun, N. Xian, and H. Duan, "Linear-quadratic regulator controller design for quadrotor based on pigeon-inspired optimization," *Aircraft Engineering and Aerospace Technology*, vol. 88, no. 6, pp. 761–770, 2016.

[18] J. Li and Q. Zhang, "A linear switching function approach to sliding mode control and observation of descriptor systems," *Automatica*, vol. 95, pp. 112–121, 2018.

[19] M. Herrera, W. Chamorro, A. P. Gómez, and O. Camacho, "Sliding mode control: An approach to control a quadrotor," in *2015 Asia-Pacific Conference on Computer Aided System Engineering*, pp. 314–319, IEEE, 2015.

[20] S. Sridhar, R. Kumar, M. Radmanesh, and M. Kumar, "Non-linear sliding mode control of a tilting-rotor quadcopter," in *Dynamic Systems and Control Conference*, vol. 58271, p. V001T09A007, American Society of Mechanical Engineers, 2017.

[21] J. Burton and A. S. Zinober, "Continuous approximation of variable structure control," *International journal of systems science*, vol. 17, no. 6, pp. 875–885, 1986.

[22] A. Levant, "Higher-order sliding modes, differentiation and output-feedback control," *International journal of Control*, vol. 76, no. 9-10, pp. 924–941, 2003.

[23] O. Kaplan and F. Bodur, "Super twisting algorithm based sliding mode controller for buck converter with constant power load," in *2021 9th International Conference on Smart Grid (icSmartGrid)*, pp. 137–142, IEEE, 2021.

[24] Y. Shtessel, M. Taleb, and F. Plestan, "A novel adaptive-gain supertwisting sliding mode controller: Methodology and application," *Automatica*, vol. 48, no. 5, pp. 759–769, 2012.

[25] N. T. Nguyen and N. T. Nguyen, *Model-reference adaptive control*. Springer, 2018.

[26] N. Fethalla, *Modelling, identification, and control of a quadrotor helicopter*. PhD thesis, École de technologie supérieure, 2019.

[27] S. Musa, "Techniques for quadcopter modeling and design: A review," *Journal of unmanned system Technology*, vol. 5, no. 3, pp. 66–75, 2018.

[28] S. Gupte, P. I. T. Mohandas, and J. M. Conrad, "A survey of quadrotor unmanned aerial vehicles," *2012 Proceedings of IEEE Southeastcon*, pp. 1–6, 2012.

[29] Q. Quan, *Introduction to multicopter design and control*. Springer, 2017.

[30] S. L. Altmann, "Hamilton, rodrigues, and the quaternion scandal," *Mathematics Magazine*, vol. 62, no. 5, pp. 291–308, 1989.

[31] B. J. Emran and H. Najjaran, "A review of quadrotor: An underactuated mechanical system," *Annual Reviews in Control*, vol. 46, pp. 165–180, 2018.

[32] A. Kelly and N. Seegmiller, "A vector algebra formulation of mobile robot velocity kinematics," in *Field and Service Robotics: Results of the 8th International Conference*, pp. 613–627, Springer, 2013.

[33] T. Bresciani, "Modelling, identification and control of a quadrotor helicopter," *MSc theses*, 2008.

[34] B. Zhang, K. Nie, X. Chen, and Y. Mao, "Development of sliding mode controller based on internal model controller for higher precision electro-optical tracking system," in *Actuators*, vol. 11, p. 16, MDPI, 2022.

[35] S. Bouabdallah, A. Noth, and R. Siegwart, "Pid vs lq control techniques applied to an indoor micro quadrotor," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3, pp. 2451–2456, IEEE, 2004.

[36] L. Besnard, Y. B. Shtessel, and B. Landrum, "Control of a quadrotor vehicle using sliding mode disturbance observer," in *2007 American Control Conference*, pp. 5230–5235, IEEE, 2007.

[37] M. Figueroa, K. E. Hammond-Kosack, and P. S. Solomon, "A review of wheat diseases—a field perspective," *Molecular plant pathology*, vol. 19, no. 6, pp. 1523–1536, 2018.

[38] P. Moghadam, D. Ward, E. Goan, S. Jayawardena, P. Sikka, and E. Hernandez, "Plant disease detection using hyperspectral imaging," in *2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–8, IEEE, 2017.

[39] Z. Feng and J. Fei, "Design and analysis of adaptive super-twisting sliding mode control for a microgyroscope," *PloS one*, vol. 13, no. 1, p. e0189457, 2018.

Appendix A

Quaternion Operation

A.1 Quaternion to Euler conversion matrix

since quaternion lack physical intuition it is possible to convert quaternion to Euler using the following equation:

$$Roll = tan^{-}1(\frac{2(q_0q_1 + q_2q_3))}{q_0^2 - q_1^2 - q_2^2 + q_3^2)}) \tag{A.1}$$

$$= atan2[2(q_0q_1 + q_2q_3), q_0^2 - q_1^2 - q_2^2 + q_3^2)] \tag{A.2}$$

$$Pitch = sin^{-}1(2(q_0q_2 + q_1q_3)) \tag{A.3}$$

$$= asin[2(q_0q_2 + q_1q_3)] \tag{A.4}$$

$$Yaw = tan^{-}1(\frac{2(q_0q_3 + q_1q_2))}{q_0^2 - q_1^2 - q_2^2 + q_3^2)}) \tag{A.5}$$

$$= atan2[2(q_0q_3 + q_1q_2), q_0^2 - q_1^2 - q_2^2 + q_3^2]] \tag{A.6}$$

A.2 Quaternion Rotation Matrix

From equation [3.8] we have :

$$v' = q \otimes \begin{bmatrix} 0 \\ V \end{bmatrix} \otimes q^{'} \tag{A.7}$$

$$R_1 = q \otimes \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \otimes q^{'} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 \\ 2(q_1q_2 + q_0q_3) \\ 2(q_1q_3 - q_0q_2) \end{bmatrix} \tag{A.8}$$

$$R_2 = q \otimes \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \otimes q^{'} = \begin{bmatrix} 2(q_1q_2 - q_0q_3) \\ q_0^2 - q_1^2 + q_2^2 - q_3^2 \\ 2(q_2q_3 - q_0q_1) \end{bmatrix} \tag{A.9}$$

$$R_3 = q \otimes \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \otimes q^{'} = \begin{bmatrix} 2(q_1q_3 + q_0q_2) \\ 2(q_2q_3 - q_0q_1) \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{A.10}$$

Only the vector part of the quaternion is extracted, resulting in the rotation matrix rotating a point on a fixed coordinate.

$$R_E^B = \begin{bmatrix} R_1 \\ R_2 \\ R_3 \end{bmatrix}^T \tag{A.11}$$

$$R_E^B = \begin{bmatrix} q_0^2 + q_1^2 + q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{A.12}$$

Appendix B

Design Time Plant Parameters

Plant Parameters	Symbol	Unit	Values
k_f	Trust factor	Ns^2	$54.2 * 10^-6$
d	Drag factor	Nms^2	$1.1 * 10^-6$
g	Gravitational Acceleration	m/s^2	9.81
l	Arm length of the quad-rotor	m	0.24
m	Mass of quad-rotor	Kg	1
I_{xx}	Trust factor	Nms^2	$8.1 * 10^-3$
I_{yy}	Trust factor	Nms^2	$8.1 * 10^-3$
I_{zz}	Trust factor	Nms^2	$14.23 * 10^-3$

Table B.1: Plant Parameter [33]

Appendix C

Trust Force

From fixed frame trust force we have $F_1 + F_2 + F_3 + F_4 = U_1$ to move from body fixed frame to inertial frame rotational matrix is used and the virtual control input in the x, y, z is U_x, U_y, U_z.

$$\begin{bmatrix} U_x \\ U_y \\ U_z \end{bmatrix} = \begin{bmatrix} q_0^2 + q_1^2 + q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{C.1}$$

$$\begin{bmatrix} U_x \\ U_y \\ U_z \end{bmatrix} = \begin{bmatrix} 2(q_1q_3 - q_0q_2)U_1 \\ 2(q_2q_3 - q_0q_1)U_1 \\ (q_0^2 - q_1^2 - q_2^2 + q_3^2)U_1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{C.2}$$

From the above equation we can see that $U_x = \ddot{x}$, $U_y = \ddot{y}$ and $U_z = \ddot{z}$

Finding the magnitude of the vector

$$U_x^2 + U_y^2 + (U_z - m_g)^2 = U_1[4(q_1q_3 - q_0q_2)^2 + 4(q_2q_3 - q_0q_1)^2 + (q_0^2 - q_1^2 - q_2^2 + q_3^2)^2] = U_1^2 \tag{C.3}$$

$$U_1 = \sqrt{U_x^2 + U_y^2 + (U_z - mg)^2} \tag{C.4}$$

Appendix D

Desired Angular Velocity Calculation

To calculate the desired angular velocity first rate of change of quaternion should be calculated :-

$$q = e^{\frac{\theta U}{2}} = cos\frac{\theta}{2} + Usin\frac{\theta}{2} \tag{D.1}$$

$$\dot{q} = e^{\frac{\theta U}{2}} \frac{U}{2} \dot{\theta} \tag{D.2}$$

Where $\dot{\theta}$ represents the angular velocity(ω). The final rate of change of quatenion become:-

$$\dot{q_d} = \frac{1}{2} q \otimes \begin{bmatrix} 0 \\ w_d \end{bmatrix} \tag{D.3}$$

$$\dot{q_d} = \frac{1}{2} \begin{bmatrix} q_{0d} & -q_{1d} & -q_{2d} & -q_{3d} \\ q_{1d} & q_{0d} & q_{3d} & q_{2d} \\ q_{2d} & q_{3d} & q_{0d} & -q_{1d} \\ q_{3d} & -q_{2d} & q_{1d} & q_{0d} \end{bmatrix} \begin{bmatrix} 0 \\ p_d \\ q_d \\ r_d \end{bmatrix} \tag{D.4}$$

By rearranging and using the inverse property of quaternion which is $q^{-1} = q^T$ the above equation we can find the desired velocity after rearranging the equation:-

$$w_d = 2q^T \dot{q_d} \tag{D.5}$$

$$\dot{w_d} = 2q^T \ddot{q_d} \tag{D.6}$$

$$w_d = 2 \begin{bmatrix} q_{0d} & q_{1d} & q_{2d} & -q_{3d} \\ -q_{1d} & q_{0d} & q_{3d} & -q_{2d} \\ -q_{2d} & q_{3d} & q_{0d} & q_{1d} \\ -q_{3d} & q_{2d} & -q_{1d} & q_{0d} \end{bmatrix} \begin{bmatrix} \dot{q_{0d}} \\ \dot{q_{1d}} \\ \dot{q_{2d}} \\ \dot{q_{3d}} \end{bmatrix} \tag{D.7}$$

$$\dot{w_d} = 2 \begin{bmatrix} q_{0d} & q_{1d} & q_{2d} & -q_{3d} \\ -q_{1d} & q_{0d} & q_{3d} & -q_{2d} \\ -q_{2d} & q_{3d} & q_{0d} & q_{1d} \\ -q_{3d} & q_{2d} & -q_{1d} & q_{0d} \end{bmatrix} \begin{bmatrix} \ddot{q_{0d}} \\ \ddot{q_{1d}} \\ \ddot{q_{2d}} \\ \ddot{q_{3d}} \end{bmatrix} \tag{D.8}$$

Appendix E

State Space Representation

State space representation of a quadcopter typically includes the following 12 state variables: Position in x,y and z axis, velocity $\dot{x}$, $\dot{y}$, $\dot{z}$, roll angle, pitch angle, yaw angle, angular velocity in x-axis, y-axis, z-axis, (p,q,r).

$$\begin{cases} \dot{x_1} & = x_2 \\ \dot{x_2} & = \frac{u_x}{m} \\ \dot{x_3} & = x_4 \\ \dot{x_4} & = \frac{u_y}{m} \\ \dot{x_5} & = x_6 \\ \dot{x_6} & = \frac{u_z}{m} \\ \dot{x_7} & = x_8 \\ \dot{x_8} & = (a_1 x_{10} x12) + (b_1 u_2) \\ \dot{x_9} & = x_{10} \\ \dot{x_{10}} & = (a_2 x_8 x_{12} + (b_2 u_3) \\ \dot{x_{11}} & = x_{12} \\ \dot{x_{12}} & = (a_3 x_8 x_{10}) + (b_3 u_4) \end{cases} \tag{E.1}$$

where

$$\begin{cases} a_1 & = \frac{I_{zz}-Iyy}{I_{xx}} \\ b_1 & = \frac{1}{I_{xx}} \\ a_2 & = \frac{I_{zz}-Ixx}{I_{yy}} \\ b_2 & = \frac{1}{I_{yy}} \\ a_3 & = \frac{I_{xx}-Iyy}{I_{zz}} \\ b_3 & = \frac{1}{I_{zz}} \end{cases} \tag{E.2}$$

Here we can redefine the state variable as follows

$$[x, \dot{x}, y, \dot{y}, z, \dot{z}, q_1, \dot{q_1}, q_2, \dot{q_2}, q_3, \dot{q_3}] = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}]^T$$

Where x, y, z are position in x,y and z axis respectively $\dot{x}, \dot{y}, \dot{z}$ are speed in x,y,z axis respectively then the final quadcopter state space equation.

Appendix F

Stability Analysis

The Lyapunov function is a positive scalar function of the system state variables $V(x) > 0$ and the control law must satisfy $\dot{V(x)} < 0$. The Lyapunov function is defined as $V_x = \frac{1}{2}\sigma^T\sigma$. To get the minimum value of the super twisting controller gains let as proof:

Stability analysis for ASTSMC U_x

$$\dot{V}_x = \sigma_x\dot{\sigma_x} < 0$$
$$\dot{V}_x = \sigma_x(\lambda_x e_x + (\ddot{x_d} - \ddot{x}))$$

From virtual control law substitute the value of $\ddot{x}$.

$$U_x = \ddot{x} = \lambda_x e_x - \ddot{x_d} - \alpha_x\sqrt{\mid \sigma_x \mid}sign(\sigma_x) - \int \beta_x sign(\sigma_x)dt$$
$$\dot{V}_x = \sigma_x(\lambda_x e_x + \ddot{x_d} - \lambda_x e_x - \ddot{x_d} - \alpha_x\sqrt{\mid \sigma_x \mid}sign(\sigma_x) - \int \beta_x sign(\sigma_x)dt)$$
$$\dot{V}_x = \sigma_x(-\alpha_x\sqrt{\mid \sigma_x \mid}sign(\sigma_x) - \int \beta_x sign(\sigma_x)dt) <= 0 \quad \text{(F.1)}$$

To make the Lyapunov function valid, the gains α_x, β_x must be greater than zero.

Stability analysis for ASTSMC U_y

$$\dot{V}_y = \sigma_y \dot{\sigma_y} < 0$$
$$\dot{V}_y = \sigma_y(\lambda_y \dot{e_y} + (\ddot{y_d} - \ddot{y}))$$

From virtual control law substitute the value of $\ddot{y}$.

$$U_y = \ddot{y} = \lambda_y \dot{e_y} - \ddot{y_d} - \alpha_y \sqrt{\mid \sigma_y \mid} sign(\sigma_y) - \int \beta_y sign(\sigma_y) dt$$
$$\dot{V}_y = \sigma_y(\lambda_y e_y + \ddot{y_d} - \lambda_y e_y - \ddot{y_d} - \alpha_y \sqrt{\mid \sigma_y \mid} sign(\sigma_y) - \int \beta_y sign(\sigma_y) dt)$$
$$\dot{V}_y = \sigma_y(-\alpha_y \sqrt{\mid \sigma_y \mid} sign(\sigma_y) - \int \beta_y sign(\sigma_y) dt) <= 0 \tag{F.2}$$

To make the Lyapunov function valid, the gains α_y, β_y must be greater than zero.

Stability analysis for ASTSMC U_z

$$\dot{V}_z = \sigma_z \dot{\sigma_z} < 0$$
$$\dot{V}_z = \sigma_z(\lambda_z \dot{e_z} + (\ddot{z_d} - \ddot{z}))$$

From virtual control law substitute the value of $\ddot{z}$.

$$U_z = \ddot{z} = \lambda_z \dot{e_z} - \ddot{z_d} - \alpha_z \sqrt{\mid \sigma_z \mid} sign(\sigma_z) - \int \beta_z sign(\sigma_z) dt$$
$$\dot{V}_z = \sigma_z(\lambda_z \dot{e_z} + \ddot{z_d} - \lambda_z \dot{e_z} - \ddot{z_d} - \alpha_z \sqrt{\mid \sigma_z \mid} sign(\sigma_z) - \int \beta_z sign(\sigma_z) dt)$$
$$\dot{V}_z = \sigma_z(-\alpha_z \sqrt{\mid \sigma_z \mid} sign(\sigma_z) - \int \beta_z sign(\sigma_z) dt) <= 0 \tag{F.3}$$

To make the Lyapunov function valid, the gains α_z, β_z must be greater than zero.

In the same way as position control, controller gain ranges in the Super Twisting sliding mode are determined using the Lyapunov stability theorem:

Stability analysis for super twisting gain for U_2

$$
\begin{aligned}
V_{q1} &= \frac{1}{2}\sigma_{q1}^{T}\sigma_{q1} \\
\dot{V_{q1}} &= \sigma_{q1}\dot{\sigma_{q1}} <= 0 \\
\dot{V_{q1}} &= \sigma_{q1}(\lambda_{q1}\dot{e_{q1}} + (\ddot{q_{1d}} - \ddot{q_1})) \\
\dot{V_{q1}} &= \sigma_{q1}(\lambda_{q1}\dot{e_{q1}} + \ddot{q_{1d}} - (\frac{I_{yy} - I_{zz}}{I_{xx}}\dot{q}_1\dot{q}_3 - \frac{J_t}{I_{xx}}\dot{q}_1\omega)) \\
U_2 &= I_{xx}(\lambda_{q1}\dot{e_{q1}} + \ddot{q_{1d}} - (\frac{I_{yy} - I_{zz}}{I_{xx}}\dot{q}_1\dot{q}_3 - \frac{J_t}{I_{xx}}\dot{q}_1\omega + U_{st})) \\
\dot{V_{q1}} &= \sigma_{q1}(-\alpha_{q1}\sqrt{\mid \sigma_{q1} \mid}sign(\sigma_{q1}) - \int \beta_{q1}sign(\sigma_{q1})dt) <= 0
\end{aligned}
$$

To make the Lyapunov function valid, the gains α_{q1}, β_{q2} must be greater than zero.

Stability analysis for super twisting gain for U_3

$$
\begin{aligned}
V_{q2} &= \frac{1}{2}\sigma_{q2}^{T}\sigma_{q2} \\
\dot{V_{q2}} &= \sigma_{q2}\dot{\sigma_{q2}} <= 0 \\
\dot{V_{q2}} &= \sigma_{q2}(\lambda_{q2}\dot{e_{q2}} + (\ddot{q_{2d}} - \ddot{q_2})) \\
\dot{V_{q2}} &= \sigma_{q2}(\lambda_{q2}\dot{e_{q2}} + \ddot{q_{2d}} - (\frac{I_{zz} - I_{xx}}{I_{yy}}\dot{q}_2\dot{q}_3 - \frac{J_{tp}}{I_{yy}}\dot{q}_2\omega - \frac{U_3}{I_{yy}})) \\
U_3 &= I_{yy}(\lambda_{q2}\dot{e_{q2}} + \ddot{q_{2d}} - (\frac{I_{zz} - I_{xx}}{I_{yy}}\dot{q}_2\dot{q}_3 - \frac{J_{tp}}{I_{yy}}\dot{q}_2\omega + U_{st})) \\
\dot{V_{q2}} &= \sigma_{q2}(-\alpha_{q2}\sqrt{\mid \sigma_{q2} \mid}sign(\sigma_{q2}) - \int \beta_{q2}sign(\sigma_{q2})dt) <= 0
\end{aligned}
$$

To make the Lyapunov function valid, the gains α_{q2}, β_{q2} must be greater than zero.

Stability analysis for super twisting gain for U_4

$$
\begin{aligned}
V_{q3} &= \frac{1}{2}\sigma_{q3}^{T}\sigma_{q3} \\
\dot{V_{q3}} &= \sigma_{q3}\dot{\sigma_{q3}} <= 0 \\
\dot{V_{q3}} &= \sigma_{q3}(\lambda_{q3}\dot{e_{q3}} + (\ddot{q_{3d}} - \ddot{q_3})) \\
\dot{V_{q3}} &= \sigma_{q3}(\lambda_{q3}\dot{e_{q3}} + \ddot{q_{3d}} - (\frac{I_{xx} - I_{yy}}{I_{zz}}\dot{q_1}\dot{q_3} - \frac{U_4}{I_{zz}})) \\
U_4 &= I_{zz}(\lambda_{q3}\dot{e_{q3}} + \ddot{q_{3d}} - \frac{I_{xx} - I_{yy}}{I_{zz}}\dot{q_1}\dot{q_2} + U_{st} \\
\dot{V_{q3}} &= \sigma_{q3}(-\alpha_{q3}\sqrt{\mid \sigma_{q3} \mid}sign(\sigma_{q3}) - \int \beta_{q3}sign(\sigma_{q3})dt) <= 0
\end{aligned}
$$

To make the Lyapunov function valid, the gains α_{q3}, β_{q3} must be greater than zero [39].

Appendix G

Horizontal Filed of View

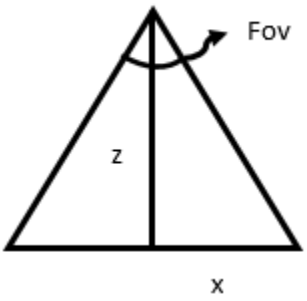


Figure G.1: Horizontal field of view

$$
\begin{aligned}
tan(HFOV) &= \frac{x}{z} \\
HFOV &= 2tan^{-}1(\frac{x}{z}) \\
HFOV &= 2tan^{-}1(\frac{12}{6}) \\
HFOV &= 26.8^{\circ}
\end{aligned}
$$

The above calculation shows the selected horizontal distance and HFOV of selected camera match.

Appendix H

Trajectory Generation

- Let see for z- trajectory

  Cost function(postion constraint): $z(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3$

  velocity constraint: $\dot{z(t)} = c_1 + 2c_2 t^2 + 3c_3 t^3$ velocity constraint at via point used to change path.

  Boundary condition: $Z(t_0)$=given, $Z(t_f)$=given, $z(t_0)$=0, and $z(t_f)$=0

- Given $t_0 = 0$ sec, $t_f = 3$sec, $z(t_0) = 0$m, and $z(t_f) = 6$m

$$z(t_0) = c_0 + c_1 t_0 + c_2 t_0^2 + c_3 t_0^3 \tag{H.1}$$

$$\dot{z(t_0)} = c_1 + 2c_2 t_0 + 3c_3 t_0^2 \tag{H.2}$$

$$z(t_f) = c_0 + c_1 t_f + c_2 t_f^2 + c_3 t_f^3 \tag{H.3}$$

$$\dot{z(t_f)} = c_1 + 2c_2 t_f + 3c_3 t_f^2 \tag{H.4}$$

By rearranging the the constant values can be calculated as follow:-

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 1 & t_o & t_o^2 & t_o^3 \\ 0 & 1 & 2t_0 & 3t_0^2 \\ 1 & t_f & 2t_f & t_f^3 \\ 0 & 1 & 2t_f & 3t_f^2 \end{bmatrix}^{-1} \begin{bmatrix} z(t_o) \\ \dot{z(t_o)} \\ z(t_f) \\ \dot{z(t_f)} \end{bmatrix} \tag{H.5}$$

Using the same method x and y trajectory can be generated.

H.1 Square Wave Trajectory Generation

```
function [x,y,z] = fcn(t)
if t<3
x = 0;
y  = 0;
z =  2 * t^2 - (4/9) * t^3;
elseif (3<=t)&&(t<=9)
x = 0;
y = 12 - 9 * t + 2 * t^2 - (1/9) * t^3;
z = 6;
elseif (9<=t)&&(t<=10.5)
x = 1620 -  504 * t + 52 * t^2 - (16/9) * t^3;
y = 12;
z = 6;
elseif (10.5<t)&&(t<=16.5)
x = 3;
y = -(1815/8) + (231/4) * t - (9/2) * t^2 + (1/9) * t^3;
z = 6;
elseif (16.5<t)&&(t<=18)
x = (9078) - (1584) * t + (92) * t^2 - (16/9) * t^3;
y = 0;
z = 6;
elseif (18<t)&&(t<=24)
x = 6;
y = 972 -144 * t + 7 * t^2 - (1/9) * t^3;
z = 6;
elseif (24<t)&&(t<=25.5)
x = (26886) - (3264) * t + (132) * t^2 - (16/9) * t^3;
y = 12;
z = 6;
elseif (25.5<t)&&(t<=31.5)
x = 9;
y = (-19845/8) + (1071/4) * t - (19/2) * t^2 + (1/9) * t^3;
z = 6;
```

```
elseif (31.5<t)&&(t<=33)
x = (59544) - (5544) * t + (172) * t^2 - (16/9) * t^3;
y = 0;
z = 6;
else
x = 12;
y = 0;
z = 6;
end
```

Appendix I

Hyper Spectral Camera Specification

Specifications	Values
Spectral range	400-1000 nm
Spectral Resolution	10 nm
Ground Sampling Distance (GSD)	1-2 cm/pixel
Horizontal Field of View (HFOV)	26.8°
Capture speed	one capture per second
Weight	100gram

Table I.1: Camera specification

Appendix J

Python Code for Wheat Rust Detection

```
from keras.layers import Input, Lambda, Dense, Flatten
# keras is high-level neural networks  compatible with TensorFlow
from keras.models import Model
from tensorflow.keras.applications.resnet_v2
import preprocess_input,decode_predictions
from keras.preprocessing import image
from keras.preprocessing.image import ImageDataGenerator
from glob import glob
from keras.layers import Activation
from keras.metrics import categorical_crossentropy
from matplotlib import pyplot as plt
#A plotting library that provides a  high-quality  visualizations.
from keras.layers.pooling
from tensorflow.keras.layers import Dropout
import Adam
import cv2
import os
# All the above are necessary libraries
```

```
import tensorflow as tf
ds_train =  tf.keras.preprocessing.image_dataset_from_directory
("/content/drive/MyDrive/dataset/data/train")
ds_validation = tf.keras.preprocessing.image_dataset_from_directory
("/content/drive/MyDrive/dataset/data/val")
#Specifying the directory where the  dataset are stored:
size = (224,224)
ds_train= ds_train.map(lambdaimage,label:(tf.image.resize(image, size),label))
ds_validation = ds_validation.map(lambda image,label:
(tf.image.resize(image, size),label))
image_size = [224,224]
model = ResNet152V2(input_shape=image_size+ [3], weights='imagenet',
include_top=False)
#Loading the weights of pretrained model
for layer in model.layers:
layer.trainable = False
#Freezing the earlier layers of the loaded pre-trained model
folders = glob('/content/drive/MyDrive/dataset/data/*')
x = Flatten()(model.output)
prediction = Dense(len(folders),activation='softmax')(x)
models = Model(inputs=model.input,outputs=prediction)
model.summary()
adam= optimizers.Adam(learning_rate=.0001)
models.compile(
loss='categorical_crossentropy',
optimizer = adam,
metrics=['accuracy']
)
train_datagen=ImageDataGenerator(
rescale = 1./255,
zoom_range =40,
shear_range=0.2,
```

```
horizontal_flip= True,)
#Applying data augumentation technique to the training data set
test_datagen = ImageDataGenerator(
rescale = 1./255,
zoom_range =40,
shear_range=0.2,
horizontal_flip= True,)
#Applying data augumentation technique to the testing adata set
model_history = models.fit(
training_set,
epochs=13,
steps_per_epoch=len(training_set),
)
```