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

Trajectory Tracking Control of Quadcopter using Fuzzy Super Twisting SMC with PID Surface for Wheat Yield Estimation

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
Elisabeth Andarge Gedefaw
Advisor
Lebsework Negash Lemma (PhD)

November 24, 2023
Addis Ababa, Ethiopia

Addis Ababa University
Addis Ababa Institute of Technology
School of Electrical and Computer Engineering

Trajectory Tracking Control of Quadcopter using Fuzzy Super Twisting SMC with PID Surface for Wheat Yield Estimation

By: Elisabeth Andarge Gedefaw

APPROVED BY BOARD OF EXAMINERS

Name	Signature	Date
Bisrat Derebssa (PhD) (School Dean)		
Lebsework Negash (PhD) (Advisor)		
Mihretab Negash (MSc) (Internal Examiner)		
Chala Merga (PhD) (External Examiner)		

Declaration

I, the undersigned, affirm that this MSc thesis is my own original work, and it has not been submitted for the completion of any degree at this university or any other institution. I further attest that I have diligently acknowledged and credited all sources and materials utilized in this thesis in accordance with academic and ethical standards.

Name	**Signature**
Elisabeth Andarge	

Place:

Addis Ababa Institute of Technology,

AAIT Addis Ababa University, AAU

Addis Ababa,

Ethiopia.

Submitted in: November 24, 2023

This thesis has been submitted for examination with my approval as a university advisor.

Advisor	**Signature**
Lebsework Negash (PhD)	

Acknowledgment

I begin by expressing my profound gratitude to GOD for giving me strength, guidance, and blessings throughout this thesis work. Additionally, I extend my heartfelt thanks to Dr. Lebsework Negash for his expert guidance and invaluable mentorship, which have been instrumental in shaping this research. My gratitude also extends to my university's faculty for fostering an enriching academic environment, my colleagues and friends for their collaborative spirit, and my family for their enduring support. Furthermore, I acknowledge the contributions of fellow researchers whose work has laid the foundation for this study. This research represents the collective efforts of many, and I am deeply appreciative of their contributions to its success.

Abstract

Accurate yield estimation during the heading stage of wheat production is pivotal for efficient harvest planning and ensuring food security. However, traditional yield estimation technique is labor-intensive and potentially harmful to the crop. Alternatively, the use of satellite imagery for estimation is hindered by its poor resolution. Another approach involves employing low-altitude quadcopters to capture high-resolution images. Yet, controlling flight of these quadcopters presents challenges due to their nonlinear and underactuated characteristics.

This thesis aims to address these challenges by designing a flight controller capable of controlling the quadcopter to track the desired trajectory using Fuzzy super twisting sliding mode controller with PID surface to capture images and estimate wheat yield. The process begins with modeling the quadcopter using the Newton Euler method, followed by designing a flight controller. This controller is divided into inner and outer loops capable of automatically adjusting its parameters, and the system is then simulated in MATLAB/Simulink. Wheat head images acquired from the quadcopter are trained using transfer learning in YOLOv8. Parameters such as kernel weight, the number of kernels per head, and sampled area are extracted from the field.

The controller's performance is rigorously assessed through various trajectory and disturbance scenarios, comparing it with sliding mode controller both with and without parameter variation. The results demonstrate the controller's efficiency in guiding the quadcopter along predefined trajectories, robustly rejecting disturbances, and effectively handling parameter variations. In the realm of image processing, the system exhibited notable advancements in both training and validation accuracy.

In summary, the proposed controller enhances robustness, the capacity to handle parameter variations, disturbance rejection, chattering minimization, and controller effort reduction. Utilizing the trained weight parameters, the system can accurately detect and count wheat heads, ultimately providing an estimation of wheat yield.

Keywords: Quadcopter, Fuzzy Super twisting sliding mode controller, PID surface, and Wheat yield estimation.

Table of Contents

List of Figures

List of Tables

List of Acronyms

ASMC	Adaptive Sliding Mode Controller
COG	Center of Gravity
CW	Clockwise
CCW	Counterclockwise
CNN	Convolutional Neural Network
DOF	Degrees of Freedom
FOV	Field of View
GPS	Global Positioning System
IoU	Intersection over Union
ITAE	Integral Time Absolute Error
LQ	Linear Quadratic
LQR	Linear Quadratic Regulator
mAP	Mean Average Precision
NDVI	Normalized Difference Vegetation Index
PD	Proportional Derivative
PID	Proportional Integral Derivative
RS	Remote Sensing
SAFY	Simple Algorithm for Yield
SMC	Sliding Mode Control
STA	Super Twisting Algorithm
STSMC	Super Twisting Sliding Mode Controller
UAV	Unmanned Aerial Vehicles
VSC	Variable Structure Control
VI	Vegetation Indices
YOLO	You Only Look Once
YOLOv8	You Only Look Once Version 8

Chapter One

Introduction

1.1 Background

Wheat is one of the most widely cultivated cereal grains globally, serving as a fundamental crop with a crucial role. It acts as the primary source of nutrition, calories, and protein for millions of people around the world [1]. Especially in the period preceding harvest, precise yield prediction holds a critical role in ensuring a nation's food security [2]. Wheat yield assessments can be conducted on multiple levels, ranging from individual fields to regional and national scales. On a field level, accurate yield forecasts enable farmers to adjust their harvest planning [3]. Additionally, these predictions serve various organizational purposes, including marketing and disaster response strategies [4]. Nevertheless, the laborious data collection frequently pose difficulties in applying traditional yield prediction methods on a regional scale [5].

Traditionally, yield estimation has relied heavily on field research, a laborious and less effective process that has harmful effects on crops. The arrival of remote sensing techniques has revolutionized crop yield estimation by offering a remote sensing platform [6,7]. Currently, satellite-based remote sensing technology plays a significant role in large-scale crop prediction [8]. Nevertheless, it has its limitations, including lengthy revisit intervals and imprecise resolution [9,10]. In contrast, low-altitude unmanned aerial vehicles (UAVs) offer advantages such as higher resolution, lower operational costs, flexibility, and repeatability [11,12]. UAVs can quickly acquire centimeter-level remote sensing imagery over wide farmlands, assisting agricultural decision-makers [13]. In this thesis, the study zone is a wheat farm located in Yebokla, Ethiopia, as shown in Figure 1.1. The coordinates of the field are (10°25'23.7084" N, 37°53'17.4702" E).

Unmanned aerial vehicles have evolved into highly advanced robotic systems employed for diverse purposes, including agricultural spraying, power line fault detection, and traffic monitoring [14]. These UAVs fall into two main categories: fixed-wing and multi-copter (rotary-wing) aircraft. Fixed-wing UAVs, which employ wings similar to traditional airplanes, are highly efficient for forward motion but lack mobility crucial to multi-copter UAVs. Multi-copter UAVs, characterized by three or more propellers, offer maneuverability and the ability to hover over targets. Among multi-copters, the quadcopter, featuring four motor-driven rotors on a fixed frame, is the most widely used [15].

A quadcopter, with its six degrees of freedom (DOF), is capable of three directional translation (X, Y, Z) and rotation (Roll, Pitch, Yaw) around three axes. The control of quadcopters poses a significant challenge due to their highly nonlinear dynamics. Therefore, robust controllers are essential, with sliding mode controllers being one such category [16,17]. However, sliding mode controllers often suffer from chattering, an undesirable phenomenon. Various strategies have been employed to mitigate chattering, including the use of saturation functions instead of discontinuous switching functions and the application of higher-order SMC methods [18-20]. However, the former reduces resilience to disturbances, and the latter presents challenges in gain tuning [21,22].

When dealing with matched perturbations, the high-order sliding mode control method can be employed to drive the sliding variable and its subsequent derivatives to zero. Nonetheless, a significant limitation of high-order SMC is its dependence on data from the high-order time derivatives of the sliding variable [23-25]. Among the higher-order sliding mode control methods, it's essential to emphasize that second-order SMC, such as the super-twisting algorithm, exclusively requires feedback from the sliding surface during the control process. The super-twisting algorithm was first proposed by Dr. Levant in 1993 [26].

Remarkably, the super-twisting sliding mode controller utilizes fixed parameters, prompting the use of a Fuzzy Logic Controller for automatic parameter adjustment in this research. The research explores the control of a quadcopter using a Fuzzy Super Twisting Sliding Mode Controller while simultaneously applying AI-based image processing techniques for wheat yield estimation.

Figure 1.1: Study Area

1.2 Problem Statement

In Ethiopia, the conventional method for estimating wheat yield involves collecting data through field visits. However, this approach is subjective, time-consuming, and prone to errors due to incomplete ground observations, ultimately leading to poor wheat yield estimations. Furthermore, the crop weather model was used for crop estimation, but it often did not consider critical factors such as soil quality and diseases, making its predictions less accurate.

Remote sensing presents an alternative with the potential to significantly improve pre-harvest crop yield estimation. Data from satellite-based sensors have been applied to estimate yields at the national level in Ethiopia. Nevertheless, many of these studies relied on low-resolution imagery, which led to suboptimal estimations. Recent advancements in image processing, coupled with the utilization of unmanned aerial platforms, have created an opportunity for faster and more accurate crop yield estimation techniques. In this study, an image processing algorithm has been developed to address these challenges.

When considering the utilization of Unmanned Aerial Vehicles (UAVs) for wheat yield estimation, it is essential to acknowledge that various physical parameters of quadcopter are influenced by diverse operating conditions. This effect becomes particularly emphasized in unpredictable outdoor environments. Traditional feedback controllers often struggle to maintain stability in such conditions, presenting a significant challenge for trajectory tracking. To tackle this problem, the suggested approach involves employing a robust control strategy referred to as the sliding mode controller. Nonetheless, it's crucial to emphasize that traditional sliding mode controllers are susceptible to chattering. To reduce chattering and improve control performance, the Super Twisting Sliding Mode Controller was introduced. Notably, this controller exerts high control effort, leading to overshoot and requiring complex parameter tuning. To overcome these challenges, Fuzzy Logic is incorporated to adjust controller parameters online, resulting in the development of a Fuzzy Super Twisting Sliding Mode Controller.

In summary, this study aims to transform wheat yield estimation practices in Ethiopia by making use of advanced image processing techniques and controlling a quadcopter using the Fuzzy Super Twisting Sliding Mode Controller.

1.3 Objective

1.3.1 General Objective

The main objective of this thesis is to design and simulate Fuzzy super twisting sliding mode controller with PID surface of quadcopter for wheat yield estimation.

1.3.2 Specific Objectives

- To define a model for the Quadcopter.
- To design a flight controller using Fuzzy Super Twisting SMC with PID surface.
- To simulate the system's controller using MATLAB.
- To count wheat head using YOLOv8
- To estimate wheat yield on a Pycharm.

1.4 Methodology

The research follows a systematic approach composed of the following stages:

1. **Literature Review:** Detailed review of relevant literature is conducted to establish foundational knowledge in the research area.
2. **Mathematical Modeling and Verification:** It is performed to describe quadrotor dynamics using the Newton-Euler method. This involves formulating mathematical relationships for translational dynamics to achieve desired positions and rotational dynamics for desired Euler angles. Model verification is conducted using MATLAB.
3. **Controller Design:** The study proceeds to controller design, where a Fuzzy super twisting sliding mode controller with PID surface is designed to control the quadcopter's motion, enabling precise trajectory tracking.
4. **Image Processing and Yield Estimation:** The next step involves the application of image processing techniques for wheat yield estimation using YOLOv8 in Pycharm. Automated detection and counting of wheat heads in images enable yield estimation based on image captured by the quadcopter.
5. **Result and Discussion:** The results consist of visual data generated from simulations in MATLAB. These visual representations illustrate the system's performance across different scenarios. Following the presentation of these graphical results, the discussion takes on the role of interpreting the findings.
6. **Conclusion and Recommendation:** The final stage entails drawing comprehensive conclusions from the research outcomes. Additionally, recommendations are provided

for potential enhancements or further investigations in the research domain, guiding future research endeavors.

1.5 Scope

This thesis encompasses several key aspects of research. A rigorous mathematical model is formulated for the quadcopter using the established Newton-Euler formulation. Building upon this model, the thesis focuses on the design and implementation of a Fuzzy Super Twisting Sliding Mode Controller to govern the quadcopter's actions, particularly in terms of tracking a predefined reference trajectory. Additionally, the study incorporates image processing techniques to estimate wheat yield, with a specific emphasis on the automatic detection and counting of wheat heads in images obtained from the quadcopter. All these aspects are simulated and analyzed using MATLAB/Simulink for the controller's performance assessment, while the image processing is done using PyCharm.

However, it's essential to acknowledge a limitation of this thesis. While it covers extensive theoretical and simulation-based investigations, it does not involve the construction of a physical prototype of the quadcopter. Instead, the research primarily focuses on modeling, control design, and simulation, offering valuable insights into these aspects within the context of wheat yield estimation and quadcopter control.

1.6 Significance

The thesis makes a significant contribution to the field by advancing wheat yield estimation through the application of image-based artificial intelligence techniques. It introduces an innovative approach that enables automatic detection and counting of wheat heads in images captured by a quadcopter, ultimately leading to precise yield estimation.

This research holds importance in enhancing the agricultural forecasting capabilities of the country. It represents a shift from the conventional approach of crop yield forecasting, which relies on stakeholder assessments and expert opinions, a process that is time-consuming and costly. Instead, it adopts an advanced remote sensing approach, which can provide more accurate and efficient predictions.

In the academic world, this thesis is very important because it provides a strong basis for future research on crop forecasting. It's like a solid foundation that future studies can build upon, helping to make new discoveries and improvements in this field.

Additionally, concerning the control of the quadcopter, the thesis incorporates an integral term into the sliding surface to reduce steady-state error. Furthermore, it employs a fuzzy logic controller to automatically adjust the parameters of the Super Twisting Sliding Mode Controller (STSMC) with PID surface. This control strategy leads to reduced chattering, increased robustness, and decreased control effort, thus enhancing the overall performance of the system.

1.7 Thesis Outline

The thesis is organized into seven chapters.

1. **Introduction:** Provides an introduction to the research topic and outlines the objectives and significance of the thesis.
2. **Literature Review:** Presents a comprehensive review of relevant literature, discussing previous research and findings related to quadcopters, control systems, image processing, and wheat yield estimation.
3. **Mathematical Modeling of the Quadcopter:** Focuses on the mathematical modeling of the quadcopter system.
4. **Design of Fuzzy Super Twisting Sliding Mode Controllers:** Details the design of Fuzzy Super Twisting Sliding Mode Controllers for the quadcopter, highlighting the control strategy employed to enhance stability and performance.
5. **Image Recognition and Wheat Yield Estimation:** Explores the application of image processing techniques, specifically in PyCharm, for wheat yield estimation by detecting and counting wheat heads in images captured by the quadcopter.
6. **Presentation of Results and Discussion:** Presents the results obtained from the simulations and engages in a thorough discussion of these results.
7. **Conclusion, Recommendations, and Future Work:** Summarizes the findings of the thesis, draws conclusions, and offers recommendations based on the research outcomes. It also outlines potential areas for future research and development in the field.

Chapter Two

Literature Review

2.1 Introduction

In the realm of research, numerous methodologies have emerged to address the control of quadcopters and the estimation of wheat yields. Quadcopters have captured the attention of researchers due to their complex operational dynamics, resulting in a wide range of exploration possibilities. In parallel, wheat yield estimation has assumed importance, given its pivotal role in ensuring food security and informing market strategies. This literature review navigates through this multifaceted research landscape, encompassing three distinct domains: quadcopter control, wheat yield estimation, and computer vision.

2.2 Quadcopter Control

Considerable research has been conducted in the field of multi-rotor control, particularly in the context of guiding them along specific trajectory. In [27], a PID and LQ controller was employed to stabilize the attitude of a quadcopter, achieved by linearizing the model around its hover position. While the controller effectively maintains stability during hovering (altitude control), it exhibits limited applicability for addressing sudden attitude perturbations, lacking robustness in such scenarios. And also it does not provide control of quadcopter for non-hovering operation.

To overcome these limitations, a nonlinear approach was employed in [28], utilizing integral backstepping to manage quadcopter's angular rotations and linear translations while dealing with aerodynamic effects. This project offered the ability to avoid collisions. In another study [29], a fractional-order backstepping SMC design was introduced for controlling the altitude and orientation of the quadcopter. It track desired trajectories quickly and reject wind disturbance. Notably, the discontinuous controller part within the SMC algorithm contributes to its ability to handle matched disturbances [30]. However, in both projects, there remains a challenge related to high-frequency oscillations that can impact system performance [31]. In [32] introduced the concept of high-order sliding mode (HOSM), which mitigates chattering phenomena [33]. This includes the development of the super twisting sliding mode controller. It's important to note that the effectiveness of the STSM relies on knowledge of perturbation bounds. In practical scenarios, drones face disturbances and parameter variations that can diminish control efficiency, necessitating online tuning of controller parameters.

In [34], the Adaptive Sliding Mode Controller is discussed, emphasizing its capacity to eliminate altitude and attitude tracking errors while ensuring asymptotic convergence to zero, even when faced with significant external disturbances. A comparison was conducted, pitting the Adaptive SMC against LQR and active disturbance rejection control. The controller demonstrated robustness against disruptions and unpredictable system behaviors not accounted for in the model. Nevertheless, it's important to note that in all these cases, the controller gains were determined through a trial-and-error process.

In [35], a Fuzzy-PI controller was created to regulate the quadcopter's position and orientation. This controller automatically adjusts its control parameters, but it exhibits a persistent steady-state error.

Inspired by prior research in the field, this paper suggests the creation of a Fuzzy super twisting SMC with PID surface to oversee the trajectory tracking of quadcopters. The paper includes simulation studies conducted in Matlab/Simulink to showcase the efficiency and success of this approach.

2.3 Wheat Yield Estimation

Conventional techniques for predicting agricultural yields primarily consist of agronomic forecasting methods [36], crop-growth models [37], and meteorological statistical methods [38]. These methods are employed to construct crop yield models from varying viewpoints. But these approaches require substantial human and material resources.

Satellite remote sensing technology has become a crucial asset in numerous fields such as resource assessments [39], agricultural progress [40], and national security applications [41]. The utilization of satellite remote sensing has demonstrated its effectiveness in predicting yields, offering advantages like easy data acquisition and broad geographical coverage [42]. However, it does have its constraints, including limited data availability, the requirement for expertise in data interpretation, sensitivity to weather conditions, limitations in image resolution, and difficulties in distinguishing between different crop types.

Over the last few decades, vegetation indices (VI) have been extensively employed for forecasting crop yields [43]. In many of these investigations, vegetation indices like the Normalized Difference Vegetation Index (NDVI) [44] have been the primary choices. However, these vegetation indices primarily convey information about the greenness of vegetation and do not completely encompass the environmental challenges affecting crops. In this thesis, wheat yield estimation is accomplished through the utilization of photos captured by a quadcopter.

2.2.1 Image Processing

While recent research in agriculture has shown promise for deep learning techniques, the fusion of image-based analytics and wheat-related applications remains relatively unexplored. To encourage future researchers to address the challenge of wheat head detection, in [45] released a comprehensive, publicly accessible dataset with annotations, facilitating further investigations in this area. Leveraging this open-source dataset, in [46] applied the YOLOv4 object detection framework to localize wheat heads using bounding boxes. Despite their successful results, their approach lacked a holistic end-to-end framework suitable for broad adoption by agronomists and farmers, especially for achieving high-throughput phenotyping in wheat head detection and counting. To address this need, this thesis focuses on automating wheat head detection and counting using YOLOv8.

Chapter Three

Modeling and Model Verification

3.1 Modeling

The mechanical structure of the quadcopter consists of four uniform arms that are evenly spaced apart. At each end of these arms, there is a motor connected to a propeller through a direct coupling mechanism. The propellers are grouped in pairs, with each pair rotating in opposite directions. The motor and propeller system together generate both thrust and torque.

3.1.1 Lift Generation

An airfoil refers to the cross-sectional shape of an object distinguished by its curved surface. The presence of an airfoil causes the airflow to curve. To achieve this curvature, there must be elevated pressure on the upper surface of the object compared to the lower surface. This difference in pressure supplies the necessary centripetal force. The increased pressure pushes the air particles downward, which is why the flow remains attached to the airfoil. This phenomenon is commonly referred to as the Coanda effect. The Coanda effect describes the inclination of a fluid jet's motion to stick to a convex surface [47].

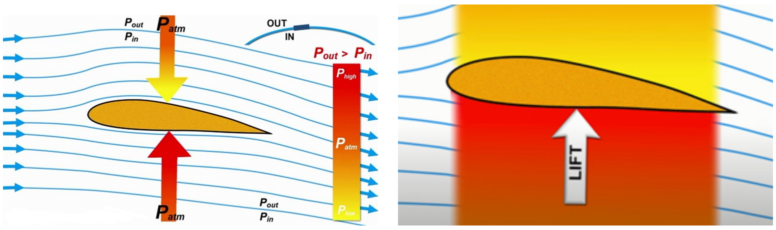


Figure 3.1: Pressure Distribution around Airfoil [48]

The airflow also curves along the underside of the airfoil, as depicted in Figure 3.1. Far from the airfoil, the pressure remains at atmospheric levels. However, in a curved flow, there is a need for higher pressure on the outer side of the curve. Consequently, as approaching the airfoil's upper surface, the pressure decreases, while it increases on the lower surface due to this curvature. This pressure disparity is the driving force behind the generation of lift. The presence of the airfoil induces this curved flow, which in turn creates the pressure difference responsible for lift. The magnitude of this lift is directly proportional to the square of the propeller's rotational speed.

Figure 3.2: Propeller [49]

A propeller is a mechanical apparatus comprised of a rotating hub and blades extending outward, positioned at an angle to create a helical spiral configuration. When this assembly is set into motion, it imparts linear thrust to a working fluid [49].

As the propeller revolves around the motor's shaft, it undergoes a phenomenon where the blade's tip covers a significantly greater distance compared to the blade root within the same time frame. Consequently, the speed of the propeller blade varies, being highest at the tip and slowest at the root. This speed differentiation occurs because as the angular distance ($\Delta\theta$) increases, so does the angular velocity ($\omega = \Delta\theta/\Delta t$). To address this speed difference, the blade is designed with a twist: it maintains a high angle of attack at the root and a low angle of attack at the tip. This twist ensures uniform thrust generation.

3.1.2 Types of Configuration

Depending on how a vehicle's blades are oriented relative to its body coordinates, there are two primary configurations commonly used in quadcopter designs, denoted as the 'Plus' and 'Cross' configurations, as depicted in Figure 3.3. In the Plus configuration, a single rotor serves as the leader for the aircraft, whereas in the Cross configuration, two rotors take the lead.

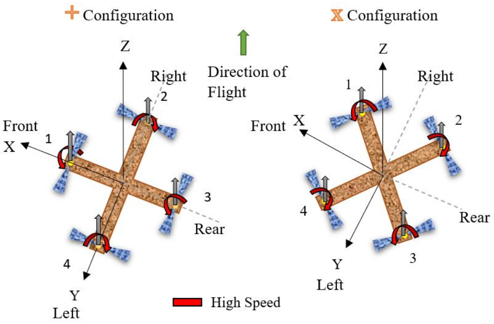


Figure 3.3: Two Types of Configuration

In the case of the Cross configuration, in pitch movement, when the two rear rotors accelerate while the two front rotors decelerate, it results in a nose-down pitching motion. Among the two rear rotors that speed up, one rotates in a clockwise (CW) direction, while the other rotates in a counterclockwise (CCW) direction, and the torques they generate cancel each other out. Similarly, when the two front rotors slow down, one turns in a CW direction, and the other in a CCW direction, once again cancelling out the torque effects. Consequently, pitch movement does not introduce a net yaw moment. Likewise, in roll movement, involving a decrease in speed for the two right rotors and an increase in speed for the two left rotors, it leads to a roll-right motion. Among the two right rotors that slow down, one rotates in a CW direction, while the other in a CCW direction, and their torques cancel out. The same holds true for the left rotors that speed up, preventing the introduction of a net yaw moment in the Cross configuration.

In the Plus configuration, during pitch movement, where a single rear rotor accelerates while a single front rotor decelerates, it results in a nose-down pitching motion. Unlike linear changes in torque with rotor speed $\tau = k_\tau \omega^2$, the increase in torque from the clockwise (CW) spinning rear rotor doesn't perfectly offset the torque reduction from the clockwise (CW) spinning front rotor, leading to a net yaw moment. Consequently, the Plus configuration requires compensation through a yaw control input. Similarly, when examining roll movement in the Plus configuration, which involves an increase in speed for a single left rotor and a decrease in speed for a single right rotor, it generates a roll-right moment.

While researchers and designers commonly use both configurations, the Cross configuration is preferred for its improved stability compared to the Plus configuration. In the Cross configuration, two rotors play a primary role in propelling the quadcopter, while the Plus configuration relies on a single rotor. Furthermore, the Plus configuration may pose challenges in terms of camera placement as it could potentially obstruct the camera with its frame.

3.1.3 Working Principle of Quadcopter

Attaining the six degrees of freedom (DOF) in a quadcopter is achieved by adjusting the speed of its four motors.

Throttle: Achieving vertical take-off and landing is made possible through throttling, a process where all propeller speeds are simultaneously set to equal values. This enables the generation of vertical thrust in relation to the vehicle's body reference frame. When the vehicle is in a horizontal, untilted position, this command results in the generation of purely vertical forces. Figure 3.4 provides a visual representation of a quadcopter responding to a throttle command.

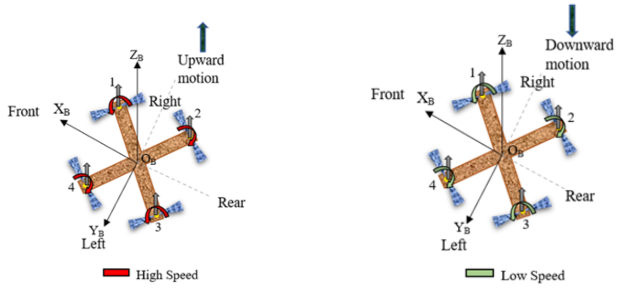


Figure 3.4: Throttle

Roll: Roll torque can be produced by either reducing the speed of the two left-side propellers while increasing the speed of the remaining right-side propellers or vice versa. This action generates torque around the X-axis of the vehicle. When executing a roll command, it's essential to maintain the overall effective thrust to ensure that only roll acceleration occurs, enabling the vehicle to maneuver along the Y-axis of the body reference frame, as depicted in Figure 3.5.

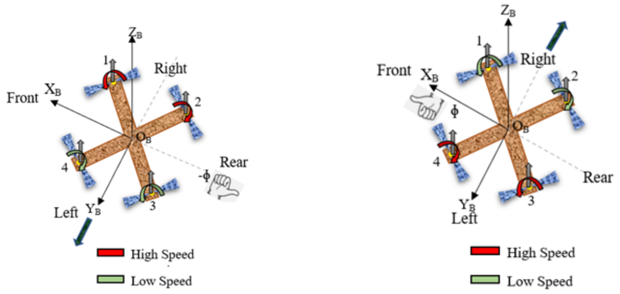


Figure 3.5: Roll

Pitch: The process of generating pitching movement is similar to that of rolling, with the difference being the axis of moment generation. In pitching, the movement is achieved by either reducing the speed of the two front propellers while increasing the speed of the rear ones or vice versa. This action results in torque being generated along the Y-axis of the body reference frame, causing the quadcopter to move in the X direction, as illustrated in Figure 3.6. Similar to the roll command, maintaining a constant effective upward thrust ensures that only pitch acceleration is produced.

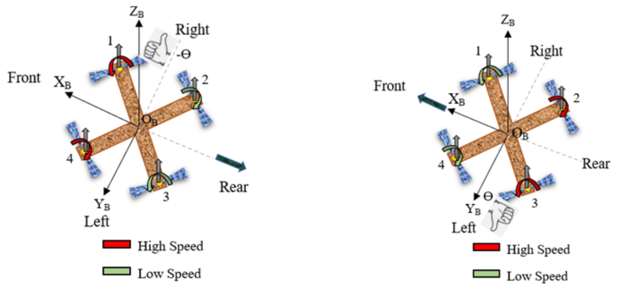


Figure 3.6: Pitch

Yaw: Yaw movement is accomplished by adjusting the rotation of each pair of propellers, with one pair spinning counterclockwise while the other spins clockwise. As a result of the imbalance in overall torque, the quadcopter rotates around the Z-axis of its body reference frame, as depicted in Figure 3.7. It's worth noting that a clockwise-rotating propeller generates torque in the same direction as its rotation, not in the opposite direction. This means that, when viewed from above, a clockwise-rotating propeller induces counterclockwise torque (referred to as 'positive torque') on the quadcopter. Conversely, a counterclockwise-rotating propeller creates clockwise torque (termed 'negative torque') when observed from above.

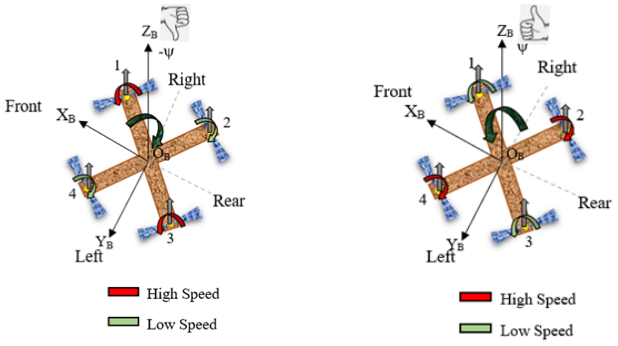


Figure 3.7: Yaw

3.1.4 Kinematics of Quadcopter

Kinematics, a subfield within mechanics, is concerned with the study of an object or system's motion in isolation from the forces and torques acting upon it. In the context of describing the motion of a rigid body with six degrees of freedom (6 DOF), it is advantageous to define two reference frames.

- **Reference Frame Alignment**

To comprehensively describe the behavior of a quadcopter, it is crucial to establish two key reference frames: the Body reference frame and the Inertial reference frame. The need for these two frames arises from the diverse sensors equipped on the quadcopter. Sensors like accelerometer and gyroscope provide measurement in relation to the body frame, aiding in the assessment of linear and angular velocity. On the other hand, sensors such as GPS and magnetometer deliver readings relative to the inertial frame, facilitating the determination of linear and angular position. As a result, a method of transformation is required to formulate system equations consistently within a single reference frame.

The Body reference frame is linked to the quadcopter itself, with its origin situated at the center of gravity (COG) of the vehicle. Given the quadcopter's symmetrical design, featuring a central core and four identical rotors mounted on its arms, the COG naturally aligns with the core's center. In contrast, the Inertial reference frame is typically represented using North-East-Up coordinate systems. The inertial frame's origin is selected as a fixed point on the Earth's surface, accounting for all the forces and torques acting upon the quadcopter.

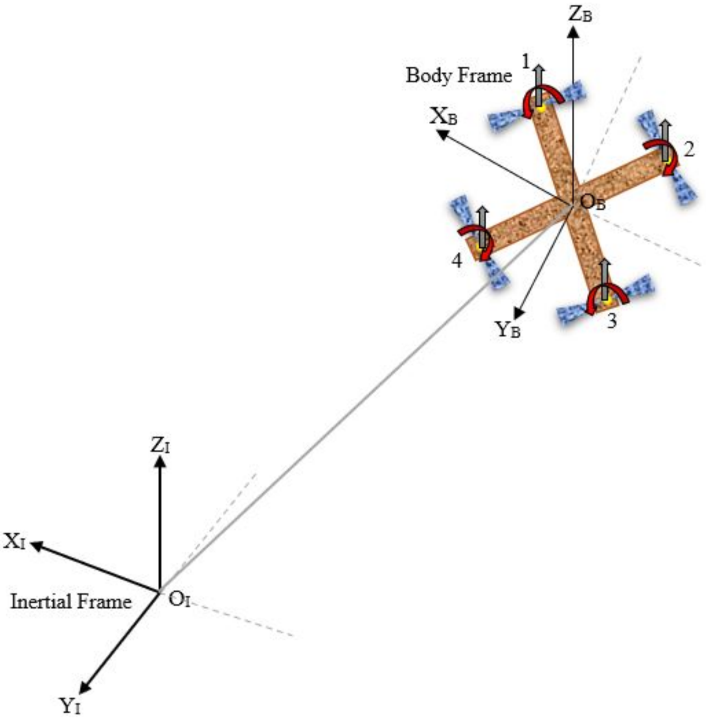


Figure 3.8: Reference Frame

A quadcopter possesses six degrees of freedom, which implies that six variables are required to define its position and orientation in three-dimensional space. To explain the quadcopter's motion, two distinct reference frames are established:

1. **Body Reference Frame** $(O_B; Z_B; Y_B; X_B)$:

- O_B coincides with the center of the quadcopter.
- Linear Velocity $(V^B = [v_x; v_y; v_z]^T)$
- Angular Velocity $(\omega^B = [\Omega_\phi; \Omega_\theta; \Omega_\psi]^T)$
- Force $(\vec{F}^B)$
- Torque (τ^B)

2. **Inertial Reference Frame** $(O_I; X_I; Y_I; Z_I)$:

- This frame is used to define the linear position $(\eta^I = [X; Y; Z]^T)$ and the angular position $(\zeta^I = [\phi, \theta, \psi]^T)$.
- η_I is determined by coordinates of the vector between O_B and O_I with respect to the Inertial frame.

- ζ_I is the orientation of the body frame with respect to the Inertial frame.

The right-hand rule is used to determine the direction of the coordinate systems.
Thumb: Positive direction of Z-axis
Index Finger: Positive direction of X-axis
Middle Finger: Positive direction of Y-axis

In the subsequent analysis, Euler angle parametrization is employed for representing the rotation of the rigid body. Euler angles, consisting of three angles (ϕ, θ, ψ), are utilized to describe the orientation of a rigid body in relation to a fixed coordinate system. The chosen convention is the Z-Y-X convention.

Assumptions:

→ The quadcopter is considered a solid and unyielding structure.
→ It possesses axis-symmetrical characteristics.
→ The analysis does not account for aerodynamic effects such as blade-flapping.
→ Thrust and torque are directly proportional to the square of rotor speed, represented as $\vec{F_i} = k_t\omega^2$ and $\tau_i = k_\tau\omega^2$, respectively.
→ The center of mass of the quadcopter coincides with the origin of the body frame.

- **Rotation Matrix**

Rotation matrices are 3×3 matrices characterized by orthogonality and a determinant of one. While Euler angles are commonly used to depict rotations, alternative methods like quaternions exist. Quaternions provide an attitude representation without singularities. In contrast to Euler angles, which involve three angles, quaternionic attitude representation utilizes a single axis and a rotation angle. It's important to note that quaternions lack the straightforward physical interpretation that Euler angles offer. For the purposes of this thesis, Euler angle representation is adopted because the application of this thesis doesn't require acrobatic movements.

Euler angles are defined by a composition of rotations, where elemental rotations around specific axes of a coordinate system suffice to attain any desired orientation. In the forthcoming analysis, the rotation of a rigid body in 3-dimensional space is expressed through the composition of three consecutive rotations. Following the Z-Y-X convention, these rotations begin with a rotation around the Z axis (ψ,Yaw), followed by a rotation around the Y axis (θ, Pitch), and lastly, a rotation around the X axis (ϕ, Roll).

The rotation from the body frame to the inertial frame, denoted as ${}^I_B R$, can be expressed as:

$$
{}^I_B R = \begin{bmatrix} c(\theta)c(\psi) & s(\phi)s(\theta)c(\psi) - c(\phi)s(\psi) & s(\phi)s(\psi) + c(\phi)s(\theta)c(\psi) \\ c(\theta)s(\psi) & c(\phi)c(\psi) + s(\phi)s(\theta)s(\psi) & c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi) \\ -s(\theta) & s(\phi)c(\theta) & c(\phi)c(\theta) \end{bmatrix} \tag{3.1}
$$

Where: $c(\theta)$ corresponds to $\cos(\theta)$, $s(\theta)$ corresponds to $\sin(\theta)$, and similar notation is used for ϕ and ψ.

The rotation matrix ${}^I_B R$ is employed to transform linear quantities from the body (rotating frame) to the Inertial frame. To obtain the transformation from the Inertial reference frame to the Body reference frame, it is achieved by transposing the Body reference frame into the Inertial reference frame, denoted as $({}^B_I R) = ({}^I_B R)^T$. Details regarding the successive rotations from the rotating frame to the Inertial frame are provided in Appendix A.1.

- **Transfer Matrix**

The angular position vector in the Inertial frame is represented as $\boldsymbol{\zeta}^I = [\phi, \theta, \psi]^T$, while the angular velocity vector in the Body frame is denoted as $\boldsymbol{\omega}^B = [\Omega_\phi, \Omega_\theta, \Omega_\psi]^T$. To establish a connection between these two vectors, a transfer matrix denoted as $\mathbf{T}$ is required, which can be expressed as $\dot{\boldsymbol{\zeta}}^I = \mathbf{T}\boldsymbol{\omega}^B$.

$$
\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \mathbf{T} \begin{bmatrix} \Omega_\phi \\ \Omega_\theta \\ \Omega_\psi \end{bmatrix} \tag{3.2}
$$

Where:

$$
\mathbf{T}^{-1} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix}, \quad \mathbf{T} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \tag{3.3}
$$

The details about the derivation of the transfer matrix are provided in Appendix A.2.

3.1.5 Dynamics of Quadcopter

Dynamics is a field within mechanics that explores the influence of forces and torques on the movement of an object or system. Various methodologies exist for deriving equations governing the motion of a rigid body with 6 degrees of freedom. In this thesis, the Newton-Euler method is employed.

Newton-Euler Approach

The modeling of quadcopter flight dynamics is achieved through the application of the Newton-Euler method. The equations governing the motion are formulated by combining the translational and rotational characteristics of the six-degree-of-freedom system, encompassing three positional variables and three angular orientations. These equations are derived based on Euler's fundamental principles of motion, which encompass Newton's second law and Newton's first law, also known as the law of inertia.

- **Analysis of Forces**

Force plays a role in influencing the translational motion of a quadcopter. According to Newton's second law, the forces acting upon a quadcopter can be represented as: $\sum \vec{F} = \vec{F_T} + \vec{F_g} = m\vec{a}$, where $\vec{F_T}$ signifies the thrust force, and $\vec{F_g}$ denotes the gravitational force. Examination of the quadcopter's translational dynamics can be conducted using reference frames in both Inertial and Body perspective.

1. Body Reference Frame:

Thrust Force: The thrust force ($\vec{F_T}$) encompasses the cumulative thrust produced by the quadcopter's propellers. This force exhibits a quadratic relationship with angular velocity, as expressed by $\vec{F_i} = k_t \omega^2$, where k_t denotes the propeller thrust coefficient. This coefficient remains constant and is influenced by various factors, including air density, the propeller's swept area, and more. Its determination is typically derived through empirical experimentation and observations, with ω representing angular velocity.

$$\vec{F}_T^B = \vec{F}_{T1} + \vec{F}_{T2} + \vec{F}_{T3} + \vec{F}_{T4} \tag{3.4}$$

$$\vec{F}_T^B = k_t\omega_1^2 + k_t\omega_2^2 + k_t\omega_3^2 + k_t\omega_4^2 = k_t(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \tag{3.5}$$

Gravitational Force: The gravitational force ($\vec{F_g}$) represents the influence of gravity upon the quadcopter. It exclusively operates in the Z-direction and is more effectively described within the Inertial frame. To express it in relation to the Body frame, a coordinate transformation from the Inertial frame to the Body frame is required.

$$\vec{F}_g^B = ({}_I^B\mathbf{R})\vec{F}_g^I = ({}_I^B\mathbf{R})\begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} = \begin{bmatrix} mgsin(\theta) \\ -mgcos(\theta)sin(\phi) \\ -mgcos(\theta)cos(\phi) \end{bmatrix} \tag{3.6}$$

Utilizing Newton's second law, which states $\vec{F} = m\vec{a}$, within the Body frame:

$$\sum \vec{F}^B = m\frac{d\vec{v}}{dt}, \quad \vec{v} = \vec{v}_x + \vec{v}_y + \vec{v}_z \tag{3.7}$$

To compute acceleration by finding the derivative of $\vec{v}$ in the Inertial frame:

$$\frac{d\vec{v}}{dt} = \frac{du}{dt}\mathbf{i}_x + \frac{dv}{dt}\mathbf{j}_y + \frac{dw}{dt}\mathbf{k}_z + \frac{di_x}{dt}u + \frac{dj_y}{dt}v + \frac{dk_z}{dt}w \tag{3.8}$$

Since the coordinate axes are in the inertial frame and remain constant, thus: $\frac{di_x}{dt} = \frac{dj_y}{dt} = \frac{dk_z}{dt} = 0$. So it results:

$$\frac{d\vec{v}}{dt} = \frac{du}{dt}i_x + \frac{dv}{dt}j_y + \frac{dw}{dt}k_z.$$

When dealing with a rotating body, it's important to consider the application of the chain rule, as explained in Appendix B. Applying the chain rule yields the following outcomes:

$$\frac{dv}{dt} = \frac{dv_x}{dt}\mathbf{i} + \frac{dv_y}{dt}\mathbf{j} + \frac{dv_z}{dt}\mathbf{k} + (\boldsymbol{\omega} \times \mathbf{i})v_x + (\boldsymbol{\omega} \times \mathbf{j})v_y + (\boldsymbol{\omega} \times \mathbf{k})v_z \tag{3.9}$$

The force acting within the Body frame is:

$$\sum \mathbf{F}^B = m\frac{dv}{dt} = m\left(\frac{dv_x}{dt}\mathbf{i} + \frac{dv_y}{dt}\mathbf{j} + \frac{dv_z}{dt}\mathbf{k}\right) + m\left((\boldsymbol{\omega} \times \mathbf{i})v_x + (\boldsymbol{\omega} \times \mathbf{j})v_y + (\boldsymbol{\omega} \times \mathbf{k})v_z\right) \tag{3.10}$$

Where: $m\left((\boldsymbol{\omega} \times \mathbf{i})v_x + (\boldsymbol{\omega} \times \mathbf{j})v_y + (\boldsymbol{\omega} \times \mathbf{k})v_z\right)$ denotes the Coriolis force, which is a pseudo-force experienced by objects in motion within a reference frame that rotates relative to an inertial frame[50]. It can be mathematically expressed as:

$$\text{Coriolis force} = m(\boldsymbol{\omega} \times (v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k})) \tag{3.11}$$

After performing the cross multiplication:

$$\sum \mathbf{F}^B = \begin{bmatrix} \vec{F}_x^B \\ \vec{F}_y^B \\ \vec{F}_z^B \end{bmatrix} = \begin{bmatrix} m(a_x + (\Omega_\theta v_z - \Omega_\psi v_y)) \\ m(a_y + (\Omega_\psi v_x - \Omega_\phi v_z)) \\ m(a_z + (\Omega_\phi v_y - \Omega_\theta v_x)) \end{bmatrix} \tag{3.12}$$

$\sum \vec{F}^B = \vec{F}_T^B + \vec{F}_g^B = m\vec{a}$, where $\vec{F}_T^B$ represents the thrust force within the Body frame, and $\vec{F}_g^B$ represents the gravitational force within the Body frame.

$$\begin{bmatrix} 0 \\ 0 \\ F_{T1} + F_{T2} + F_{T3} + F_{T4} \end{bmatrix} + \begin{bmatrix} mg\sin(\theta) \\ -mg\cos(\theta)\sin(\phi) \\ -mg\cos(\theta)\cos(\phi) \end{bmatrix} = \begin{bmatrix} m(a_x + (\Omega_\theta v_z - \Omega_\psi v_y)) \\ m(a_y + (\Omega_\psi v_x - \Omega_\phi v_z)) \\ m(a_z + (\Omega_\phi v_y - \Omega_\theta v_x)) \end{bmatrix}$$

Now, by dividing both sides by mass and reorganizing the equation:

$$\begin{aligned} a_x &= g\sin(\theta) - \Omega_\theta v_z + \Omega_\psi v_y \\ a_y &= -g\cos(\theta)\sin(\phi) - \Omega_\psi v_x + \Omega_\phi v_z \\ a_z &= -g\cos(\theta)\cos(\phi) - \Omega_\phi v_y + \Omega_\theta v_x \end{aligned} \tag{3.13}$$

Where: $a_x = \frac{dv_x}{dt}$, $a_y = \frac{dv_y}{dt}$, and $a_z = \frac{dv_z}{dt}$. This constitutes the dynamic equation describing translational motion within the Body reference frame.

2. Inertial Reference Frame:

Thrust Force: The thrust force is initially expressed within the Body frame, and its description in the Inertial frame can be obtained through the rotation matrix ${}^{I}_{B}R$.

$$\vec{F}_T^I = ({}^{I}_{B}R)\vec{F}_T^B \tag{3.14}$$

$$\vec{F}_T^I = \begin{bmatrix} (\cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi))\,\vec{F}_T^B \\ (\cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi))\,\vec{F}_T^B \\ \cos(\phi)\cos(\theta)\vec{F}_T^B \end{bmatrix} \tag{3.15}$$

Gravitational Force: The gravitational force is described within the Inertial frame.

$$\vec{F}_g^I = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \tag{3.16}$$

Applying Newton's second law, which states $\vec{F} = m\vec{a}$, within the Inertial reference frame, the equation becomes:

$$\sum \vec{F}^I = \vec{F}_T^I + \vec{F}_g^I = m\vec{a} \tag{3.17}$$

$$\sum \vec{F}^I = \begin{bmatrix} (\cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi))\,\vec{F}_T^B \\ (\cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi))\,\vec{F}_T^B \\ \cos(\phi)\cos(\theta)\vec{F}_T^B \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} = m\begin{bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{bmatrix} \tag{3.18}$$

After dividing both sides by mass and rearranging the equation:

$$\begin{bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{bmatrix} = \begin{bmatrix} (\cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi))\frac{\vec{F}_T^B}{m} \\ (\cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi))\frac{\vec{F}_T^B}{m} \\ \cos(\phi)\cos(\theta)\frac{\vec{F}_T^B}{m} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \tag{3.19}$$

Defining U_1 as $\vec{F}_T^B$, the equation is structured as:

$$\begin{aligned}\ddot{X} &= (\cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi))\frac{U_1}{m} \\ \ddot{Y} &= (\cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi))\frac{U_1}{m} \\ \ddot{Z} &= \cos(\phi)\cos(\theta)\frac{U_1}{m} - g\end{aligned} \tag{3.20}$$

This constitutes the dynamic equation describing translational motion within the Inertial reference frame.

- **Analysis of Torque**

Torque plays a crucial role in influencing the rotational motion of a quadcopter. Similar to the previously derived translational dynamics, the rotational dynamics can be expressed as follows, taking into account the net torque and reaction forces: $\sum \tau = \tau_T + \tau_g$. Here, τ_T represents thrust torque, and τ_g signifies gyroscopic torque. The quadcopter's rotational dynamics can be analyzed from both reference frames:

1. **Body Reference Frame:**

Moments of trust forces: $\sum(r_i X \vec{F}_i)$ These reaction moments demonstrate a quadratic relationship with angular velocity, described as: $\tau_{Ti} = k_\tau \omega_i^2$ Here, k_τ is a constant influenced by propeller characteristics such as the number of blades, diameter, material, and air viscosity. The torque analysis derivation is presented in Appendix C.1.

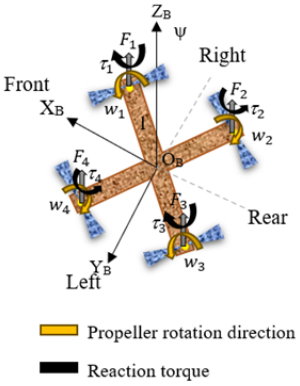


Figure 3.9: Moments of Trust Forces

$$\sum \tau_x^B = \vec{F}_1 l\cos 45° - \vec{F}_2 l\cos 45° - \vec{F}_3 l\cos 45° + \vec{F}_4 l\cos 45° = \frac{l\sqrt{2}}{2}(\vec{F}_1 - \vec{F}_2 - \vec{F}_3 + \vec{F}_4) \tag{3.21}$$

$$\sum \tau_y^B = -\vec{F}_1 l\sin 45° - \vec{F}_2 l\sin 45° + \vec{F}_3 l\sin 45° + \vec{F}_4 l\sin 45° = \frac{l}{\sqrt{2}}(-\vec{F}_1 - \vec{F}_2 + \vec{F}_3 + \vec{F}_4) \tag{3.22}$$

$$\sum \tau_z^B = -\tau_1 + \tau_2 - \tau_3 + \tau_4 \tag{3.23}$$

In this thesis, counterclockwise (CCW) direction is considered positive, while clockwise (CW) direction is regarded as negative.

$$\sum \tau_x^B = \frac{lk_t}{\sqrt{2}}(\omega_1^2 + \omega_2^2 - \omega_3^2 - \omega_4^2) \tag{3.24}$$

$$\sum \tau_y^B = \frac{lk_t}{\sqrt{2}}(\omega_1^2 + \omega_4^2 - \omega_2^2 - \omega_3^2) \tag{3.25}$$

$$\sum \tau_z^B = k_\tau(-\omega_1^2 - \omega_3^2 + \omega_2^2 + \omega_4^2) \tag{3.26}$$

Let:

$$U_2 = \tau_x^B = \frac{lk_t}{\sqrt{2}}(\omega_1^2 + \omega_2^2 - \omega_3^2 - \omega_4^2) \tag{3.27}$$

$$U_3 = \tau_y^B = \frac{lk_t}{\sqrt{2}}(\omega_1^2 - \omega_2^2 - \omega_3^2 + \omega_4^2) \tag{3.28}$$

$$U_4 = \tau_z^B = -k_\tau(\omega_1^2 + \omega_3^2 - \omega_2^2 - \omega_4^2) \tag{3.29}$$

Gyroscopic effect: When the propeller's plane of rotation experiences a change in angular velocity (Ω), it gives rise to the generation of torque. Conversely, when the plane of rotation remains constant, there is no torque induced. This means that during yaw movements of the quadcopter, there are no gyroscopic effects. However, during pitch or roll maneuvers, which involve frequent changes in the plane of rotation, torque is generated. This resulting torque is referred to as gyroscopic torque and represents the force applied to the rotating body, causing a shift in its axis of rotation.

$$\tau_g = J\alpha_g \tag{3.30}$$

Where: τ_g = gyroscopic torque, J = moment of inertia about propeller axis, α_g = gyroscopic acceleration.

$$\tau_g = J(\omega \times \Omega) \tag{3.31}$$

Where: $\omega = \begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix}$ and $\Omega = \begin{bmatrix} \Omega_\phi \\ \Omega_\theta \\ \Omega_\psi \end{bmatrix}$

In the horizontal plane of the quadcopter, four propellers are present, and each of them contributes to this gyroscopic torque. The derivation of the gyroscopic effect is provided in Appendix C.2.

$$\tau_{g\text{ total}} = \tau_{g1} + \tau_{g2} + \tau_{g3} + \tau_{g4} \tag{3.32}$$

$$\tau_{g\ \text{total}} = \begin{bmatrix} \Omega_\theta(-\omega_1 + \omega_2 - \omega_3 + \omega_4) \\ \Omega_\phi(\omega_1 - \omega_2 + \omega_3 - \omega_4) \\ 0 \end{bmatrix} \tag{3.33}$$

Let $\omega_r = -\omega_1 + \omega_2 - \omega_3 + \omega_4$, so:

$$\tau^B_{g\ \text{total}} = J \begin{bmatrix} \Omega_\theta \omega_r & -\Omega_\phi \omega_r & 0 \end{bmatrix}^T \tag{3.34}$$

In summary, torque is generated when the propeller's plane of rotation undergoes angular changes at a rate (Ω), resulting in torque induction during the quadcopter's rolling and pitching maneuvers. Conversely, when the plane of rotation remains constant, gyroscopic effects are absent. During pitch or roll maneuvers of the quadcopter, which involve frequent changes in the plane of rotation, torque is induced, known as gyroscopic torque. This gyroscopic torque is combined with the torque produced by the propeller, as it has components in both the X and Y directions.

Euler's rotation equations represent a set of first-order ordinary differential equations that describe the rotation of a rigid body. These equations are formulated in a reference frame that rotates along with the body and has fixed axes attached to it. The equation is:

$$\tau^B = I\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (I\boldsymbol{\omega}) \tag{3.35}$$

Inertia Tensor: serves as a convenient method to consolidate all the moments of inertia of an object into a single quantity. The inertia tensor of a rigid body can be represented as:

$$I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \tag{3.36}$$

I_{xx}: Represents the moment of inertia about X-axis when objects are rotated about the X-axis.
I_{xy}: Represents the Y-axis moment of inertia for objects rotated about the X-axis.
$I_{xy} = I_{yx}$, $I_{xz} = I_{zx}$, and $I_{zy} = I_{yz}$. In the case of objects with geometric symmetry like a standard multi-copter, I_{xy}, I_{xz}, and I_{yz} are all equal to zero. So equation 3.35 becomes:

$$\tau^B = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{\Omega}_\phi \\ \dot{\Omega}_\theta \\ \dot{\Omega}_\psi \end{bmatrix} + \begin{bmatrix} \Omega_\phi \\ \Omega_\theta \\ \Omega_\psi \end{bmatrix} \times \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \Omega_\phi \\ \Omega_\theta \\ \Omega_\psi \end{bmatrix} = \begin{bmatrix} I_{xx}(\dot{\Omega}_\phi) + \Omega_\theta \Omega_\psi (I_{zz} - I_{yy}) \\ I_{yy}(\dot{\Omega}_\theta) + \Omega_\phi \Omega_\psi (I_{xx} - I_{zz}) \\ I_{zz}(\dot{\Omega}_\psi) + \Omega_\phi \Omega_\theta (I_{yy} - I_{xx}) \end{bmatrix} \tag{3.37}$$

The derivation is in the appendix C.3

$$\sum \tau = \tau_T + \tau_g = I\dot{\omega} + \omega \times (I\omega) \tag{3.38}$$

$$\begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} + J \begin{bmatrix} \Omega_\theta(\omega_r) \\ \Omega_\phi(-\omega_r) \\ 0 \end{bmatrix} = \begin{bmatrix} I_{xx}(\dot{\Omega}_\phi) + \Omega_\theta\Omega_\psi(I_{zz} - I_{yy}) \\ I_{yy}(\dot{\Omega}_\theta) + \Omega_\phi\Omega_\psi(I_{xx} - I_{zz}) \\ I_{zz}(\dot{\Omega}_\psi) + \Omega_\phi\Omega_\theta(I_{yy} - I_{xx}) \end{bmatrix} \tag{3.39}$$

By rearranging equation (3.39)

$$\begin{aligned} \dot{\Omega}_\phi &= \frac{U_2 + J\Omega_\theta(\omega_r) - \Omega_\theta\Omega_\psi(I_{zz} - I_{yy})}{I_{xx}} \\ \dot{\Omega}_\theta &= \frac{U_3 + J\Omega_\phi(-\omega_r) - \Omega_\phi\Omega_\psi(I_{xx} - I_{zz})}{I_{yy}} \\ \dot{\Omega}_\psi &= \frac{U_4 - \Omega_\phi\Omega_\theta(I_{yy} - I_{xx})}{I_{zz}} \end{aligned} \tag{3.40}$$

2. Inertial Reference Frame:

The description of dynamics in the inertial reference frame is achieved through the use of the transfer matrix:

$$\begin{bmatrix} \ddot{\phi} & \ddot{\theta} & \ddot{\psi} \end{bmatrix}^T = T \begin{bmatrix} \dot{\Omega}_\phi & \dot{\Omega}_\theta & \dot{\Omega}_\psi \end{bmatrix}^T \tag{3.41}$$

$$T = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix} \tag{3.42}$$

$$\ddot{\phi} = \dot{\Omega}_\phi + \dot{\Omega}_\theta \sin\phi\tan\theta + \dot{\Omega}_\psi \cos\phi\tan\theta \tag{3.43}$$

$$\ddot{\theta} = \dot{\Omega}_\theta \cos\phi - \dot{\Omega}_\psi \sin\phi \tag{3.44}$$

$$\ddot{\psi} = \dot{\Omega}_\theta \sin\phi\sec\theta + \dot{\Omega}_\psi \cos\phi\sec\theta \tag{3.45}$$

Under the small angle approximation, for angles θ that are small, $\sin(\theta)$ and $\tan(\theta)$ become nearly zero, while $\cos(\theta)$ approximates to one. The same is true for ϕ and ψ. Consequently, the transfer matrices simplify to the identity matrix.

$$\begin{bmatrix} \ddot{\phi} & \ddot{\theta} & \ddot{\psi} \end{bmatrix}^T = \begin{bmatrix} \dot{\Omega}_\phi & \dot{\Omega}_\theta & \dot{\Omega}_\psi \end{bmatrix}^T$$

Quadcopter Characteristics: The quadcopter is influenced by four control inputs denoted as the vector $\mathbf{U} = [U_1, U_2, U_3, U_4]^T$. Specifically, U_1 controls altitude, governing the lift force. U_2 impacts the roll angle, U_3 influences the pitch angle, and U_4 plays a role in adjusting the yaw angle of the quadcopter.

$$\ddot{X} = (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)\left(\frac{U_1}{m}\right) \tag{3.46}$$

$$\ddot{Y} = (\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)\left(\frac{U_1}{m}\right) \tag{3.47}$$

$$\ddot{Z} = (\cos\phi\cos\theta)\left(\frac{U_1}{m}\right) - g \tag{3.48}$$

$$\ddot{\phi} = \frac{U_2}{I_{xx}} + \frac{(I_{yy} - I_{zz})}{I_{xx}}\dot{\theta}\dot{\psi} + \frac{J\dot{\theta}\omega_r}{I_{xx}} \tag{3.49}$$

$$\ddot{\theta} = \frac{U_3}{I_{yy}} + \frac{(I_{zz} - I_{xx})}{I_{yy}}\dot{\phi}\dot{\psi} - \frac{J\dot{\phi}\omega_r}{I_{yy}} \tag{3.50}$$

$$\ddot{\psi} = \frac{U_4}{I_{zz}} + \frac{(I_{xx} - I_{yy})}{I_{zz}}\dot{\phi}\dot{\theta} \tag{3.51}$$

The control inputs influence the rotor's angular velocity, and this connection is described inversely as:

$$\omega_1 = \sqrt{\frac{U_1}{4k_t} + \sqrt{2}\frac{U_2}{4k_t l} + \sqrt{2}\frac{U_3}{4k_t l} - \frac{U_4}{4k_\tau}} \tag{3.52}$$

$$\omega_2 = \sqrt{\frac{U_1}{4k_t} + \sqrt{2}\frac{U_2}{4k_t l} - \sqrt{2}\frac{U_3}{4k_t l} + \frac{U_4}{4k_\tau}} \tag{3.53}$$

$$\omega_3 = \sqrt{\frac{U_1}{4k_t} - \sqrt{2}\frac{U_2}{4k_t l} - \sqrt{2}\frac{U_3}{4k_t l} - \frac{U_4}{4k_\tau}} \tag{3.54}$$

$$\omega_4 = \sqrt{\frac{U_1}{4k_t} - \sqrt{2}\frac{U_2}{4k_t l} + \sqrt{2}\frac{U_3}{4k_t l} + \frac{U_4}{4k_\tau}} \tag{3.55}$$

Since the quadcopter is an underactuated system, it is controlled by four input variables denoted as U_1, U_2, U_3, and U_4, while it exhibits six degrees of freedom (DOF) represented as X, Y, Z, ϕ, θ, and ψ. To achieve independent control, it becomes necessary to transform the underactuated system into a fully actuated one. Assuming no external forces act on the quadcopter, only the thrust force and gravitational force are considered. In the body frame, the thrust force $\vec{F}_T^B$ is denoted as U_1. To transition from the body frame to the inertial frame, a rotational matrix is employed.

Consider the virtual control inputs in the X, Y, and Z directions, denoted as U_x, U_y, and U_z.

$$\begin{bmatrix} U_x \\ U_y \\ U_z \end{bmatrix} = \begin{bmatrix} c(\theta)c(\psi) & s(\phi)s(\theta)c(\psi) - c(\phi)s(\psi) & s(\phi)s(\psi) + c(\phi)s(\theta)c(\psi) \\ c(\theta)s(\psi) & c(\theta)c(\psi) + s(\phi)s(\theta)s(\psi) & -s(\phi)c(\psi) + c(\phi)s(\theta)s(\psi) \\ -s(\theta) & s(\phi)c(\theta) & c(\phi)c(\theta) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{3.56}$$

$$\begin{bmatrix} U_x \\ U_y \\ U_z \end{bmatrix} = \begin{bmatrix} (s(\phi)s(\psi) + c(\phi)s(\theta)c(\psi))U_1 \\ (-s(\phi)c(\psi) + c(\phi)s(\theta)s(\psi))U_1 \\ (c(\phi)c(\theta))U_1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{3.57}$$

Finding the magnitude of the vector:

$$U_1 = \sqrt{U_x^2 + U_y^2 + (U_z + mg)^2} \tag{3.58}$$

Since:

$\mathrm{U}_x = (\sin(\phi)\sin(\psi) + \cos(\phi)\sin(\theta)\cos(\psi))U_1$

$\mathrm{U}_y = (-\sin(\phi)\cos(\psi) + \cos(\phi)\sin(\theta)\sin(\psi))U_1$

$\mathrm{U}_z = (\cos(\phi)\cos(\theta))U_1 - mg$

So,

$$\ddot{X} = \frac{U_x}{m} \tag{3.59}$$

$$\ddot{Y} = \frac{U_y}{m} \tag{3.60}$$

$$\ddot{Z} = \frac{U_z + mg}{m} - g = \frac{U_z}{m} + \frac{mg}{m} - g = \frac{U_z}{m} + g - g = \frac{U_z}{m} \tag{3.61}$$

To find the desired Roll (ϕ) and Pitch (θ) angles:

$$\theta_d = \sin^{-1}\left[\frac{\ddot{X}m}{U_1} - \sin(\phi)\sin(\psi)\frac{1}{\cos(\phi)\cos(\psi)}\right] \tag{3.62}$$

$$\phi_d = \sin^{-1}\left(\frac{m}{U_1}(\ddot{X}\sin(\psi) - \ddot{Y}\cos(\psi))\right) \tag{3.63}$$

The derivation is in the appendix D.

3.1.6 State Space Representation

A state-space representation is a method for characterizing the dynamics of a physical system using state variables and first-order differential equations. The state of a quadcopter is a collection of variables that define its status at a specific time, encompassing attributes like position, velocity, orientation, and angular velocity. These state variables constitute the quadcopter's state, including parameters like the X, Y, and Z coordinates of its position, the roll, pitch, and yaw angles of its orientation, and their respective derivatives [51].

Let the state variables be: X, $\dot{X}$, Y, $\dot{Y}$, Z, $\dot{Z}$, ϕ, $\dot{\phi}$, θ, $\dot{\theta}$, ψ, and $\dot{\psi}$, denoted as $X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9, X_{10}, X_{11}, X_{12}$:

$$a_1 = \frac{I_{yy} - I_{zz}}{I_{xx}}, \quad a_2 = \frac{J \cdot W_r}{I_{xx}}, \quad b_1 = \frac{1}{I_{xx}}$$

$$a_3 = \frac{I_{zz} - I_{xx}}{I_{yy}}, \quad a_4 = \frac{J \cdot W_r}{I_{yy}}, \quad b_2 = \frac{1}{I_{yy}}$$

$$a_5 = \frac{I_{xx} - I_{yy}}{I_{zz}}, \quad b_3 = \frac{1}{I_{zz}}$$

The state-space representation for the quadcopter is as follows:

$$
\begin{aligned}
\dot{X}_1 &= X_2\\
\dot{X}_2 &= \frac{U_x}{m}\\
\dot{X}_3 &= X_4\\
\dot{X}_4 &= \frac{U_y}{m}\\
\dot{X}_5 &= X_6\\
\dot{X}_6 &= \frac{U_z}{m}\\
\dot{X}_7 &= X_8\\
\dot{X}_8 &= b_1U_2 + a_1X_{10}X_{12} + a_2X_{10}\\
\dot{X}_9 &= X_{10}\\
\dot{X}_{10} &= b_2U_3 + a_3X_8X_{12} - a_4X_8\\
\dot{X}_{11} &= X_{12}\\
\dot{X}_{12} &= b_3U_4 + a_5X_8X_{10}
\end{aligned}
$$

3.2 Modeling Verification

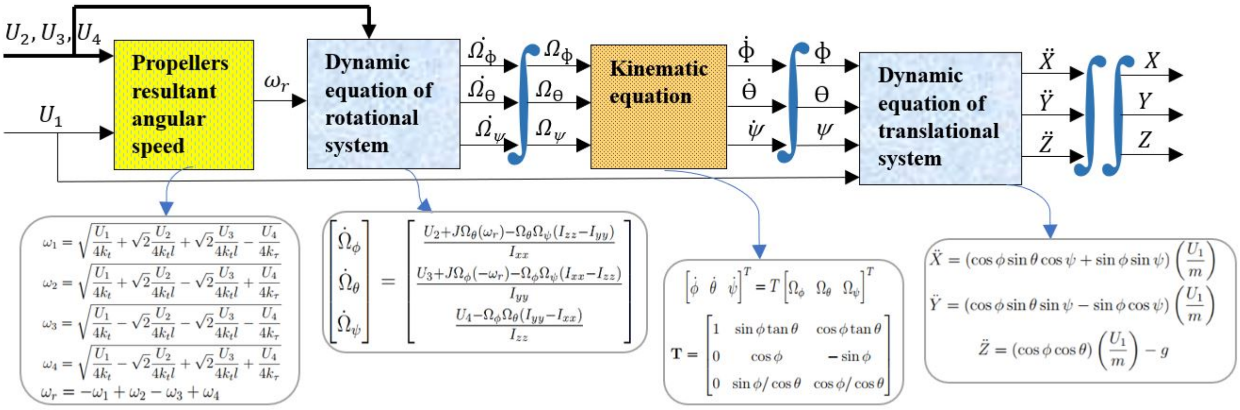


Figure 3.10: Quadcopter Mathematical Model

Parameter	Symbol	Value (Unit)
Quadcopter mass	m	0.650 kg
Inertia constants	$I_{xx} = I_{yy}$ I_{zz}	7.5×10^{-3} kg·m^2 1.3×10^{-2}
Thrust coefficient	k_T	3.13×10^{-5} N·s^2
Drag coefficient	k_τ	7.5×10^{-7} N·m·s^2
Inertia about propeller axis	J	6×10^{-5} kg·m^2
Arm length	l	0.23 m

Table 3.1: Parameters of the Quadcopter Used for Simulation [52]

Throttle (Movement along Z-axis): This is accomplished by utilizing the altitude control signal U_1, resulting in uniform motor rotation speeds.

Hovering Condition: If the thrust value ($T = mg$) is maintained, the quadcopter remains at a constant altitude above the ground. In the given scenario where $U_1 = mg = 6.3765$ N, $U_2 = U_3 = U_4 = 0$ N·m, and the initial state of the quadcopter is 5 meters above the ground in a hovering condition, the observed angular velocities are $w_1 = w_2 = w_3 = w_4 = 225.7$ rad/s. All angles remain in their zero state.

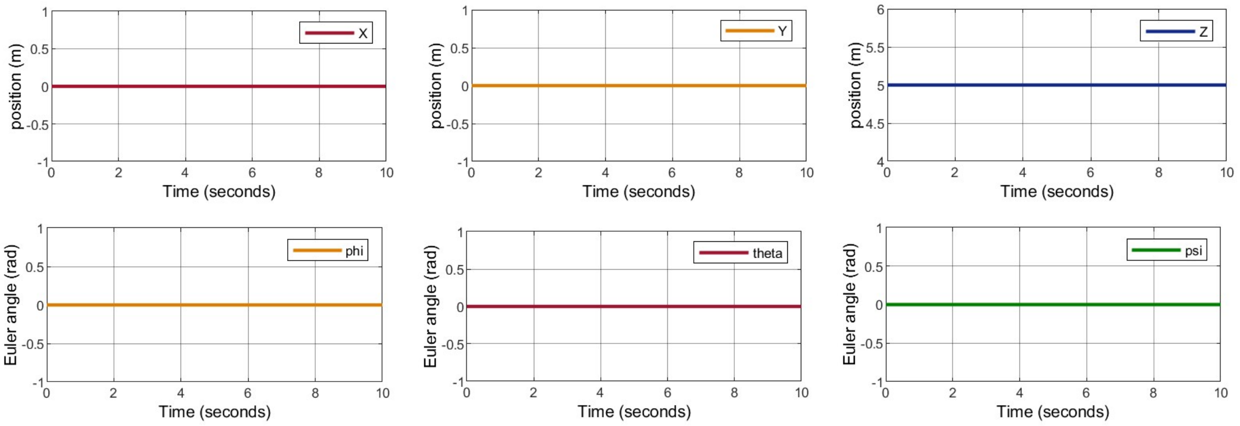


Figure 3.11: Positions and Euler Angles at Hovering Condition

Rising Action: In cases where the thrust (T) surpasses the gravitational force (mg), the vehicle undergoes an upward movement along the positive Z-axis. For instance, with $U_1 = 8$ N, $U_2 = U_3 = U_4 = 0$ N·m, and the quadcopter initially positioned at ground level (0 meters), the recorded angular velocities are $w_1 = w_2 = w_3 = w_4 = 252.8$ rad/s. All angles maintain their zero-degree orientation.

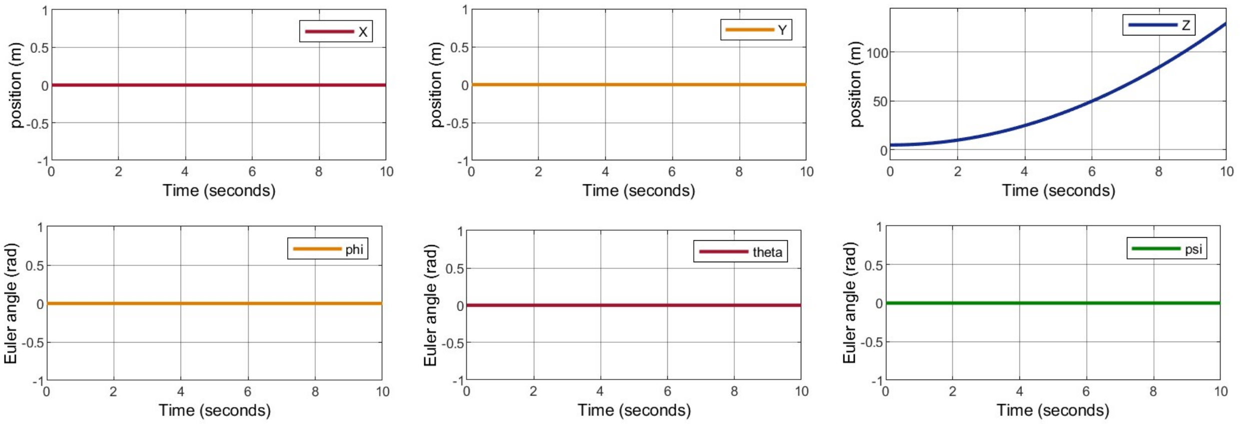


Figure 3.12: Positions and Euler Angles at Rising Action

Descending Action: When the thrust value (T) falls below the gravitational force (mg), the vehicle moves in a downward direction along the negative Z-axis. For instance, with

$U_1 = 5$ N, $U_2 = U_3 = U_4 = 0 N.m$, and the initial quadcopter altitude set at 5 meters above the ground, the recorded angular velocities are $w_1 = w_2 = w_3 = w_4 = 199.8$ rad/s. All angles remain at their zero-degree orientation.

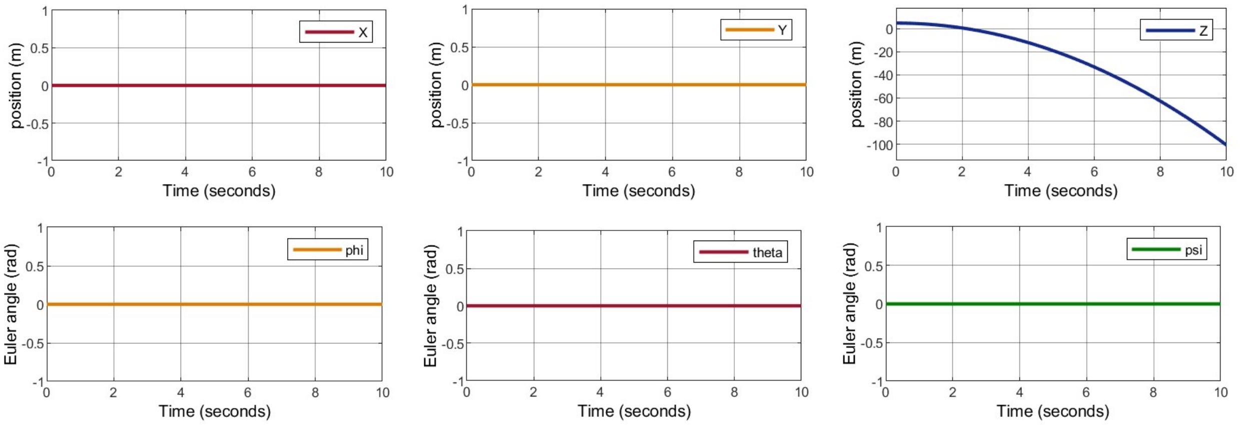


Figure 3.13: Positions and Euler Angles at Descending Action

Roll Movement: This describes the quadcopter's motion along both the Y-axis and Z-axis. When applying specific control signals, with values like $U_2 = 0.0001$ N·m, $U_1 = 6.3765$ N, and $U_3 = U_4 = 0$ N·m, and considering the initial quadcopter state at an elevation of 5 meters above the ground, it leads to changes in rotational speeds: w_1 and w_2 become 225.690 rad/s, while w_3 and w_4 reach 225.695 rad/s. These alterations result in a modification of the quadcopter's roll angle. Consequently, the quadcopter descends along the negative Z-axis and moves towards the negative Y-axis, while the X-axis remains at a standstill.

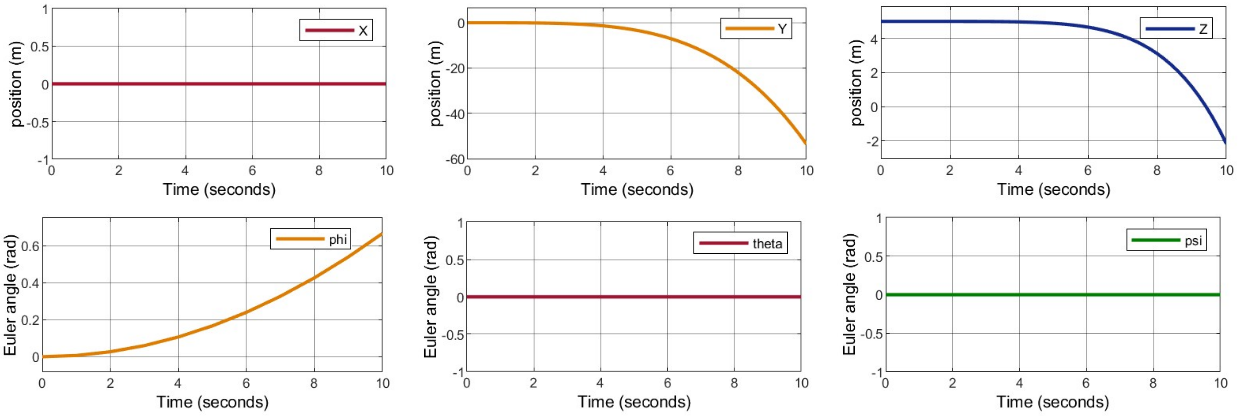


Figure 3.14: Positions and Euler Angles at Roll Movement

Pitch Movement: This corresponds to the quadcopter's motion along both the X-axis and Z-axis. When applying specific control signals, such as $U_3 = 0.0001$ N·m, $U_1 = 6.3765$ N, and $U_2 = U_4 = 0$ N·m, with the initial state of the quadcopter positioned 5 meters above the ground, it results in an increase in speed ($W_1 = W_4 = 225.69$ rad/s) and a simultaneous decrease in speed ($W_2 = W_3 = 225.665$ rad/s), leading to a change in the quadcopter's pitch angle. As a consequence, the quadcopter descends along the negative Z-axis and moves towards the positive X-axis, while the Y-axis remains unchanged, resulting in a modification of the pitch angle.

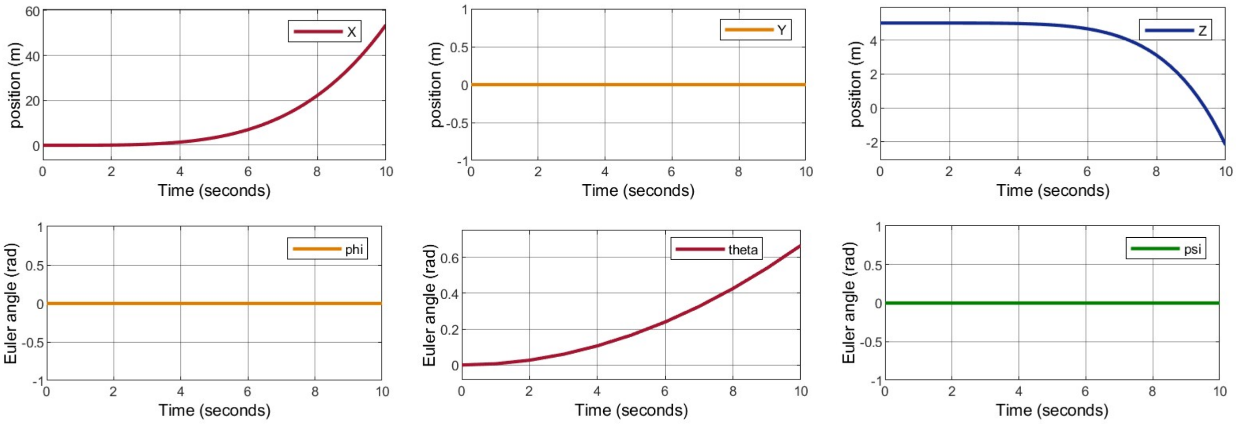


Figure 3.15: Positions and Euler Angles at Pitch Movement

Yaw Movement: In the scenario where $U_4 = 0.0001$ N·m, $U_2 = U_3 = 0$ N·m, and $U_1 = 6.3765$ N, and considering the initial state of the quadcopter positioned 5 meters above the ground, an increase in rotational speed ($W_1 = W_3 = 225.76$ rad/s) occurs alongside a simultaneous decrease in speed ($W_2 = W_4 = 225.6$ rad/s). Importantly, this change in speed doesn't affect the quadcopter's positions but causes the quadcopter to rotate in a counter clockwise direction.

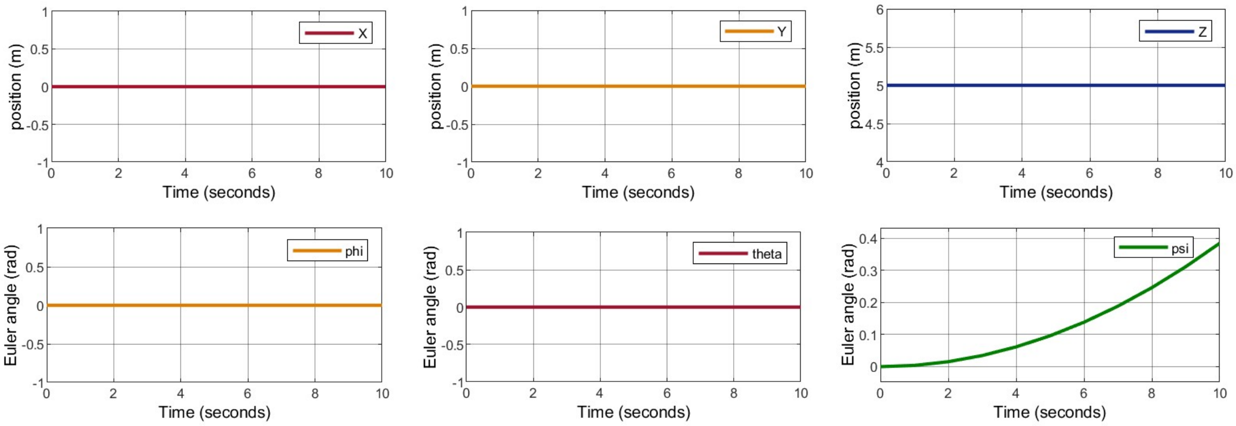


Figure 3.16: Positions and Euler Angles at Yaw Movement

Chapter Four

Controller Design

4.1 Introduction

In the field of control systems engineering, creating effective controllers is essential to achieve desired performance and uphold the stability of dynamic systems. In aerospace systems, controllers have a pivotal role in ensuring that a system responds appropriately to inputs and fulfills its intended objectives. This chapter intends to explore the design of fuzzy STSMC with PID surface. The block diagram illustrating the overall control system is depicted in the figure below.

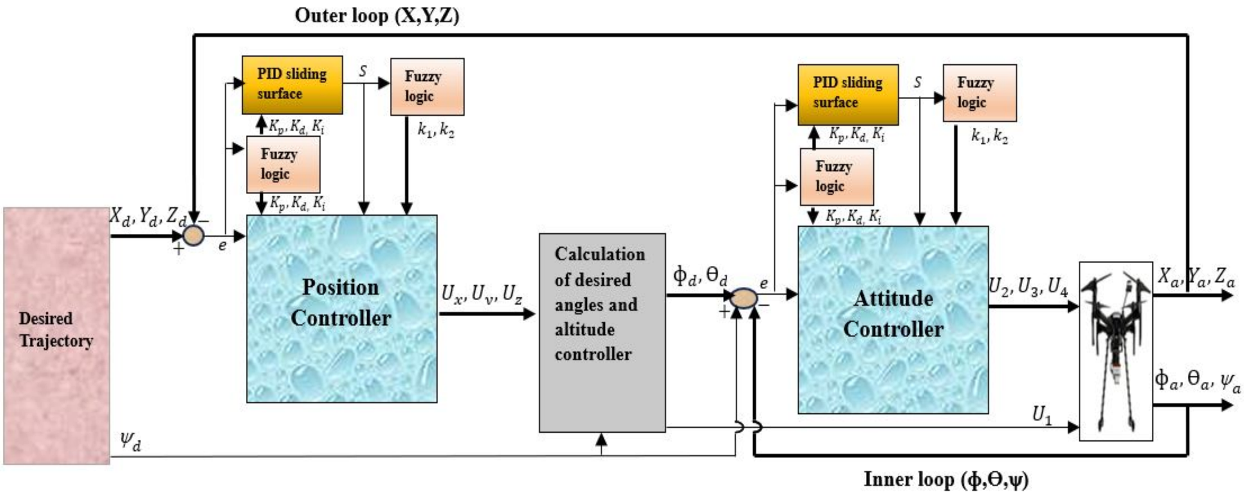


Figure 4.1: Block Diagram of Overall Control System

The process begins with a predefined desired trajectory. The system calculates the error by comparing this trajectory with the actual position and sends this error to a fuzzy logic controller. The fuzzy logic controller uses the error to determine the appropriate values for the control parameters (K_p, K_i, K_d). These parameters are then applied to the sliding surface, which computes additional values. Subsequently, this sliding surface data enters another fuzzy logic controller, where it influences parameters k_1 and k_2. These values, K_p, K_i, K_d, k_1, and k_2, are then integrated into the position controller, which, in turn, generates virtual control inputs (U_x, U_y, U_z). These virtual control inputs are further transformed into desired angles (ϕ and θ), and attitude controller. These angles serve as inputs to an inner loop controller, where they are subtracted from the actual orientation to produce an error signal. To tune the parameters of the inner loop controller, like the outer loop controller, Fuzzy logic is employed. The resulting parameters are sent to the attitude controller, which outputs U_2, U_3, and U_4 inputs. The control inputs (U_1, U_2, U_3, and U_4) are then applied to the quadcopter model, which delivers the actual position (X, Y, Z) and orientation (ϕ, θ, ψ) of the quadcopter, as shown in Figure 4.1.

4.2 Sliding Mode Control

A sliding mode controller (SMC) is a robust control technique employed for managing dynamic systems in the presence of parameter variation and disturbances. It finds particular utility in systems exhibiting nonlinear dynamics. The control strategy is created to direct the system's state variables to a predetermined sliding surface and then keep them on it. As the system state trajectory approaches this sliding surface, it is designed in a way that ensures the system state trajectory converges towards an equilibrium point, which represents the desired behavior of the controlled system. This capability allows the controller to effectively mitigate disturbances and variations in the system's dynamics. To achieve this, the controller employs discontinuous control signals based on the system's error and its derivatives [53].

Sliding mode control is a suitable algorithm for addressing the challenges posed by intricate high-order systems operating in uncertain conditions. It possesses several advantageous features, including the ability to reduce system order, reject disturbances effectively, and remain resilient in the face of parameter variations [16]. However, it suffers from a drawback known as chattering. Chattering manifests as a rapid, high-frequency switching motion between two control laws, occurring in the nearby area of the sliding surface. In an effort to keep the state trajectory on this surface, the switching functions switch at a high frequency. Regrettably, this unwanted chattering effect generates heat due to the rapid and continuous transitions between the switching functions when the control is applied to electronic or mechanical systems. This heat can potentially lead to electronic systems overheating and sustaining damage, while mechanical components can wear out prematurely and suffer eventual damage as a consequence of the abrupt shifts between the control laws.

Variable-structure control systems encompass a set of distinct but typically straightforward feedback control laws and a decision rule. This decision rule, often referred to as the switching function, determines which of the control laws should be active or "online" at any given moment based on the system's status. On the other hand, sliding mode control systems are composed of two primary modes: a "reaching mode" (or "no sliding mode") followed by a "sliding mode." Therefore, the design process for SMC involves two main aspects. First, it involves designing an appropriate switching function to achieve the desired sliding mode dynamics. Second, it involves coming up with a control strategy for the reaching mode, ensuring that a reaching condition is satisfied. In the context of sliding mode dynamics, the desired response typically aims for a rapid and stable convergence to the desired state, characterized by asymptotic convergence during the sliding mode phase. In contrast, for the reaching mode, the objective typically revolves around reaching the switching manifold within a finite time frame [54].

4.2.1 Sliding Surface

A sliding surface is a mathematical concept that clearly defines a portion within the state space of a system. Within this defined region, the control action switches and operates in a prescribed manner to achieve the desired system performance. Essentially, it serves as a representation of the intended error behavior that the controller strives to attain. The primary purpose of the sliding surface is to guide the system's state towards a predetermined trajectory or a specified reference input, facilitating the controller in achieving its control objectives [55].

When the system's states enter the sliding mode, the system's behavior is solely determined by the properties of the sliding surface. As a result, the design of the sliding surface is a critical step in achieving specific system characteristics during sliding mode operation. In accordance with the principles of the sliding mode controller, the primary focus in designing this controller lies in creating an effective sliding surface. This sliding surface is characterized by the time-varying function $S(t)$ within the n-dimensional state space $\mathbb{R}^n$, and its formulation is expressed as follows [55].

The tracking error can be defined as follows. Let $e = X_d - X_a = [e, \dot{e}, \ddot{e}, \ldots, e^{(n-1)}]^T$ represents the tracking error vector. Moreover, a time-varying surface can be defined as follows.

$$S(t) = \left(\frac{d}{dt} + c\right)^{n-1} e \tag{4.1}$$

According to Equation (4.1), c represents the coefficient governing the slope of the sliding surface, maintaining a positive constant value. It's essential to note that during the sliding mode, when state trajectories align with the sliding surface, the system's behavior remains unaffected by variations in plant parameters or disturbances. This property, known as invariance, is promising for designing feedback control systems, particularly for dynamic plants operating under uncertainty conditions [56, 16].

The sliding surface can be characterized by three distinct formulations, which are known as Proportional Derivative (PD), Proportional Integral (PI), and Proportional Integral Derivative (PID). These three groups can be represented by the following formulations [55]:

$$S_{PD} = k_d\dot{e} + k_p e \tag{4.2}$$

$$S_{\mathrm{PI}} = k_p e + k_i \int e \tag{4.3}$$

$$S_{\mathrm{PID}} = k_d\dot{e} + k_p e + k_i \int e \tag{4.4}$$

Considering the persistent nature of static error that cannot be entirely eliminated, especially when dealing with disturbance and variations in parameters, the inclusion of an integral term in the error expressions is a deliberate choice. In the context of this thesis, the PID surface is selected because it improves the tracking performance by eliminating the steady-state error.

4.2.2 Control Law

The primary objective of the control law is to guide the system trajectories onto the sliding surface and subsequently keep them within the sliding surface. There are two distinct control laws utilized in the design of sliding mode controllers. The first one is known as the equivalent control law. The equivalent control law represents the control action required to maintain the desired sliding motion on the sliding surface within the sliding mode control framework. It should be noted that this isn't the actual control input but rather reflects the cumulative effect of the discontinuous control input [57]. The second control law is referred to as the reaching law. The reaching condition signifies the circumstances under which the system's state moves toward and reaches the sliding surface. When the system follows this condition, it operates in what's known as the reaching mode.

4.2.3 SMC with PID Surface Design

The sliding mode controller is the summation of the equivalent and discontinuous controller, given as:

$$U_{\mathrm{smc}} = U_{\mathrm{eq}} + U_{\mathrm{dis}} \tag{4.5}$$

Where:

U_{smc} : Sliding mode controller, U_{eq} : Equivalent controller, U_{dis} : Discontinuous controller

- **Design for X**

For the first translational subsystem:

$$\dot{X}_1 = X_2 \tag{4.6}$$

$$\dot{X}_2 = \frac{U_x}{m} \tag{4.7}$$

Define the tracking error variable for X_1:

$$e_1 = X_1 - X_d \tag{4.8}$$

Taking the derivative of e_1:

$$\dot{e}_1 = \dot{X}_1 - \dot{X}_d = X_2 - \dot{X}_d \tag{4.9}$$

$$\ddot{e}_1 = \dot{X}_2 - \ddot{X}_d = \frac{U_x}{m} - \ddot{X}_d \tag{4.10}$$

Now, select sliding surface S_1:

$$S_1 = K_{px}e_1 + K_{ix}\int e_1\,dt + K_{dx}\dot{e}_1 \tag{4.11}$$

Substitute equation (4.9) into (4.11):

$$S_1 = K_{px}e_1 + K_{ix}\int e_1\,dt + K_{dx}(X_2 - \dot{X}_d) \tag{4.12}$$

Taking the derivative of S_1:

$$\dot{S}_1 = K_{px}\dot{e}_1 + K_{ix}e_1 + K_{dx}\ddot{e}_1 \tag{4.13}$$

Substitute equation (4.10) into (4.13):

$$\dot{S}_1 = K_{px}\dot{e}_1 + K_{ix}e_1 + K_{dx}\left(\frac{U_x}{m} - \ddot{X}_d\right) \tag{4.14}$$

The equivalent control law is found by recognizing that $\dot{S}_1$ at U_{eq} is 0. Let $\dot{S}_1 = 0$:

$$\frac{K_{dx}}{m}U_{\text{eq}(x)} = K_{dx}\ddot{X}_d - K_{px}\dot{e}_1 - K_{ix}e_1 \tag{4.15}$$

Rearranging equation (4.15):

$$U_{\text{eq}(x)} = \frac{m}{K_{dx}}\left(K_{dx}\ddot{X}_d - K_{px}\dot{e}_1 - K_{ix}e_1\right) \tag{4.16}$$

Choosing constant rate reaching:

$$\dot{S}_1 = -Q_x\text{Sign}(S_1) \tag{4.17}$$

$\dot{S}_1 = \dot{S}_1$ so equating equation (4.14) and (4.17):

$$-Q_x \mathrm{Sign}(S_1) = K_{px}\dot{e}_1 + K_{ix}e_1 + \frac{K_{dx}U_x}{m} - K_{dx}\ddot{X}_d \tag{4.18}$$

Rearranging equation (4.18), the entire control law is:

$$U_x = \frac{m}{K_{dx}}\Big(-Q_x \mathrm{Sign}(S_1) + K_{dx}\ddot{X}_d - K_{px}\dot{e}_1 - K_{ix}e_1\Big) \tag{4.19}$$

To find U_{dis}, since:

$$U_x = U_{\mathrm{eq}(x)} + U_{\mathrm{dis}(x)} \tag{4.20}$$

Where: U_x is the sliding mode controller for X, $U_{\mathrm{eq}(x)}$ is the equivalent controller for X, and $U_{\mathrm{dis}(x)}$ is the discontinuous controller for X.

Rearranging equation (4.20) and substituting equations (4.16) and (4.19):

$$\begin{aligned} U_{\mathrm{dis}(x)} = U_x - U_{\mathrm{eq}(x)} = \frac{m}{K_{dx}}\Big(-Q_x\, \mathrm{Sign}(S_1) + K_{dx}\ddot{X}_d - K_{px}\dot{e}_1 - K_{ix}e_1\Big) \\ - \frac{m}{K_{dx}}\Big(K_{dx}\ddot{X}_d - K_{px}\dot{e}_1 - K_{ix}e_1\Big) \end{aligned} \tag{4.21}$$

The discontinuous controller is:

$$U_{\mathrm{dis}(x)} = \frac{m}{K_{dx}}(-Q_x \mathrm{Sign}(S_1)) \tag{4.22}$$

- **Design for Y**

Let us consider the second translational subsystem:

$$\dot{X}_3 = X_4 \tag{4.23}$$

$$\dot{X}_4 = \frac{U_y}{m} \tag{4.24}$$

Taking error variable for X_3:

$$e_3 = X_3 - Y_d \tag{4.25}$$

Taking derivative of e_3:

$$\dot{e}_3 = \dot{X}_3 - \dot{Y}_d = X_4 - \dot{Y}_d \tag{4.26}$$

$$\ddot{e}_3 = \dot{X}_4 - \ddot{Y}_d = \frac{U_y}{m} - \ddot{Y}_d \tag{4.27}$$

Now select sliding surface S_3:

$$S_3 = K_{py}e_3 + K_{iy}\int e_3\, dt + K_{dy}\dot{e}_3 \tag{4.28}$$

Substitute equation (4.26) into (4.28):

$$S_3 = K_{py}e_3 + K_{iy}\int e_3\,dt + K_{dy}\left(X_4 - \dot{Y}_d\right) \tag{4.29}$$

Taking derivative of S_3:

$$\dot{S}_3 = K_{py}\dot{e}_3 + K_{iy}e_3 + K_{dy}\ddot{e}_3 \tag{4.30}$$

Substitute equation (4.27) into (4.30):

$$\dot{S}_3 = K_{py}\dot{e}_3 + K_{iy}e_3 + K_{dy}\left(\frac{U_y}{m} - \ddot{Y}_d\right) \tag{4.31}$$

The equivalent control law is found by recognizing that: $\dot{S}_3$ at U_{eq} is 0. Let $\dot{S}_3 = 0$:

$$K_{dy}\frac{U_{eq(y)}}{m} = K_{dy}\ddot{Y}_d - K_{py}\dot{e}_3 - K_{iy}e_3 \tag{4.32}$$

Rearranging equation (4.32):

$$U_{eq(y)} = \frac{m}{K_{dy}}\left(K_{dy}\ddot{Y}_d - K_{py}\dot{e}_3 - K_{iy}e_3\right) \tag{4.33}$$

Choosing constant rate reaching:

$$\dot{S}_3 = -Q_y\text{Sign}(S_3) : \tag{4.34}$$

$\dot{S}_3 = \dot{S}_3$ so equating equation (4.31) and (4.34) :

$$-Q_y\text{Sign}(S_3) = K_{py}\dot{e}_3 + K_{iy}e_3 + \frac{K_{dy}U_y}{m} - K_{dy}\ddot{Y}_d \tag{4.35}$$

By rearranging equation (4.35), the entire control law is:

$$U_y = \frac{m}{K_{dy}}\left(-Q_y\text{Sign}(S_3) + K_{dy}\ddot{Y}_d - K_{py}\dot{e}_3 - K_{iy}e_3\right) \tag{4.36}$$

To find U_{dis}, since:

$$U_y = U_{eq(y)} + U_{dis(y)} \tag{4.37}$$

Where: U_y is the sliding mode controller for Y, $U_{\text{eq}(y)}$ is the equivalent controller for Y, and $U_{\text{dis}(y)}$ is the discontinuous controller for Y.

Rearranging equation (4.37) and substituting equations (4.33) and (4.36):

$$\begin{aligned} U_{\text{dis}(y)} = U_y - U_{\text{eq}(y)} = \frac{m}{K_{dy}}&\left(-Q_y\,\text{Sign}(S_3) + K_{dy}\ddot{Y}_d - K_{py}\dot{e}_3 - K_{iy}e_3\right) \\ &- \frac{m}{K_{dy}}\left(K_{dy}\ddot{Y}_d - K_{py}\dot{e}_3 - K_{iy}e_3\right) \end{aligned} \tag{4.38}$$

The discontinuous controller is:

$$U_{dis(y)} = \frac{m}{K_{dy}}(-Q_y \text{Sign}(S_3)) \tag{4.39}$$

- **Design for Z**

Let us consider the third translational subsystem:

$$\dot{X}_5 = X_6 \tag{4.40}$$

$$\dot{X}_6 = \frac{U_z}{m} \tag{4.41}$$

Taking error variable for X_5:

$$e_5 = X_5 - Z_d \tag{4.42}$$

Taking derivative of e_5:

$$\dot{e}_5 = \dot{X}_5 - \dot{Z}_d = X_6 - \dot{Z}_d \tag{4.43}$$

$$\ddot{e}_5 = \dot{X}_6 - \ddot{Z}_d = \frac{U_z}{m} - \ddot{Z}_d \tag{4.44}$$

Now select sliding surface S_5:

$$S_5 = K_{pz}e_5 + K_{iz}\int e_5\, dt + K_{dz}\dot{e}_5 \tag{4.45}$$

Substitute equation (4.43) into (4.45):

$$S_5 = K_{pz}e_5 + K_{iz}\int e_5\, dt + K_{dz}\left(X_6 - \dot{Z}_d\right) \tag{4.46}$$

Taking derivative of S_5:

$$\dot{S}_5 = K_{pz}\dot{e}_5 + K_{iz}e_5 + K_{dz}\ddot{e}_5 \tag{4.47}$$

Substitute equation (4.47) into (4.44):

$$\dot{S}_5 = K_{pz}\dot{e}_5 + K_{iz}e_5 + K_{dz}\left(\frac{U_z}{m} - \ddot{Z}_d\right) \tag{4.48}$$

The equivalent control law is found by recognizing that: $\dot{S}_5$ at $U_{eq(z)}$ is 0. Let $\dot{S}_5 = 0$:

$$K_{dz}\frac{U_{eq(z)}}{m} = K_{dz}\ddot{Z}_d - K_{pz}\dot{e}_5 - K_{iz}e_5 \tag{4.49}$$

Rearranging equation (4.49):

$$U_{eq(z)} = \frac{m}{K_{dz}}\left(K_{dz}\ddot{Z}_d - K_{pz}\dot{e}_5 - K_{iz}e_5\right) \tag{4.50}$$

Choosing constant rate reaching:

$$\dot{S}_5 = -Q_z \text{Sign}(S_5) \tag{4.51}$$

$\dot{S}_5 = \dot{S}_5$ so equating equation (4.48) and (4.51)

$$-Q_z \text{Sign}(S_5) = K_{pz}\dot{e}_5 + K_{iz}e_5 + \frac{K_{dz}U_z}{m} - K_{dz}\ddot{Z}_d \tag{4.52}$$

By rearranging equation (4.52), the entire control law is:

$$U_z = \frac{m}{K_{dz}}\Big(-Q_z \text{Sign}(S_5) + K_{dz}\ddot{Z}_d - K_{pz}\dot{e}_5 - K_{iz}e_5\Big) \tag{4.53}$$

To find $U_{dis(z)}$, since:

$$U_z = U_{eq(z)} + U_{dis(z)} \tag{4.54}$$

Where: U_z is the sliding mode controller for Z, $U_{\text{eq}(z)}$ is the equivalent controller for Z, and $U_{\text{dis}(z)}$ is the discontinuous controller for Z.

Rearranging equation (4.54) and substituting equation (4.50) and (4.53)

$$\begin{aligned} U_{\text{dis}(z)} = U_z - U_{\text{eq}(z)} = \frac{m}{K_{dz}}\Big(-Q_z\,\text{Sign}(S_5) + K_{dz}\ddot{Z}_d - K_{pz}\dot{e}_5 - K_{iz}e_5\Big) \\ - \frac{m}{K_{dz}}\Big(K_{dz}\ddot{Z}_d - K_{pz}\dot{e}_5 - K_{iz}e_5\Big) \end{aligned} \tag{4.55}$$

The discontinuous controller is:

$$U_{dis(z)} = \frac{m}{K_{dz}}(-Q_z \text{Sign}(S_5)) \tag{4.56}$$

- **Design for Phi**

Let us consider the first rotational subsystem:

$$\dot{X}_7 = X_8 \tag{4.57}$$

$$\dot{X}_8 = b_1U_2 + a_1X_{10}X_{12} + a_2X_{10} \tag{4.58}$$

Taking error variable for X_7:

$$e_7 = X_7 - \phi_d \tag{4.59}$$

Taking derivative of e_7:

$$\dot{e}_7 = \dot{X}_7 - \dot{\phi}_d = X_8 - \dot{\phi}_d \tag{4.60}$$

$$\ddot{e}_7 = \dot{X}_8 - \ddot{\phi}_d = b_1U_2 + a_1X_{10}X_{12} + a_2X_{10} - \ddot{\phi}_d \tag{4.61}$$

Now select sliding surface S_7:

$$S_7 = K_{p\phi} e_7 + K_{i\phi} \int e_7 \, dt + K_{d\phi} \dot{e}_7 \tag{4.62}$$

Substitute equation (4.60) into (4.62):

$$S_7 = K_{p\phi} e_7 + K_{i\phi} \int e_7 \, dt + K_{d\phi} \left(X_8 - \dot{\phi}_d \right) \tag{4.63}$$

Taking derivative of S_7:

$$\dot{S}_7 = K_{p\phi} \dot{e}_7 + K_{i\phi} e_7 + K_{d\phi} \ddot{e}_7 \tag{4.64}$$

Substitute equation (4.61) into (4.64):

$$\dot{S}_7 = K_{p\phi} \dot{e}_7 + K_{i\phi} e_7 + K_{d\phi} \left(b_1 U_2 + a_1 X_{10} X_{12} + a_2 X_{10} - \ddot{\phi}_d \right) \tag{4.65}$$

The equivalent control law is found by recognizing that: $\dot{S}_7$ at $U_{eq(2)}$ is 0. Let $\dot{S}_7 = 0$:

$$K_{d\phi} b_1 U_{eq(2)} = K_{d\phi} \ddot{\phi}_d - K_{p\phi} \dot{e}_7 - K_{i\phi} e_7 - K_{d\phi} a_1 X_{10} X_{12} - K_{d\phi} a_2 X_{10} \tag{4.66}$$

Rearranging equation (4.66):

$$U_{eq(2)} = \frac{1}{K_{d\phi} b_1} \left(K_{d\phi} \ddot{\phi}_d - K_{p\phi} \dot{e}_7 - K_{i\phi} e_7 - K_{d\phi} a_1 X_{10} X_{12} - K_{d\phi} a_2 X_{10} \right) \tag{4.67}$$

Choosing constant rate reaching:

$$\dot{S}_7 = -Q_\phi \, \mathrm{Sign}(S_7) \tag{4.68}$$

$\dot{S}_7 = \dot{S}_7$ so equating equation (4.65) and (4.68)

$$-Q_\phi \, \mathrm{Sign}(S_7) = K_{p\phi} \dot{e}_7 + K_{i\phi} e_7 + K_{d\phi} \left(b_1 U_2 + a_1 X_{10} X_{12} + a_2 X_{10} - \ddot{\phi}_d \right) \tag{4.69}$$

Rearranging equation (4.69), the entire control law is:

$$U_2 = \frac{1}{K_{d\phi} b_1} \left(-Q_\phi \, \mathrm{Sign}(S_7) + K_{d\phi} \ddot{\phi}_d - K_{p\phi} \dot{e}_7 - K_{i\phi} e_7 - K_{d\phi} a_1 X_{10} X_{12} - a_2 X_{10} K_{d\phi} \right) \tag{4.70}$$

To find $U_{dis(2)}$, since:

$$U_2 = U_{eq(2)} + U_{dis(2)} \tag{4.71}$$

Where: U_2 is sliding mode controller for ϕ, $U_{eq(2)}$ is equivalent controller for ϕ, and $U_{dis(2)}$ is discontinuous controller for ϕ.

Rearranging equation (4.71) and substituting equation (4.67) and (4.70):

$$
\begin{aligned}
U_{\text{dis}(2)} &= U_2 - U_{\text{eq}(2)} \\
&= \frac{1}{K_{d\phi}b_1}\left(-Q_\phi \operatorname{Sign}(S_7) + K_{d\phi}\ddot{\phi}_d - K_{p\phi}\dot{e}_7 - K_{i\phi}e_7 - K_{d\phi}a_1X_{10}X_{12} - a_2X_{10}K_{d\phi}\right) \\
&\qquad - \frac{1}{K_{d\phi}b_1}\left(K_{d\phi}\ddot{\phi}_d - K_{p\phi}\dot{e}_7 - K_{i\phi}e_7 - K_{d\phi}a_1X_{10}X_{12} - K_{d\phi}a_2X_{10}\right)
\end{aligned}
\tag{4.72}
$$

The discontinuous controller is :

$$U_{\text{dis}(2)} = \frac{1}{K_{d\phi}b_1}\left(-Q_\phi \operatorname{Sign}(S_7)\right) \tag{4.73}$$

- **Design for Theta**

Let us consider the second rotational subsystem:

$$\dot{X}_9 = X_{10} \tag{4.74}$$

$$\dot{X}_{10} = b_2U_3 + a_3X_8X_{12} - a_4X_8 \tag{4.75}$$

Taking error variable for X_9:

$$e_9 = X_9 - \theta_d \tag{4.76}$$

Taking derivative of e_9:

$$\dot{e}_9 = \dot{X}_9 - \dot{\theta}_d = X_{10} - \dot{\theta}_d \tag{4.77}$$

$$\ddot{e}_9 = \dot{X}_{10} - \ddot{\theta}_d = b_2U_3 + a_3X_8X_{12} - a_4X_8 - \ddot{\theta}_d \tag{4.78}$$

Now select sliding surface S_9:

$$S_9 = K_{p\theta}e_9 + K_{i\theta}\int e_9\,dt + K_{d\theta}\dot{e}_9 \tag{4.79}$$

Substitute equation (4.77) into (4.79):

$$S_9 = K_{p\theta}e_9 + K_{i\theta}\int e_9\,dt + K_{d\theta}\left(X_{10} - \dot{\theta}_d\right) \tag{4.80}$$

Taking derivative of S_9:

$$\dot{S}_9 = K_{p\theta}\dot{e}_9 + K_{i\theta}e_9 + K_{d\theta}\ddot{e}_9 \tag{4.81}$$

Substitute equation (4.78) into (4.81):

$$\dot{S}_9 = K_{p\theta}\dot{e}_9 + K_{i\theta}e_9 + K_{d\theta}\left(b_2U_3 + a_3X_8X_{12} - a_4X_8 - \ddot{\theta}_d\right) \tag{4.82}$$

The equivalent control law is found by recognizing that: $\dot{S}_9$ at U_{eq} is 0. Let $\dot{S}_9 = 0$:

$$K_{d\theta}b_2U_{\text{eq}(3)} = K_{d\theta}(\ddot{\theta_d}) - K_{p\theta}(\dot{e_9}) - K_{i\theta}e_9 - K_{d\theta}a_3X_8X_{12} + K_{d\theta}a_4X_8 \tag{4.83}$$

Rearranging equation (4.83):

$$U_{\mathrm{eq}(3)} = \frac{1}{K_{d\theta} b_2} \left(K_{d\theta} \ddot{\theta}_d - K_{p\theta} \dot{e}_9 - K_{i\theta} e_9 - K_{d\theta} a_3 X_8 X_{12} + K_{d\theta} a_4 X_8 \right) \tag{4.84}$$

Choosing constant rate reaching:

$$\dot{S}_9 = -Q_\theta \mathrm{Sign}(S_9) \tag{4.85}$$

$\dot{S}_9 = \dot{S}_9$, so equating equation (4.82) and (4.85):

$$-Q_\theta \mathrm{Sign}(S_9) = K_{p\theta} \dot{e}_9 + K_{i\theta} e_9 + K_{d\theta} \left(b_2 U_3 + a_3 X_8 X_{12} - a_4 X_8 - \ddot{\theta}_d \right) \tag{4.86}$$

Rearranging equation (4.86), the entire control law is:

$$U_3 = \frac{1}{K_{d\theta} b_2} \left(-Q_\theta \mathrm{Sign}(S_9) + K_{d\theta} \ddot{\theta}_d - K_{p\theta} \dot{e}_9 - K_{i\theta} e_9 - K_{d\theta} a_3 X_8 X_{12} + a_4 X_8 K_{d\phi} \right) \tag{4.87}$$

To find U_{dis}, since:

$$U_3 = U_{\mathrm{eq}(3)} + U_{\mathrm{dis}(3)} \tag{4.88}$$

Where: U_3 is sliding mode controller for θ, $U_{\mathrm{eq}(z)}$ is equivalent controller for θ, and $U_{\mathrm{dis}(z)}$ is discontinuous controller for θ.

Rearranging equation (4.88) and substituting equation (4.84) and (4.87):

$$\begin{aligned} U_{\mathrm{dis}(3)} &= U_3 - U_{\mathrm{eq}(3)} \\ &= \frac{1}{K_{d\theta} b_2} \left(-Q_\theta \, \mathrm{Sign}(S_9) + K_{d\theta} (\ddot{\theta}_d) - K_{p\theta} (\dot{e}_9) - K_{i\theta} e_9 - K_{d\theta} a_3 X_8 X_{12} + a_4 X_8 K_{d\theta} \right) \\ &\qquad - \frac{1}{K_{d\theta} b_2} \left(K_{d\theta} (\ddot{\theta}_d) - K_{p\theta} (\dot{e}_9) - K_{i\theta} e_9 - K_{d\theta} a_3 X_8 X_{12} + K_{d\theta} a_4 X_8 \right) \end{aligned} \tag{4.89}$$

The discontinuous controller is:

$$U_{\mathrm{dis}(3)} = \frac{1}{K_{d\theta} b_2} (-Q_\theta \mathrm{Sign}(S_9)) \tag{4.90}$$

- **Design for Psi**

Let us consider the third rotational subsystem:

$$\dot{X}_{11} = X_{12} \tag{4.91}$$

$$\dot{X}_{12} = b_3 U_4 + a_5 X_8 X_{10} \tag{4.92}$$

Taking error variable for X_{11}:

$$e_{11} = X_{11} - \psi_d \tag{4.93}$$

Taking derivative of e_{11}:

$$\dot{e}_{11} = \dot{X}_{11} - \dot{\psi}_d = X_{12} - \dot{\psi}_d \tag{4.94}$$

$$\ddot{e}_{11} = \dot{X}_{12} - \ddot{\psi}_d = b_3 U_4 + a_5 X_8 X_{10} - \ddot{\psi}_d \tag{4.95}$$

Now select sliding surface S_{11}:

$$S_{11} = K_{p\psi} e_{11} + K_{i\psi} \int e_{11}\, dt + K_{d\psi} \dot{e}_{11} \tag{4.96}$$

Substitute equation (4.94) into (4.96):

$$S_{11} = K_{p\psi} e_{11} + K_{i\psi} \int e_{11}\, dt + K_{d\psi} (X_{12} - \dot{\psi}_d) \tag{4.97}$$

Taking derivative of S_{11}:

$$\dot{S}_{11} = K_{p\psi} \dot{e}_{11} + K_{i\psi} e_{11} + K_{d\psi} \ddot{e}_{11} \tag{4.98}$$

Substitute equation (4.95) into (4.98):

$$\dot{S}_{11} = K_{p\psi} \dot{e}_{11} + K_{i\psi} e_{11} + K_{d\psi} (b_3 U_4 + a_5 X_8 X_{10} - \ddot{\psi}_d) \tag{4.99}$$

The equivalent control law is found by recognizing that: $\dot{S}_{11}$ at U_{eq} is 0. Let $\dot{S}_{11} = 0$:

$$K_{d\psi} b_3 U_{\text{eq}(4)} = K_{d\psi} (\ddot{\psi}_d) - K_{p\psi} (\dot{e_{11}}) - K_{i\psi} e_{11} - K_{d\psi} a_5 X_8 X_{10} \tag{4.100}$$

Rearranging equation (4.100):

$$U_{\text{eq}(4)} = \frac{1}{K_{d\psi} b_3} \Big(K_{d\psi} \ddot{\psi}_d - K_{p\psi} \dot{e}_{11} - K_{i\psi} e_{11} - K_{d\psi} a_5 X_8 X_{10} \Big) \tag{4.101}$$

Choosing constant rate reaching:

$$\dot{S}_{11} = -Q_{\psi} \text{Sign}(S_{11}) \tag{4.102}$$

$\dot{S}_{11} = \dot{S}_{11}$, so equating equations (4.99) and (4.102)

$$-Q_{\psi}\, \text{Sign}(S_{11}) = K_{p\psi} \dot{e}_{11} + K_{i\psi} e_{11} + K_{d\psi} (b_3 U_4 + a_5 X_8 X_{10} - \ddot{\psi}_d) \tag{4.103}$$

Rearranging equation (4.103), the entire control law is:

$$U_4 = \frac{1}{K_{d\psi} b_3} \Big(-Q_{\psi} \text{Sign}(S_{11}) + K_{d\psi} \ddot{\psi}_d - K_{p\psi} \dot{e}_{11} - K_{i\psi} e_{11} - K_{d\psi} a_5 X_8 X_{10} \Big) \tag{4.104}$$

To find U_{dis}, since:

$$U_4 = U_{\mathrm{eq}(4)} + U_{\mathrm{dis}(4)} \tag{4.105}$$

Where: U_4 is the sliding mode controller for ψ, $U_{\mathrm{eq}(z)}$ is the equivalent controller for ψ, and $U_{\mathrm{dis}(z)}$ is the discontinuous controller for ψ.

Rearranging equation (4.105) and substituting equation (4.101) and (4.104):

$$\begin{aligned} U_{\mathrm{dis}(4)} &= U_4 - U_{\mathrm{eq}(4)} \\ &= \frac{1}{K_{d\psi} b_3}\Big(-Q_\psi \operatorname{Sign}(S_{11}) + K_{d\psi}\ddot{\psi}_d - K_{p\psi}\dot{e}_{11} - K_{i\psi}e_{11} - K_{d\psi}a_5 X_8 X_{10}\Big) \\ &\quad - \frac{1}{K_{d\psi} b_3}\Big(K_{d\psi}\ddot{\psi}_d - K_{p\psi}\dot{e}_{11} - K_{i\psi}e_{11} - K_{d\psi}a_5 X_8 X_{10}\Big) \end{aligned} \tag{4.106}$$

The discontinuous controller is:

$$U_{\mathrm{dis}(4)} = \frac{1}{K_{d\psi} b_3}(-Q_\psi \operatorname{Sign}(S_{11})) \tag{4.107}$$

The stability analysis is in Appendix F.1.

4.3 Super Twisting Sliding Mode Controller

The Super Twisting Sliding Mode Controller (STSMC) is a specific approach within sliding mode control used for controlling dynamic systems. It elevates the conventional sliding mode control by introducing a super twisting algorithm. This algorithm enables rapid and finite-time convergence to the sliding surface, even in the presence of parameter variations and disturbances. It accomplishes this by utilizing a continuous control law that minimizes chattering. As in the case with any control method, it comes with its own set of advantages and disadvantages [58].

Advantages:

1. **Robustness:** STSMC is robust in dealing with disturbances and parameter variations. It can effectively handle variations in system parameters and disturbances.

2. **Fast Response:** The super twisting algorithm enables fast convergence towards the sliding surface. This leads to a quick response and reduced settling time, which is beneficial in dynamic processes.

3. **Chattering Reduction:** It reduces chattering by smoothing control signal transitions near the sliding surface.

Disadvantages:

1. **High Control Effort:** The control effort in STSMC is determined by the parameters chosen for the controller. If the parameters are set too aggressively, the control action can become excessively large, causing the system to overshoot the desired trajectory.

2. **Tuning Complexity:** Tuning the controller parameters is a challenging task. Selecting appropriate parameters requires expertise and understanding of the controlled system dynamics.

In the context of the super twisting algorithm, an additional nonlinear term is introduced into the sliding mode control law to improve control performance. This extra component is referred to as the 'super twisting term.' The primary control objective is to guide the system trajectory to attain the sliding manifold condition, represented as $S = \dot{S} = 0$, within a finite time. The super twisting algorithm is formulated as follows:

$$\dot{S} = -k_1 \left|S\right|^{\frac{1}{2}} \operatorname{sign}(S) - k_2 \int \operatorname{sign}(S)\, dt$$

Super twisting algorithm has two terms. The first term is primarily responsible for achieving finite-time convergence. It is designed to guide the system's state onto the sliding surface within a finite time. This term introduces a continuous function that is proportional to the rate of change of the sliding variable. The second term contributes to chattering reduction and enhances the control system's performance. It introduces a continuous function that acts to attenuate high-frequency oscillations in the control signal. This function helps smooth the control action further and reduces the amplification of high-frequency components that can lead to chattering. Both terms in the super twisting algorithm work together to mitigate chattering and improve the performance of sliding mode control systems. The continuous nature of these terms aids in achieving smoother convergence, reducing oscillations, and enhancing control accuracy [59].

Through computer simulations and careful selection of suitable constants, denoted as k_1 and k_2, the controller guides the sliding variable S and its derivative to converge to zero within a finite time frame. The design of the equivalent control within the controller follows a similar procedure as that used for the conventional sliding mode controller.

4.3.1 STSMC with PID Surface Design

The Super twisting sliding mode controller is the summation of equivalent and continuous controller, given as:

$$U_{\mathrm{STsmc}} = U_{\mathrm{eq}} + U_{\mathrm{ST}} \tag{4.108}$$

Where: U_{STsmc}: Super twisting sliding mode controller, U_{eq}: Equivalent controller, and U_{ST}: Continuous controller

- **Design for X**

The equivalent controller $U_{\mathrm{eq}(x)}$ is the same as equation (4.16), which is done in the design of SMC.

Reaching law: Based on STA, the closed-loop sliding dynamics for X, set as follows:

$$\dot{S}_1 = -k_1|S_1|^{\frac{1}{2}} \operatorname{sign}(S_1) - k_2 \int \operatorname{sign}(S_1)\, dt \tag{4.109}$$

Where: k_1 and k_2 are positive constant numbers.

$\dot{S}_1 = \dot{S}_1$, So equating equation (4.14) and (4.109) yields:

$$-k_1|S_1|^{\frac{1}{2}} \operatorname{sign}(S_1) - k_2 \int \operatorname{sign}(S_1)\, dt = K_{px}\dot{e}_1 + K_{ix}e_1 + \frac{K_{dx}U_x}{m} - K_{dx}\ddot{X}_d \tag{4.110}$$

Rearranging equation (4.110), the entire control law is:

$$U_x = \frac{m}{K_{dx}} \left(-k_1|S_1|^{\frac{1}{2}} \operatorname{sign}(S_1) - k_2 \int \operatorname{sign}(S_1)\, dt + K_{dx}\ddot{X}_d - K_{px}\dot{e}_1 - K_{ix}e_1 \right) \tag{4.111}$$

To find U_{ST}, since:

$$U_x = U_{\mathrm{eq}(x)} + U_{\mathrm{ST}(x)} \tag{4.112}$$

Where: U_x: Super twisting sliding mode controller for X, $U_{\mathrm{eq}(x)}$: Equivalent controller for X, $U_{\mathrm{ST}(x)}$: Continuous controller for X

Rearranging equation (4.112) and substituting equation (4.16) and (4.111) yields:

$$\begin{aligned} U_{ST}(x) &= U_x - U_{eq}(x) \\ &= \frac{m}{K_{d_x}} \left(-k_1|S_1|^{1/2}\operatorname{sign}(S_1) - k_2 \int \operatorname{sign}(S_1)\, dt \right) + \frac{m}{K_{d_x}} \left(K_{d_x}\ddot{X}_d - K_{p_x}\dot{e}_1 - K_{i_x}e_1 \right) \\ &- \frac{m}{K_{d_x}} \left(K_{d_x}\ddot{X}_d - K_{p_x}\dot{e}_1 - K_{i_x}e_1 \right) \end{aligned} \tag{4.113}$$

The continuous controller is:

$$U_{\mathrm{ST}(x)} = \frac{m}{K_{dx}} \left(-k_1|S_1|^{\frac{1}{2}} \operatorname{sign}(S_1) - k_2 \int \operatorname{sign}(S_1)\, dt \right) \tag{4.114}$$

- **Design for Y**

The equivalent controller $U_{\text{eq}(y)}$ is the same as equation (4.33), which is done in the design of SMC.

Reaching law: Based on STA, the closed-loop sliding dynamics for Y, set as follows:

$$\dot{S}_3 = -k_1|S_3|^{1/2}\text{sign}(S_3) - k_2 \int \text{sign}(S_3)\, dt \tag{4.115}$$

Where: k_1 and k_2 are positive constant numbers.

$\dot{S}_3 = \dot{S}_3$, so equating equation (4.31) and (4.115) yields:

$$-k_1|S_3|^{1/2}\text{sign}(S_3) - k_2 \int \text{sign}(S_3)\, dt = K_{py}\dot{e}_3 + K_{iy}e_3 + \frac{K_{dy}U_y}{m} - K_{dy}\ddot{y}_d \tag{4.116}$$

Rearranging equation (4.116), the entire control law is:

$$U_y = \frac{m}{K_d y}\left(-k_1|S_3|^{1/2}\text{sign}(S_3) - k_2 \int \text{sign}(S_3)\, dt + K_{dy}\ddot{Y}_d - K_{py}\dot{e}_3 - K_{iy}e_3\right) \tag{4.117}$$

To find $U_{ST(y)}$, since:

$$U_y = U_{\text{eq}(y)} + U_{ST(y)} \tag{4.118}$$

Where: U_y is the Super twisting sliding mode controller for Y, $U_{\text{eq}(y)}$ is the equivalent controller for Y, and $U_{ST(y)}$ is the continuous controller for Y.

Rearranging equation (4.118) and substituting equations (4.33) and (4.117) yields:

$$\begin{aligned} U_{ST}(y) &= U_y - U_{\text{eq}(y)} \\ &= \frac{m}{K_{dy}}\left(-k_1|S_3|^{1/2}\text{sign}(S_3) - k_2 \int \text{sign}(S_3)\, dt + K_{dy}\ddot{Y}_d - K_{py}\dot{e}_3 - K_{iy}e_3\right) \\ &\quad - \frac{m}{K_{dy}}\left(K_{dy}\ddot{Y}_d - K_{py}\dot{e}_3 - K_{iy}e_3\right) \end{aligned} \tag{4.119}$$

The continuous controller is:

$$U_{ST}(y) = \frac{m}{K_{dy}}\left(-k_1|S_3|^{1/2}\text{sign}(S_3) - k_2 \int \text{sign}(S_3)\, dt\right) \tag{4.120}$$

- **Design for Z**

The equivalent controller $U_{\text{eq}(z)}$ is the same as equation (4.50), which is done in the design of SMC.

Reaching law: Based on STA, the closed-loop sliding dynamics for Z is set as follows:

$$\dot{S}_5 = -k_1|S_5|^{1/2}\text{sign}(S_5) - k_2\int \text{sign}(S_5)\,dt \tag{4.121}$$

Where: k_1 and k_2 are positive constant numbers.

$\dot{S}_5 = \dot{S}_5$, so equating equation (4.48) and (4.121) yields:

$$-k_1|S_5|^{1/2}\text{sign}(S_5) - k_2\int \text{sign}(S_5)\,dt = K_{pz}\dot{e}_5 + K_{iz}e_5 + \frac{K_{dz}U_z}{m} - K_{dz}\ddot{Z}_d \tag{4.122}$$

Rearranging equation (4.122), the entire control law is:

$$U_z = \frac{m}{K_{dz}}\left(-k_1|S_5|^{1/2}\text{sign}(S_5) - k_2\int \text{sign}(S_5)\,dt + K_{dz}\ddot{Z}_d - K_{pz}\dot{e}_5 - K_{iz}e_5\right) \tag{4.123}$$

To find $U_{ST(z)}$, since:

$$U_z = U_{\text{eq}(z)} + U_{ST(z)} \tag{4.124}$$

Where: U_z is the Super twisting sliding mode controller for Z, $U_{\text{eq}(z)}$ is the equivalent controller for Z, and $U_{ST(z)}$ is the continuous controller for Z.

Rearranging equation (4.124) and substituting equations (4.50) and (4.123) yields:

$$\begin{aligned}U_{ST(z)} &= U_z - U_{\text{eq}(z)}\\ &= \frac{m}{K_{dz}}\left(-k_1|S_5|^{1/2}\text{sign}(S_5) - k_2\int \text{sign}(S_5)\,dt + K_{dz}\ddot{Z}_d - K_{pz}\dot{e}_5 - K_{iz}e_5\right)\\ &\quad - \frac{m}{K_{dz}}\left(K_{dz}\ddot{Z}_d - K_{pz}\dot{e}_5 - K_{iz}e_5\right)\end{aligned} \tag{4.125}$$

The continuous controller is:

$$U_{ST(z)} = \frac{m}{K_{dz}}\left(-k_1|S_5|^{1/2}\text{sign}(S_5) - k_2\int \text{sign}(S_5)\,dt\right) \tag{4.126}$$

- **Design for Phi**

The equivalent controller $U_{\text{eq}(\phi)}$ is the same as equation (4.67), which is done in the design of SMC.

Reaching law: Based on STA, the closed-loop sliding dynamics for ϕ is set as follows:

$$\dot{S}_7 = -k_1|S_7|^{1/2}\text{sign}(S_7) - k_2\int \text{sign}(S_7)\,dt \tag{4.127}$$

Where: k_1 and k_2 are positive constant numbers.

$\dot{S}_7 = \dot{S}_7$, So equating equation (4.65) and (4.127):

$$-k_1|S_7|^{1/2}\text{sign}(S_7) - k_2\int \text{sign}(S_7)\,dt = K_{p\phi}\dot{e}_7 + K_{i\phi}e_7 + K_{d\phi}(b_1U_2 + a_1X_{10}X_{12} + a_2X_{10} - \ddot{\phi}_d) \tag{4.128}$$

Rearranging equation (4.128), the entire control law is:

$$\begin{aligned} U_2 = \frac{1}{K_{d\phi}b_1}\Big(&-k_1|S_7|^{1/2}\text{sign}(S_7) - k_2\int \text{sign}(S_7)\,dt + K_{d\phi}\ddot{\phi}_d - K_{p\phi}\dot{e}_7 - K_{i\phi}e_7 \\ &-K_{d\phi}a_1X_{10}X_{12} - a_2X_{10}K_{d\phi}\Big) \end{aligned} \tag{4.129}$$

To find $U_{ST(2)}$, since:

$$U_2 = U_{\text{eq}(2)} + U_{ST(2)} \tag{4.130}$$

Where: U_2 is the Super Twisting Sliding Mode Controller for ϕ, $U_{\text{eq}(2)}$ is the equivalent controller for ϕ, and $U_{ST(2)}$ is the continuous controller for ϕ.

Rearranging equation (4.130) and substituting equations (4.67) and (4.129):

$$\begin{aligned} U_{ST(2)} = U_2 - U_{\text{eq}(2)} = \frac{1}{K_{d\phi}b_1}\Big(&-k_1|S_7|^{1/2}\text{sign}(S_7) - k_2\int \text{sign}(S_7)\,dt + K_{d\phi}\ddot{\phi}_d - K_{p\phi}\dot{e}_7 \\ &- K_{i\phi}e_7 - K_{d\phi}a_1X_{10}X_{12} - a_2X_{10}K_{d\phi}\Big) - \frac{1}{K_{d\phi}b_1}\Big(K_{d\phi}\ddot{\phi}_d - K_{p\phi}\dot{e}_7 - K_{i\phi}e_7 \\ &- K_{d\phi}a_1X_{10}X_{12} - K_{d\phi}a_2X_{10}\Big) \end{aligned} \tag{4.131}$$

The continuous controller is:

$$U_{ST(2)} = \frac{1}{K_{d\phi}b_1}\left(-k_1|S_7|^{1/2}\text{sign}(S_7) - k_2\int \text{sign}(S_7)\,dt\right) \tag{4.132}$$

- **Design for Theta**

The equivalent controller $U_{\text{eq}(\theta)}$ is the same as equation (4.84), which is done in the design of SMC.

Reaching law: Based on STA, the closed-loop sliding dynamics is set as follows:

$$\dot{S}_9 = -k_1|S_9|^{1/2}\text{sign}(S_9) - k_2\int \text{sign}(S_9)\,dt \tag{4.133}$$

Where: k_1 and k_2 are positive constant numbers.

$\dot{S}_9 = \dot{S}_9$, so equating equation (4.82) and (4.133) yields:

$$-k_1|S_9|^{1/2}\text{sign}(S_9) - k_2\int \text{sign}(S_9)\,dt = K_{p\theta}\dot{e}_9 + K_{i\theta}e_9 + K_{d\theta}\left(b_2U_3 + a_3X_8X_{12} - a_4X_8 - \ddot{\theta}_d\right) \tag{4.134}$$

Rearranging equation (4.134), the entire control law is:

$$\begin{aligned} U_3 = \frac{1}{K_{d\theta} b_2} \Big(& -k_1 |S_9|^{1/2} \text{sign}(S_9) - k_2 \int \text{sign}(S_9)\, dt \; + K_{d\theta} \ddot{\theta}_d - K_{p\theta} \dot{e}_9 - K_{i\theta} e_9 \\ & - K_{d\theta} a_3 X_8 X_{12} + a_4 X_8 K_{d\phi} \Big) \end{aligned} \tag{4.135}$$

To find U_{ST}, Since:

$$U_3 = U_{\text{eq}(3)} + U_{\text{ST}(3)} \tag{4.136}$$

Where: U_3 is Super twisting sliding mode controller for θ, $U_{\text{eq}(3)}$ is equivalent controller for θ, and $U_{\text{ST}(3)}$ is continuous controller for θ.

Rearranging equation (4.136) and substituting equation (4.84) and (4.135):

$$\begin{aligned} U_{\text{ST}(3)} = U_3 - U_{\text{eq}(3)} = & \frac{1}{K_{d\theta} b_2} \Big(-k_1 |S_9|^{1/2} \text{sign}(S_9) - k_2 \int \text{sign}(S_9)\, dt + K_{d\theta} \ddot{\theta}_d - K_{p\theta} \dot{e}_9 \\ & - K_{i\theta} e_9 - K_{d\theta} a_3 X_8 X_{12} + a_4 X_8 K_{d\phi} \Big) - \frac{1}{K_{d\theta} b_2} \Big(K_{d\theta} \ddot{\theta}_d - K_{p\theta} \dot{e}_9 - K_{i\theta} e_9 \\ & - K_{d\theta} a_3 X_8 X_{12} + K_{d\theta} a_4 X_8 \Big) \end{aligned} \tag{4.137}$$

The continuous controller is:

$$U_{\text{ST}(3)} = \frac{1}{K_{d\theta} b_2} \left(-k_1 |S_9|^{1/2} \text{sign}(S_9) - k_2 \int \text{sign}(S_9)\, dt \right) \tag{4.138}$$

- **Design for Psi**

The equivalent controller $U_{\text{eq}(\psi)}$ is the same as equation (4.101), which is done in the design of SMC.

Reaching law: Based on STA, the closed-loop sliding dynamics are set as follows:

$$\dot{S}_{11} = -k_1 |S_{11}|^{\frac{1}{2}} \, \text{sign}(S_{11}) - k_2 \int \text{sign}(S_{11})\, dt \tag{4.139}$$

Where: k_1 and k_2 are positive constant numbers.

So $\dot{S}_{11} = \dot{S}_{11}$, equating equation (4.99) and (4.139) yields:

$$-k_1 |S_{11}|^{\frac{1}{2}} \, \text{sign}(S_{11}) - k_2 \int \text{sign}(S_{11})\, dt = K_{p\psi} \dot{e}_{11} + K_{i\psi} e_{11} + K_{d\psi} (b_3 U_4 + a_5 X_8 X_{10} - \ddot{\psi}_d) \tag{4.140}$$

Rearranging equation (4.140), the entire control law is:

$$\begin{aligned} U_4 = \frac{1}{K_{d\psi} b_3} \Big(& -k_1 |S_{11}|^{1/2} \text{sign}(S_{11}) - k_2 \int \text{sign}(S_{11})\, dt \; + K_{d\psi} \ddot{\psi}_d - K_{p\psi} \dot{e}_{11} - K_{i\psi} e_{11} \\ & - K_{d\psi} a_5 X_8 X_{10} \Big) \end{aligned} \tag{4.141}$$

To find U_{ST}, since:

$$U_4 = U_{\text{eq}(4)} + U_{\text{ST}(4)} \tag{4.142}$$

Where: U_4: Super twisting sliding mode controller for ψ, $U_{\text{eq}(4)}$: Equivalent controller for ψ and $U_{\text{ST}(4)}$: Continuous controller for ψ

Rearranging equation (4.142) and substituting equation (4.101) and (4.141) yields:

$$\begin{aligned}U_{\text{ST}(4)} = U_4 - U_{\text{eq}(4)} = \frac{1}{K_{d\psi}b_3}\bigg(&-k_1|S_{11}|^{1/2}\text{sign}(S_{11}) - k_2\int \text{sign}(S_{11})\,dt + K_{d\psi}\ddot{\psi}_d - K_{p\psi}\dot{e}_{11}\\ &- K_{i\psi}e_{11} - K_{d\psi}a_5X_8X_{10}\bigg) - \frac{1}{K_{d\psi}b_3}\bigg(K_{d\psi}\ddot{\psi}_d - K_{p\psi}\dot{e}_{11} - K_{i\psi}e_{11} - K_{d\psi}a_5X_8X_{10}\bigg)\end{aligned} \tag{4.143}$$

The continuous controller is:

$$U_{\text{ST}(4)} = \frac{1}{K_{d\psi}b_3}\left(-k_1|S_{11}|^{\frac{1}{2}}\,\text{sign}(S_{11}) - k_2\int \text{sign}(S_{11})\,dt\right) \tag{4.144}$$

The stability analysis is in the Appendix F.2.

The position and attitude Super Twisting SMC of the quadcopter is summarized as:

$$U_x = \frac{m}{K_{dx}}\left(-k_1\sqrt{|S_1|}\,\text{sign}(S_1) - k_2\int \text{sign}(S_1)\,dt + K_{dx}\ddot{X}_d - K_{px}\dot{e}_1 - K_{ix}e_1\right)$$

$$U_y = \frac{m}{K_{dy}}\left(-k_1\sqrt{|S_3|}\,\text{sign}(S_3) - k_2\int \text{sign}(S_3)\,dt + K_{dy}\ddot{Y}_d - K_{py}\dot{e}_3 - K_{iy}e_3\right)$$

$$U_z = \frac{m}{K_{dz}}\left(-k_1\sqrt{|S_5|}\,\text{sign}(S_5) - k_2\int \text{sign}(S_5)\,dt + K_{dz}\ddot{Z}_d - K_{pz}\dot{e}_5 - K_{iz}e_5\right)$$

$$\begin{aligned}U_2 = \frac{1}{K_{d\phi}b_1}\bigg(&- k_1\sqrt{|S_7|}\,\text{sign}(S_7) - k_2\int \text{sign}(S_7)\,dt + K_{d\phi}\ddot{\phi}_d - K_{p\phi}\dot{e}_7 - K_{i\phi}e_7\\ &- K_{d\phi}a_1X_{10}X_{12} - a_2X_{10}K_{d\phi}\bigg)\end{aligned}$$

$$\begin{aligned}U_3 = \frac{1}{K_{d\theta}b_2}\bigg(&- k_1\sqrt{|S_9|}\,\text{sign}(S_9) - k_2\int \text{sign}(S_9)\,dt + K_{d\theta}\ddot{\theta}_d - K_{p\theta}\dot{e}_9 - K_{i\theta}e_9\\ &- K_{d\theta}a_3X_8X_{12} + a_4X_8K_{d\phi}\bigg)\end{aligned}$$

$$\begin{aligned}U_4 = \frac{1}{K_{d\psi}b_3}\bigg(&- k_1\sqrt{|S_{11}|}\,\text{sign}(S_{11}) - k_2\int \text{sign}(S_{11})\,dt + K_{d\psi}\ddot{\psi}_d - K_{p\psi}\dot{e}_{11} - K_{i\psi}e_{11}\\ &- K_{d\psi}a_5X_8X_{10}\bigg)\end{aligned}$$

4.4 Parameters Tuning

Since STSMC has high control effort and complexity in parameter twining problem then a careful selection of the control parameters is crucial to strike a balance between achieving fast convergence and minimizing overshoot.

Using fixed PID sliding surface parameters in a sliding mode controller has:

Low Performance: The fixed parameters do not provide the best control performance when the system characteristics change, leading to reduced overall control performance.

Sensitivity to Parameter Selection: Selecting appropriate fixed parameters is challenging. This sensitivity to parameter selection makes it difficult to achieve the desired control performance consistently.

Tuning parameters using fuzzy logic enable the automatic adjustment of the controller parameters based on the system's dynamics, parameter variation, and perturbations encountered during operation.

4.4.1 Fuzzy Logic Control

Fuzzy logic control is a control system approach that employs fuzzy logic to make decisions. Fuzzy logic is a mathematical system for making choices and thinking in situations where the information is not very clear, so it helps create more detailed results. In contrast to traditional binary logic, which confines options to either true or false values, fuzzy logic accommodates degrees of truth, thus offering a more flexible way to express knowledge and information [60].

Fuzzy logic, pioneered by Lotfi Zadeh in the 1960s as an extension of classical logic, recognizes that real-world problems often lack clear-cut distinctions and instead involve various shades of interpretation. Fuzzy logic offers a means to model imprecision and ambiguity by assigning membership degrees to different categories. The foundation of fuzzy logic is built upon fuzzy sets, which consist of elements with varying degrees of membership. Each element can possess a membership value ranging from 0 to 1, representing the degree of association with that specific set. By employing linguistic variables such as "high" and "low," and formulating fuzzy rules that define the relationships between these variables, fuzzy logic can effectively capture the subjective nature of human reasoning [61].

Fuzzy logic is applied in a range of fields, including control systems and decision-making, due to its capability to manage uncertainty. This feature makes it well-suited for modeling complex systems where conventional logic is not effective [60].

Fuzzy logic offers a significant advantage in its capacity to effectively manage non-linear systems. Through the utilization of fuzzy rules, fuzzy logic controllers can attain robust control, even in scenarios involving parameter variation. Additionally, fuzzy logic provides interpretability, enabling experts to readily grasp the decision-making process. The linguistic labels, membership functions, and rule sets employed in fuzzy logic offer a clear representation of the fundamental reasoning process [62].

4.4.2 Fuzzy Logic-Based Gain Tuning

- **Fuzzy Logic-Based Gain Tuning for PID Sliding Surfaces**

Using large PID sliding surface parameters in STSMC for controlling a quadcopter when the error is small have disadvantages, such as:

1. Overshoot: Large sliding surface parameters causes the control action to be more aggressive, leading to overshoot.

2. Increased Control Effort: Large sliding surface parameters increases the control effort required to stabilize the system. This leads to higher motor torque or thrust inputs.

Using small PID sliding surface parameters in STSMC for controlling a quadcopter when the error is large have slower convergence problem.

Slower Convergence: Small sliding surface parameters limit the controller's ability to quickly converge the system states to the desired sliding surface. This results longer settling time.

A large value K_p, K_i, and K_d can be selected when $|e|$ is large, to make the control input high and able to converge fast. A small value of K_p, K_i, and K_d can be selected when $|e|$ is small, to decrease overshoot and control effort. Therefore, if the parameters can be tuned based on the above argument, a more acceptable performance (fast convergence, small control effort, and eliminate overshoot) can be achieved. Thus, to have a value of K_p, K_i, and K_d which can maintain a balance between the two facts, a fuzzy controller rule base is used for tuning K_p, K_i, and K_d based on $|e|$ value. A one-input three-output fuzzy system is designed for this application. Input taken is $|e|$ while the output is K_p, K_i, and K_d. The rule base used is given in Table 4.1. Input $|e|$ has four membership functions, and output K_p, K_i, and K_d also have four membership functions for each.

Four fuzzy sets named ZR, PS, PM, and PB respectively for Zero, Positive small, Positive medium, and Positive big are chosen to fuzzify the error. Similarly, the parameters K_p, K_i, and K_d are fuzzified into four fuzzy sets named ZR, PS, PM, and PB respectively for Zero, Positive small, Positive medium, and Positive big.

$\|e\|$	ZR	PS	PM	PB
K_p	ZR	PS	PM	PB
K_i	ZR	PS	PM	PB
K_d	ZR	PS	PM	PB

Table 4.1: Fuzzy Rules for Error and PID Sliding Surface Parameters

The rules used to map the input and output fuzzy sets are as follows:

Rule1: if $|e|$ is ZR, then K_p, K_i, and K_d is ZR.

Rule2: if $|e|$ is PS, then K_p, K_i, and K_d is PS.

Rule3: if $|e|$ is PM, then K_p, K_i, and K_d is PM.

Rule4: if $|e|$ is PB, then K_p, K_i, and K_d is PB.

- **Fuzzy Logic-Based Gain Tuning for k_1 and k_2**

Using a large super twisting algorithm parameter in STSMC for controlling a quadcopter when the sliding surface is small leads to the following disadvantages:

1. High Control Effort: A large super twisting algorithm parameter results in high control efforts being applied by the system. This high control effort leads to increased power consumption.

2. Reduced Precision: The large parameter value causes a reduction in the precision of the control system. As the sliding surface becomes smaller, the high control effort make it challenging to achieve control over the quadcopter's motion. This leads to decreased tracking precision.

Using small super twisting algorithm parameters in STSMC for controlling a quadcopter when the sliding surface is large have disadvantages:

1. Slow Convergence: A slower convergence delays the quadcopter's ability to stabilize and correct the error, leading to reduced performance.

2. Reduced Control Authority: Small super twisting algorithm parameters limit the control authority of the system. With a large sliding surface, the quadcopter requires more control effort to bring it back to the desired state. In such cases, having small parameters reduces the effectiveness of the control action.

A large value k_1 and k_2 can be selected when $|S|$ is large, to increase control authority and able to converge fast. A small value of k_1 and k_2 can be selected when $|S|$ is small, to decrease control effort and increase precision. Therefore, if the parameters can be tuned based on the above argument, a more acceptable performance (fast convergence, small control effort, best precision, and increased control authority) can be achieved. Thus, to have a value of k_1 and k_2 which can maintain a balance between the two facts, a fuzzy controller rule base is used for tuning k_1 and k_2 based on $|S|$ value. A one-input two-output fuzzy system is designed for this application. Input taken is $|S|$ while the output is k_1 and k_2. The rule base used is given in Table 4.2. Input $|S|$ has five membership functions, and output k_1 and k_2 also have five membership functions for each.

Five fuzzy sets named VS, S, M, L, and VL respectively for Very small, small, medium, large, and very large are chosen to fuzzify the error. Similarly, the parameters k_1 and k_2 are fuzzified into five fuzzy sets named VS, S, M, L, and VL respectively for Very small, small, medium, large, and very large.

$\|S\|$	VS	S	M	L	VL
k_1	VS	S	M	L	VL
k_2	VS	S	M	L	VL

Table 4.2: Fuzzy Rules for Sliding Surface and Super Twisting Algorithm Parameters

The rules used to map the input and output fuzzy sets are as follows [63]:

Rule 1: if $|S|$ is VS, then k_1 and k_2 are VS.

Rule 2: if $|S|$ is S, then k_1 and k_2 are S.

Rule 3: if $|S|$ is M, then k_1 and k_2 are M.

Rule 4: if $|S|$ is L, then k_1 and k_2 are L.

Rule 5: if $|S|$ is VL, then k_1 and k_2 are VL.

The membership function used here is Triangular function because of its:

1. Simplicity: Triangular membership functions are straightforward and easy to implement. They exhibit a basic triangular configuration defined by three parameters: the lower limit, upper limit, and peak value. This simplicity makes them more understandable compared to other complex membership functions.

2. Interpretability: The triangular membership functions provide a clear and interpretable representation of linguistic terms. Each vertex of the triangle corresponds to a specific linguistic term, such as "low," "medium," or "high," which makes it easier to understand the fuzzy logic system and its decision-making process.

3. Reduced Computational Complexity: Computational operations is relatively simpler when using triangular membership functions. The calculations involved in determining the centroid or center of gravity of a triangular shape are less computationally intensive compared to other membership functions with more complex shapes.

Membership functions of input ($|e|$ and $|S|$) and output for (X, Y, Z, ϕ, θ, ψ, k_1, and k_2) are shown in Appendix G.

Motor Selection: For this thesis DX2205 2300KV Brushless DC Motor is selected. The specification are in Table 4.3

Specification	**Value**
Brand	FancyWhoop
Speed	2000 RPM
Voltage	7.4 Volts
Horsepower	408 Watts
Item Dimensions (L x W x H)	31 x 31 x 25 mm
Item Weight	28 grams

Table 4.3: Specifications of DX2205 2300KV Brushless DC Motor [64]

The maximum torque is calculated as follows:

$$\tau = \frac{P}{\omega} = \frac{408\,\mathrm{W}}{2000\,\mathrm{rpm}} = 0.204\,\mathrm{Nm}$$

Where: P = Power, ω = Angular velocity, and τ = Torque

Since the radius of propeller is 0.15m;

$$\text{Thrust force} = \left(\frac{\pi}{2} * D^2 * \rho * P^2\right)^{1/3} = \left(\frac{\pi}{2}\,(0.09)^2\,(1.225)\,(408)^2\right)^{1/3} = 55\,\mathrm{N}$$

Where:

D: represents the diameter of the object.

ρ : denotes the fluid density in which the object is moving.

P: represents Power.

Chapter Five

Wheat Yield Estimation

5.1 Introduction

The application of this thesis is to estimate the wheat yield. Introducing an advanced system for wheat yield estimation that utilizes YOLOv8 for the detection and counting of wheat heads. This state of the art technology revolutionizes crop assessment, providing farmers with real-time data for data driven decisions and enhanced harvest predictions.

5.2 System Description

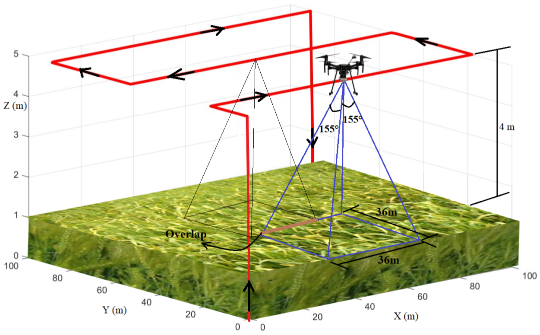


Figure 5.1: Quadcopter Movement in a Wheat Farm and Camera's Capture Range

Assumptions:

→ The terrain of the farm is level, and it exhibits uniform elevation throughout.
→ The entirety of the farm is devoid of obstacles.

The process of estimating wheat yield begins with a quadcopter equipped with an imaging system that captures images along a predetermined trajectory, as discussed in the previous chapter. These images are then stitched together to eliminate any overlapping regions. The stitched image is subsequently divided into smaller segments, typically measuring 1 square meter each, and fed into a specially trained system for analysis.

This intelligent system excels at its first task: detecting and counting wheat heads within each image segment. It iterates through all the image segments, summing up the total number of heads detected. To calculate yield estimation, the system requests input for the following parameters: the average number of kernels per wheat head, the weight of one kernel, and the area of the ground, which is generally derived from ground samples.

With all the essential inputs collected, the system generates a wheat yield estimation. It combines the head count, kernels per head, weight, and area, thus providing an assessment of the wheat yield. In this manner, the integration of quadcopter technology, image processing, and ground-based data collection ensures a reliable approach to wheat yield estimation.

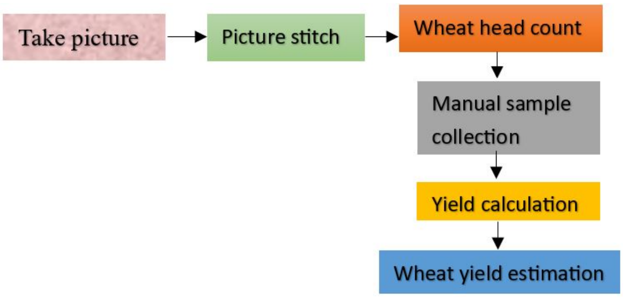


Figure 5.2: Flow Chart for Wheat Yield Estimation

For this thesis, the GoPro HERO9 Black camera with a 155 degree field of view (FOV) has been selected. With this camera's wide FOV, it can capture an area of 1296 square meters from a distance of 4 meters. To minimize overlapping regions in the photos, the camera will be adjusted to capture an image every 72 seconds. This 72-second interval is chosen because the quadcopter's speed is 0.5 meters per second. In 72 seconds, it can move 36 meters, which is the distance needed to capture a photo. The reason for selecting 36 meters is that the camera's field of view (FOV) is 155 degrees, and at a height of 4 meters, it can cover an area of 36 meters by 36 meters. With this capture rate, it captures 10 photos in one hectare of land, as depicted in Figure 5.3, with an overlap of 0.04m.

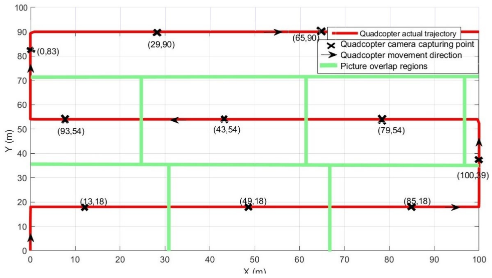


Figure 5.3: Quadcopter Camera Capturing Points and Overlap Regions

Specification	**Value**
Weight	158g
Wide-Angle Lens	155° FOV (field of view)
Photo Resolutions	20 Megapixel
Size	71.0 x 55.0 x 33.6 mm
Timer	Equipped with a timer feature that enables users to capture photos with a preset delay.

Table 5.1: GoPro HERO9 Black Camera Specifications [65]

5.3 Image Recognition

Image recognition is a computer vision process where a computer system analyzes and identifies objects, patterns, or features within images. It enables the categorization and understanding of visual content, which is a critical capability with applications ranging from facial recognition and autonomous vehicles to medical diagnostics. It has witnessed remarkable advancement in recent years [66]. YOLOv8 (You Only Look Once version8) is at the forefront of this progress, offering state of the art capabilities in real-time detection of objects [67].

YOLO is groundbreaking object detection algorithm developed by Joseph Redmon and Santosh Divvala in 2016. It is widely utilized for purposes such as object recognition, tracking, and counting in images and videos. YOLO's key characteristic is its ability to detect objects in an image frame using a single deep neural network pass [67]. For this thesis training is done using YOLOv8 in PyCharm software.

PyCharm is a software created with a specific focus on Python programming. It is developed by JetBrains and is widely used by Python developers for coding [68].

5.3.1 The Training Process of YOLOv8

Training YOLOv8 involves:

1. **Data Collection and Annotation:** The first step in training any object detection model is to collect a diverse dataset that represents the types of objects the model will detect. These images must be annotated with bounding boxes indicating the location of objects and their corresponding class labels. High-quality annotations are crucial for training a robust model. For this thesis data is taken from the global wheat head dataset [45], totaling 3530 images. From these images, 3000 are used for training, 500 for validation, and 30 for testing. Data annotation for YOLOv8 format is done by using makesense.ai.

   MakeSense.ai is an online annotation tool created for computer vision applications, including tasks like object detection. It provides an interface for annotating and labeling datasets, making it easier to generate training data [69].

2. **Loss Function:** YOLOv8 utilizes a mix of loss functions during training. These functions primarily assess the precision of bounding box predictions (localization loss) and object

presence (confidence loss). These losses guide the model to make accurate predictions.

3. **Training:** Training YOLOv8 involves optimizing the model's parameters using a large dataset. Training continues until the model converges, meaning it achieves satisfactory object detection performance on the validation dataset.

   In the realm of YOLOv8, transfer learning is a pivotal technique that empowers object detection tasks. Transfer learning involves the initialization of a YOLOv8 model using the weights and architecture of a pretrained neural network. These pretrained models have undergone extensive training on vast image datasets for tasks like image recognition.

   By initializing with pretrained models, YOLOv8 inherits valuable feature representations, allowing it to grasp essential patterns within images. Fine-tuning this initialized model with specific datasets related to object detection tasks, such as wheat head counting, enables the model to adapt efficiently. This approach not only speed up the model's training but also substantially enhances its performance, demonstrating the central role of transfer learning and pretrained models in achieving precise and efficient object detection within the YOLOv8 framework.

4. **Evaluation:** After the training phase, the model's ability to detect objects is assessed using metrics like Mean Average Precision (mAP) on a separate test dataset. The model undergoes several iterations of training and evaluation to achieve the desired performance.

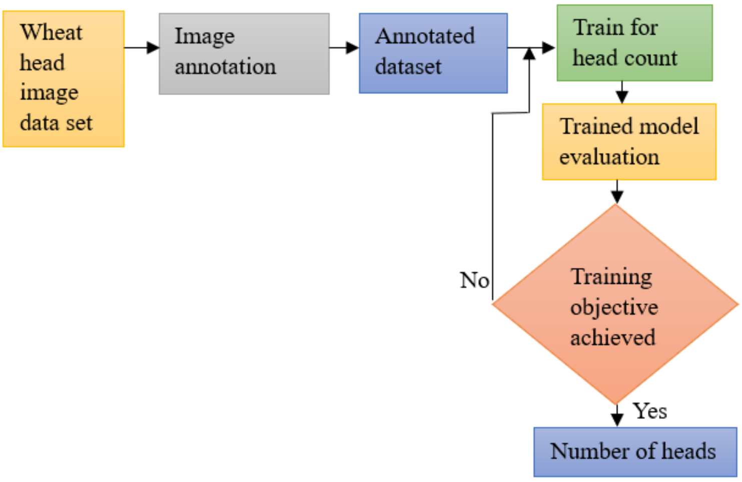


Figure 5.4: Flow Chart of Wheat Head Count

Working Steps of YOLO:

1. **Input Image:** YOLO takes an input image, and it divides the image into a grid.

2. **Grid Division:** The image is partitioned into a grid of cells, and each cells role is to predict objects present within its assigned region.

3. **Feature Extraction:** YOLO uses a deep convolutional neural network to extract features from the entire image.

   A Convolutional Neural Network (CNN) is a specialized deep learning model created for the analysis of visual data, like images. It employs a layered structure, incorporating convolutional layers to extract features from the input data, pooling layers to reduce spatial dimensions while retaining crucial information, and fully connected layers for integrating extracted features and making high-level decisions. These fully connected layers utilize the extracted features to generate predictions, making CNNs effective in tasks like image recognition and object detection.

4. **Bounding Box Prediction:** YOLO predicts bounding boxes for each cell in the grid, with each bounding box characterized by:
   a. Center coordinates (x, y) of bounding box.
   b. The bounding box's width (w) and height (h) .
   c. Confidence score, which indicates how likely it is that an object exists in the bounding box.

5. **Class Prediction:** For each cell, YOLO predicts class probabilities for the objects detected within that cell.

6. **Non-Maximum Suppression:** Following predictions, YOLO employs non-maximum suppression to eliminate redundant bounding boxes.

7. **Output:** The ultimate result from YOLO comprises the remaining bounding boxes, along with their corresponding class labels and confidence scores.

Chapter Six

Simulation Result and Discussion

6.1 Introduction

This chapter is structured into two distinct sections. The first part explores the test results obtained from employing the Fuzzy STSMC with PID surface for controlling the quadcopter across various trajectory scenarios. The second section provides a comprehensive explanation of the outcomes of image processing and how the trained system derives wheat yield estimates, along with the resulting findings.

6.2 Controller Outcome

Graphical representations of simulation results, created using MATLAB/SIMULINK, provide insights into the performance of the Fuzzy STSMC with PID surface. This controller integrates the adaptability of fuzzy logic with the robustness of STSMC. Testing covers spiral infinity trajectories without disturbances and square wave trajectories with and without disturbances. Furthermore, Comparison is made in spiral cylindrical trajectory with and without variations in parameters.

6.2.1 Spiral Infinity Trajectory

In the context of space navigation, a Spiral infinity trajectory was utilized as the tracking reference, with sinusoidal input trajectories applied to both the X and Y axes, and ramp input to Z axes which is: $X_d = 1 - \cos(t)$, $Y_d = 0.5\sin(2t)$, $Z_d = t$ and $\psi_d = 0$. The gains are automatically tuned using fuzzy logic, which is provided in Appendix G.

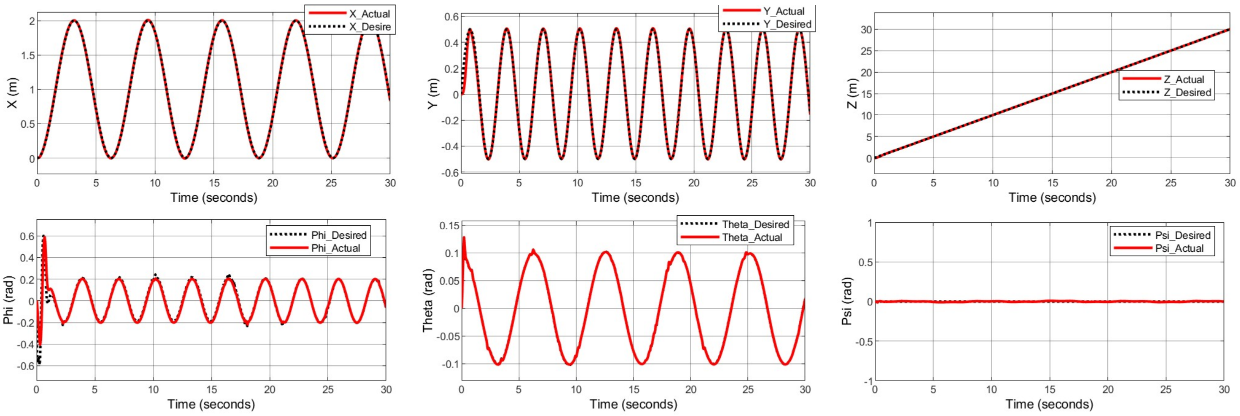


Figure 6.1: Spiral Infinity Trajectory Tracking of Position and Attitude

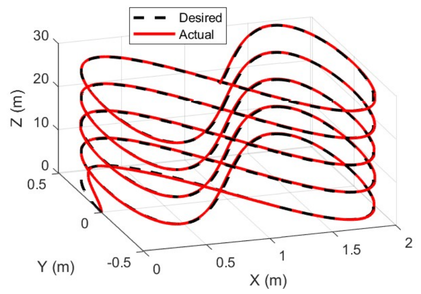


Figure 6.2: 3D Plot of Spiral Infinity Trajectory Tracking

The figures showcase the tracked positions in the X, Y, Z axes and the orientations in ϕ, θ, and ψ angles, as depicted in Figure 6.1. Additionally, Figure 6.2 illustrates the resulting 3D trajectory tracking response. The acquired responses indicate that the proposed Fuzzy STSMC with PID sliding surface controller effectively ensures both accurate reference tracking and fast convergence while maintaining stability.

6.2.2 Application Based Trajectory (Square Wave)

Area Coverage

The square wave serves as a simple path to ensure complete area coverage. In this thesis, the waypoints are manually defined. A navigation speed of 0.5 m/s is selected for the creation of the flight time table, which is provided in Table 6.1. First, the Quadcopter moves from the ground to 5 meters in the Z-direction. Then it moves 18 meters in the Y-direction, followed by moving 100 meters in the X-direction. The plant height is 1 meters at the heading stage, so the photo will be taken from 4 meters from the tip of the plant. To calculate the trajectory tracking reference, polynomials are employed. For this thesis, a 3$^{\text{rd}}$ order polynomial is used.

The calculation and code of input trajectories are written in Appendix E.1. The gains are automatically tuned using fuzzy logic, which is provided in Appendix G.

Time (sec)	X (m)	Y (m)	Z (m)
0	0	0	0
2.5	0	0	5
11.5	0	18	5
61.5	100	18	5
79.5	100	54	5
129.5	0	54	5
147.5	0	90	5
197.5	100	90	5
200	100	90	0

Table 6.1: Flight Time Table

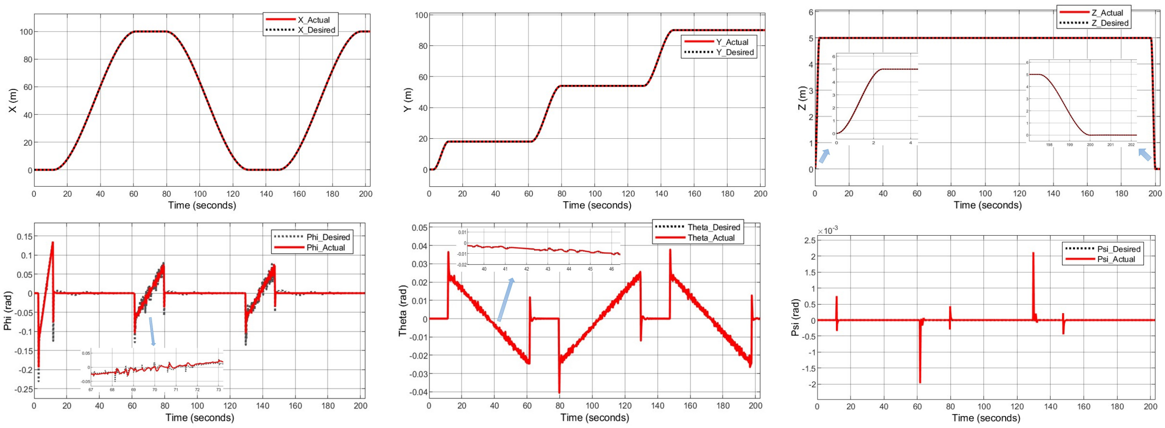


Figure 6.3: Square Wave Tracking of Position and Attitude

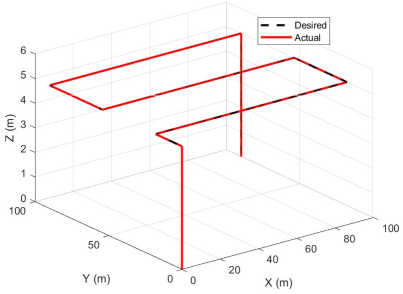


Figure 6.4: 3D Plot of Square Wave Trajectory

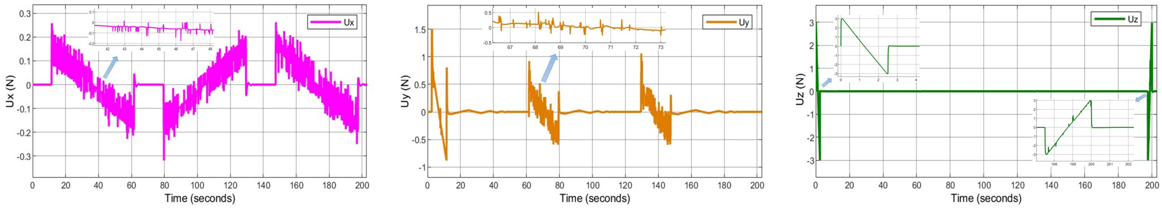


Figure 6.5: Virtual Controller Response for Position Trajectory

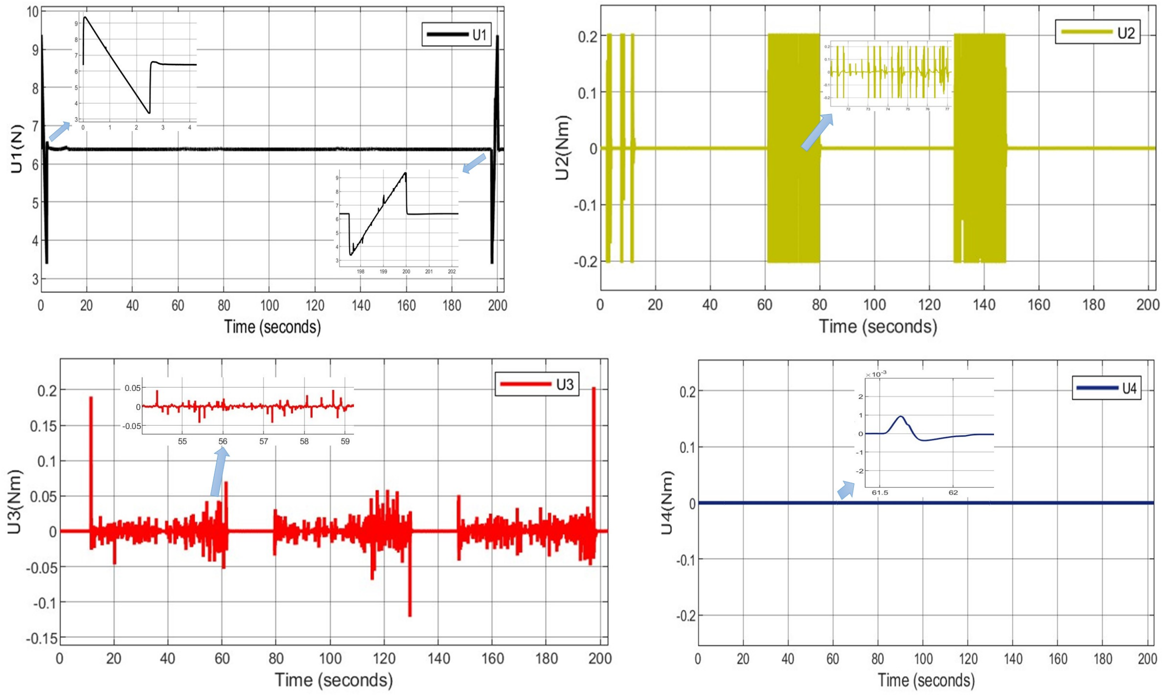


Figure 6.6: Altitude and Attitude Controllers Effort of Quadcopter

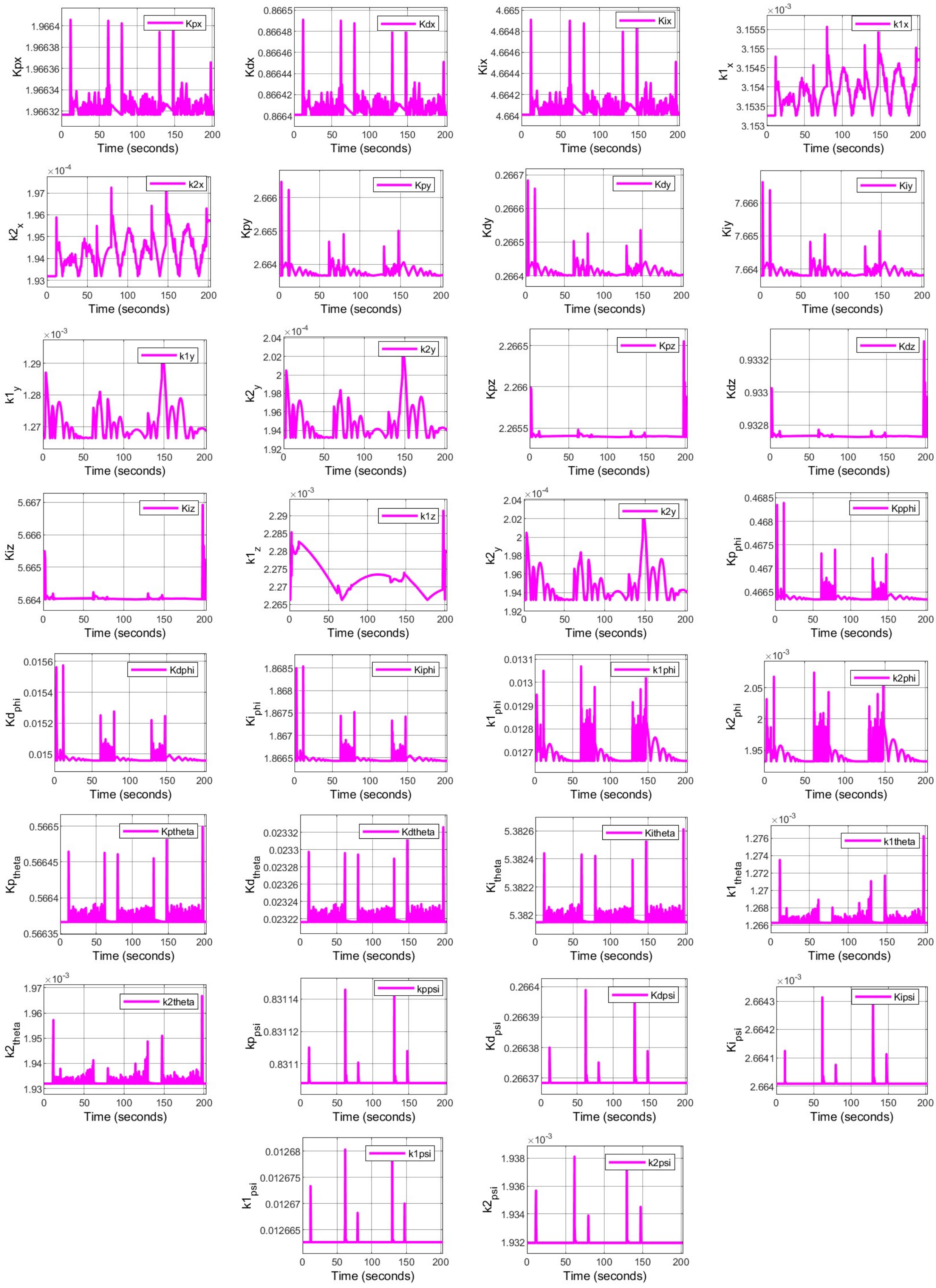


Figure 6.7: Parameters of Square Wave

It can be noticeably observed from Figure 6.3 that the proposed controllers have good performances (reduced errors) for positions (X, Y, Z) control. And the control scheme ensures that the quadrotor maintains its attitude angles to allow it to perform tracking of the target configuration. Figure 6.4 shows the 3D response of the system obtained by the proposed method, so the controller makes the quadcopter track the desired input.

The outer loop or virtual controller signals U_x, U_y, and U_z, which are used to determine the desired roll ϕ_d and pitch θ_d and attitude controller U_1, are shown in Figure 6.5. The amplitude of the controller signal fluctuates in response to the input trajectory to determine the necessary force required to sustain the desired trajectory.

The altitude controller U_1 and yaw angle controller U_4 are responsible for controlling the specified altitude and yaw angle provided by the task manager, respectively. Conversely, the attitude roll angle controller U_2 and pitch angle controller U_3 control the reference trajectory generated from outer loop controller.

6.2.3 Disturbance Rejection Capacity

In order to show the efficiency and robustness of Fuzzy STSMC with PID surface scheme in tracking trajectory, bounded disturbance is applied to utilized in real life applications tracking response for a square wave reference trajectory. The external disturbance is represented by a random signal exhibiting differing force amplitudes along U1, U2, and U3 at different time intervals shown in Figure 6.8.

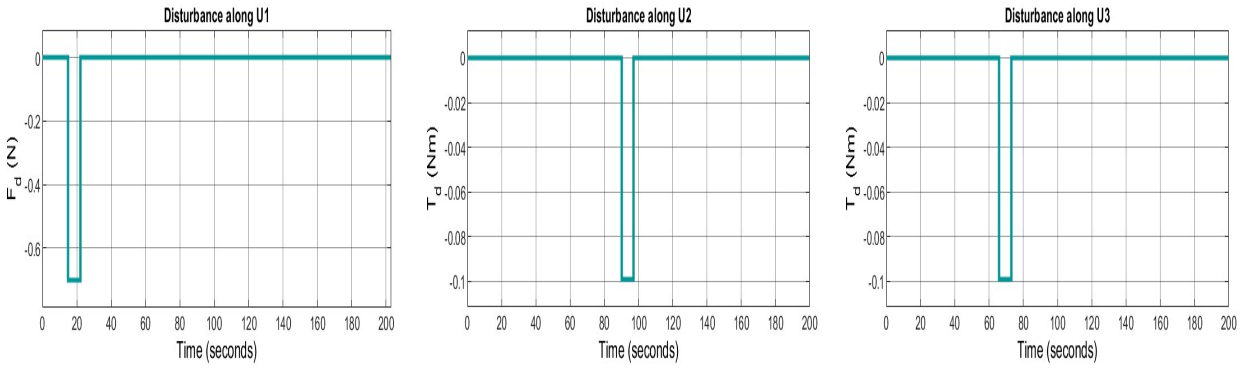


Figure 6.8: Applied Disturbance along U1, U2 and U3

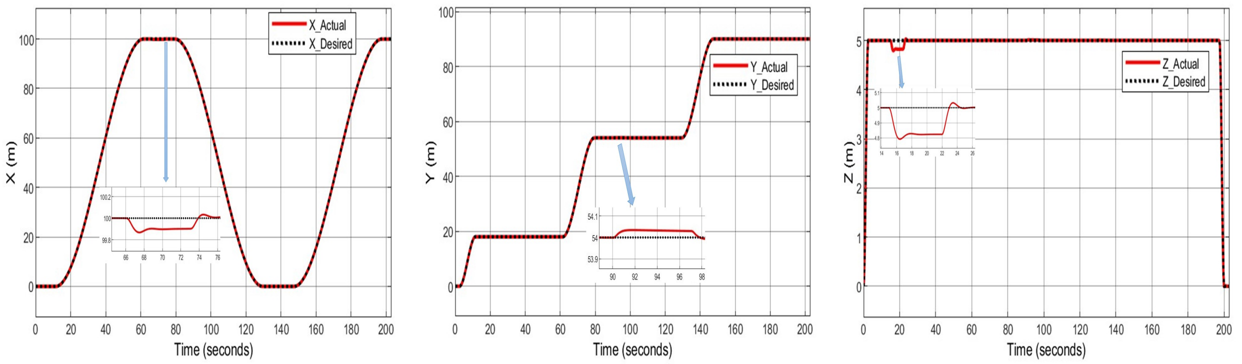


Figure 6.9: Effect of Disturbance

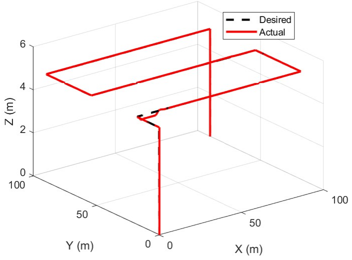


Figure 6.10: Square Wave Response with Disturbance

Figure. 6.9 illustrates the quadcopter's trajectory during the experiment. It commences its ascent from ground level, reaching an altitude of 5 meters within a mere 2.5 seconds. This trajectory persists for the next 15 seconds, during which the quadcopter maintains its position at the 5 meter. However, at the 15th second, it encounters a 0.7 N force in the negative Z direction, causing a temporary descent of 0.2 meters from its initial position. It travels with this height from 15th to 22nd second. Remarkably, within a mere 2.5 seconds, it ascends back to its original altitude of 5 meters, displaying the controller's remarkable disturbance rejection capability.

Subsequently, from the 65th to the 70th second, the quadcopter contends with a 0.099 Nm torque in the negative pitch, which constitutes almost half of the maximum torque the motor can generate. This disturbance leads to a 0.02 meter displacement in the negative X direction. Once the disturbance ceases, the quadcopter promptly retraces its trajectory to the 100 meter and continues its journey.

Between the 90th and 98th seconds, it confronts a 0.1 Nm torque in the negative roll. This disturbance causes a consequential displacement of 0.1 meters in the positive Y direction.

Nevertheless, upon the removal of the disturbance, the quadcopter promptly reverts to its previous position at the 54 meter and seamlessly continues its trajectory. The controller gives control input to counteract the disturbance. These remarkable control outcomes underscore the system's ability to efficiently reject disturbances and maintain the desired flight trajectory, emphasizing the robustness and reliability of the Fuzzy STSMC with PID control strategy.

The system response of the proposed controllers under disturbances in 3D is presented in Figure 6.10. From the simulation result, it is observed that the suggested controller by automatically tuning its parameters and give the amount of appropriate force and torque to the quadcopter, it achieves better performance of the desired trajectory tracking in the presence of disturbances.

6.2.4 Comparison of Fuzzy STSMC with SMC

To demonstrate the performance of FUZZY STSMC with PID surface compared to SMC, a spiral cylindrical trajectory is employed. This trajectory is complex; therefore, achieving the best controller performance is essential. The input is $X_d = 1 - \cos(t)$, $Y_d = \sin(t)$, $Z_d = t$, and $\psi_d = \sin(t)$.

- **Based on Tracking Performance and Chattering Reduction Capacity**

The simulation result for the Fuzzy STSMC with PID sliding surface shows good tracking performance with fast convergence of the states to the desired reference signals. Whereas the sliding mode controller gives low performance and slow convergence as shown in Figure 6.11 and 6.12. The proposed controller reduces chattering that occurs in the control inputs of SMC as shown in Figure 6.13 and 6.14.

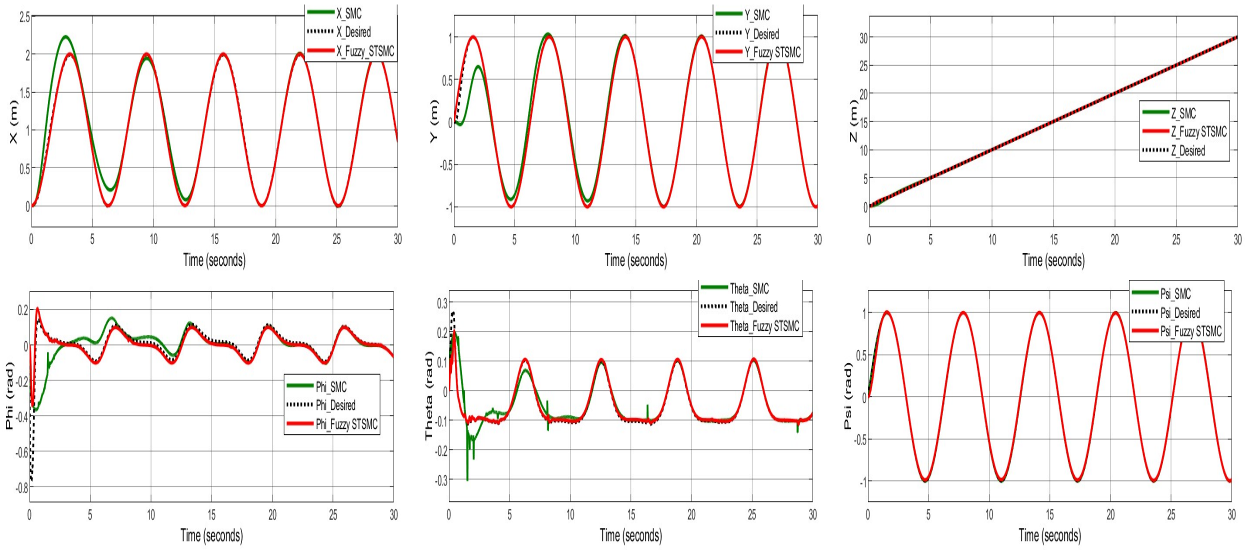


Figure 6.11: Spiral Cylindrical Trajectory Tracking of Positions and Angles

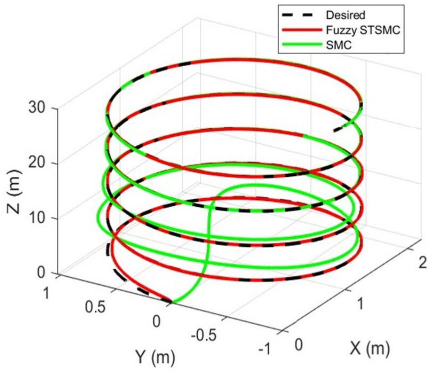


Figure 6.12: 3D Plot of Fuzzy STSMC with SMC in Spiral Cylindrical Trajectory

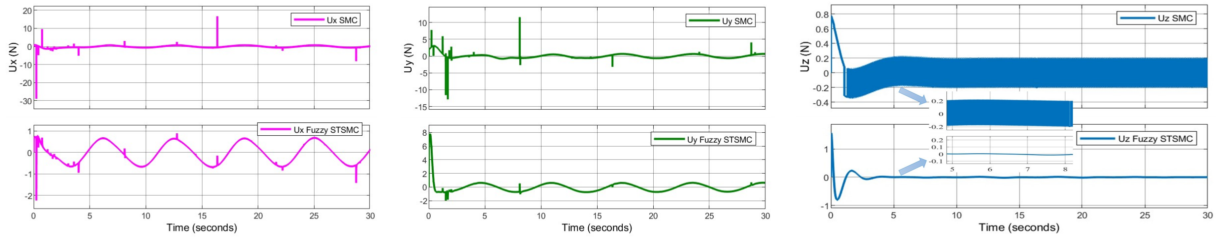


Figure 6.13: Virtual Controller Response for Spiral Cylindrical Trajectory

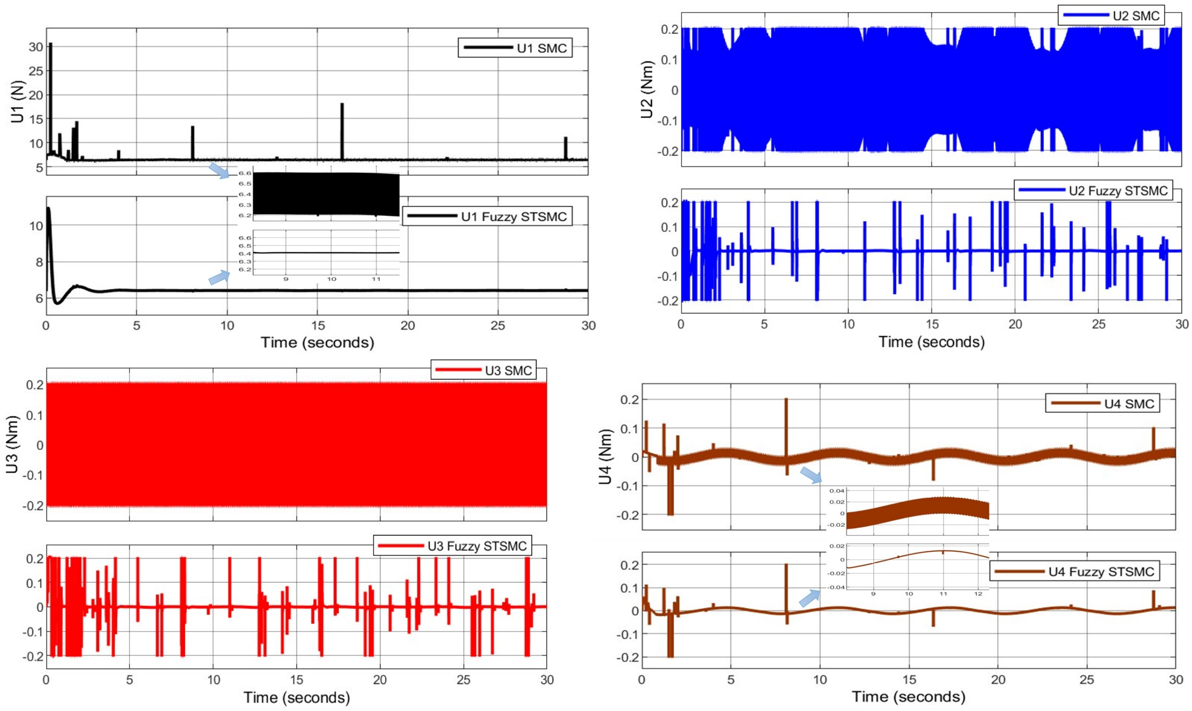


Figure 6.14: Altitude and Attitude Controllers Effort of Quadcopter

Performance Index

Integral Time Absolute Error (ITAE) is used for this thesis as a performance index because ITAE is a useful metric for comparing different control strategies by quantifying the error magnitude over time in a specific way and it tells which control setup results in better error reduction and tracking performance.

	SMC	Fuzzy STSMC
X	10.43	3.495
Y	10.3	1.868
Z	1.981	0.583
Phi	5.469	4.354
Theta	1.702	1.162
Psi	9.33	3.023

Table 6.2: ITAE for $t = 30$ seconds

Table 6.2 shows the proposed controller reduces the error in the $X, Y, Z, Roll(\phi), Pitch(\theta)$, and $Yaw(\psi)$ direction by 66, 49%, 81.86%, 70.57%, 20.38%, 31.72%, 67.59% compared to SMC.

- **Based on Parameter Variation Handling Capacity**

In order to show the efficiency and robustness of the Fuzzy STSMC scheme in tracking trajectory when there is a parameter variation in the quadcopter mass, 0.35 kg that means 53.84% of quadcopter mass is added to the quadcopter, so the mass of the quadcopter becomes 1kg.

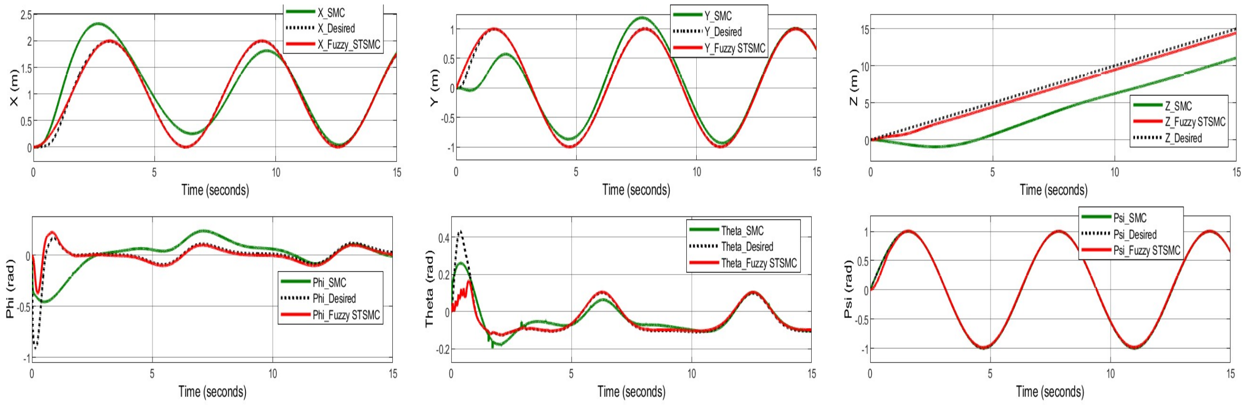


Figure 6.15: Effect of Parameter Variation for Positions and Angles

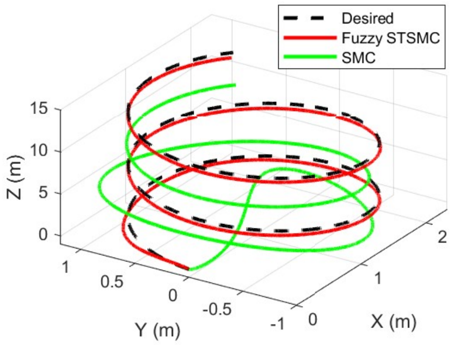


Figure 6.16: 3D Plot of Parameter Variation in Spiral Cylindrical Trajectory

As seen in Figures 6.15 and 6.16, the proposed controller perfectly handles the parameter variation on the mass of the quadcopter.

6.3 Wheat Yield Estimation Outcome

This part consists of two main components: the training and validation of a wheat head detection and count system using YOLOv8 for precise wheat yield estimation and on-ground case studies conducted for comparison. This section presents the outcomes of these efforts, revealing the performance and implications of the YOLOv8-based wheat head detection system in real-world scenarios.

6.3.1 Image Processing

Training is done for 70 epochs and it is completed in 123.783 hr. Training mean average precision is 97.1% and validation mean average precision 96.9%.

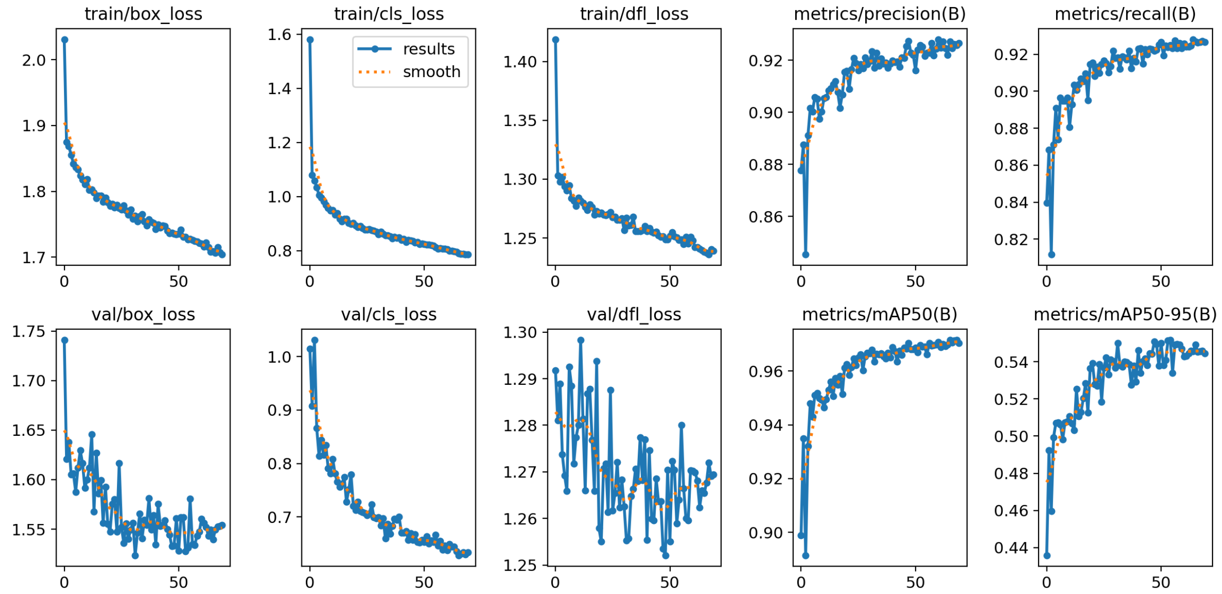


Figure 6.17: Training and Validation Loss and Precision

Train/box_loss and Validation/box_loss:

Train/box_loss and Validation/box_loss are metrics that evaluate the model's precision in detecting objects within images during training and validation, respectively. These gauges how accurately the model identifies object presence, which is crucial for object detection tasks.

Throughout the 70-epoch training of the YOLOv8 model, the "train/box_loss" displayed a notable trend. It initiated at 2.0 in the first epoch, reflecting the model's initial learning phase. Over subsequent epochs, there was a consistent decrease, reaching 1.7 by the 70th epoch. This decline signifies the model's growing proficiency in accurately detecting objects within images, a fundamental aspect of object detection tasks.

Simultaneously, the "Validation/box_loss" followed a similar pattern, commencing at 1.75 and concluding at 1.55. While minor fluctuations occurred in the middle epochs, the overall trajectory indicated gradual improvement. The convergence in both training and validation losses suggests that the model effectively learned to detect objects within images and consistently improved its accuracy throughout training.

The final "train/box_loss" of 1.7 and "Validation/box_loss" of 1.55 reflect the model's optimization in object detection, specifically in object presence detection.

Train/cls_loss (Training Classification Loss): This measures the model's proficiency in classifying objects during the training process. The recorded values show a consistent decline, starting at 1.6 in the first epoch and reaching 0.8 by the 70th epoch. This indicates that the model steadily improved its classification accuracy as training progressed.

Val/cls_loss (Validation Classification Loss): This quantify the model's ability to classify objects. It notably decreased from 1.01 in the first epoch to 0.61 by the 70th epoch, demonstrating consistent improvement in object classification accuracy during training. A lower value at the 70th epoch indicates the model's enhanced ability to accurately classify objects in the validation dataset.

Train/dfl_loss (Training Detection Focal Loss): This assesses how effectively the model distinguishes objects from the background and accurately detects them during the training phase. A decreasing train/dfl_loss indicates improved object detection performance during training. The "train/dfl_loss" metric exhibited consistent improvement during the 70th epoch YOLOv8 training, decreasing from 1.45 to 1.25, indicating the model's effective learning for object detection.

Val/dfl_loss (Validation Detection Focal Loss): This evaluates the model's ability to generalize its object detection capabilities to new, unseen data during the validation process. A consistent or decreasing val/dfl_loss suggests the model's capacity to apply its object detection skills to previously unencountered scenarios. In the case of "val/dfl_loss," while it displayed minor fluctuations in the middle epochs, the overall trend remained smooth, moving from 1.29 to 1.27. This suggests the model's capability to generalize its object detection skills to new data, with minor variations during validation.

Matrics/precision(B): This serves as a measure of precision in object detection, indicating how accurately objects are identified within images. Over the training duration, exhibited a steady increase from 0.84 in the first epoch to 0.93 by the 70th epoch, indicating improved precision in object detection without significant fluctuations.

Matrics/mAP50(B): This evaluates the mean average precision of the model at a 50% intersection-over-union threshold, a critical aspect of object detection tasks. Throughout training, it maintained a consistent upward trajectory, starting at 0.89 and reaching 0.971 by the 70th epoch, without notable fluctuations. This pattern highlights the model's enhanced overall detection accuracy, particularly at a critical 50% intersection-over-union threshold.

Matrics/recall(B): This measures the recall or sensitivity of the model in object detection, indicating how effectively it identifies all relevant objects within images. Over the training duration, indicated a consistent increase in recall from 0.81 in the first epoch to 0.93 by the 70th epoch. This signifies the model's effectiveness in identifying relevant objects within images, with stable performance throughout training.

Matrics/mAP50-95(B): This evaluates the mean average precision over a range of IoU thresholds (from 50% to 95%), a crucial aspect of object detection tasks. Throughout training, showed steady improvement, starting at 0.43 and reaching 0.54 by the 70th epoch, with minimal fluctuations. This highlights the model's enhanced overall performance in object detection tasks across various IoU thresholds.

Figure 6.18: Wheat Head After Detection

As shown in figure 6.18 each wheat heads are labled and detected with high confidence. The number on the box is confidence score and the wheat_head is the label.

6.3.2 Case Study

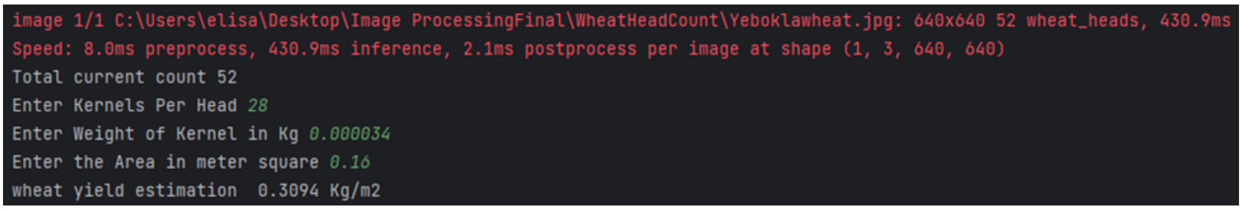
Wheat crops in a field were studied in Yebokla, located near Debre Markos, Ethiopia. Manual counting of heads in a 0.16 m^2 area is 60 heads, with the number of kernels per head being 28 and the weight of 150 kernels being 5.1 grams.

$$\begin{array}{c}\text{Wheat Yield}\\\text{Estimation}\end{array} = \frac{\text{Number of wheat heads} \times \text{Number of kernels per head} \times \text{Weight of each kernel}}{\text{Area}}$$

So:

$$\text{Wheat Yield Estimation} = \frac{60 \times 28 \times 0.000034\,\text{kg}}{0.16\,\text{m}^2} = 0.357\,\text{kg}/\text{m}^2$$

Estimating wheat yield using image processing



```
image 1/1 C:\Users\elisa\Desktop\Image ProcessingFinal\WheatHeadCount\Yeboklawheat.jpg: 640x640 52 wheat_heads, 430.9ms
Speed: 8.0ms preprocess, 430.9ms inference, 2.1ms postprocess per image at shape (1, 3, 640, 640)
Total current count 52
Enter Kernels Per Head 28
Enter Weight of Kernel in Kg 0.000034
Enter the Area in meter square 0.16
wheat yield estimation  0.3094 Kg/m2
```

Estimating wheat yield using image processing for the same area results in 52 wheat head count and a wheat yield estimation of 0.3094 kg/m^2.

Figure 6.19: Wheat Head Before and After Detection

For the 0.16 square meter area, manual calculation from field data of wheat yield estimation produced a result of 0.357 kg/m^2, while image processing resulted in 0.3094 kg/m^2. It's evident that the image processing result for this smaller area is slightly lower than the manual estimation.

The true advantage of the image processing method becomes apparent when considering larger areas, such as entire farms. Here, the AI system excels by meticulously counting each wheat head across expansive fields, ensuring a comprehensive assessment. The scalability and consistent accuracy of image processing make it a powerful tool for large-scale wheat yield estimation, offering the potential for the most reliable and precise results.

Chapter Seven

Conclusion and Recommendation

7.1 Conclusion

This thesis has encompassed a multifaceted exploration, ranging from the modeling of quadcopters using the Newton Euler method to the design and simulation of a Fuzzy STSMC with PID sliding surface in MATLAB/Simulink, and the innovative application of image processing for wheat yield estimation in PyCharm.

The integration of a Fuzzy logic controller has proven to be a pivotal solution to the inherent complexity of tuning STSMC parameters with PID sliding surfaces. By incorporating the super twisting algorithm, Successfully reduced the chattering problem often associated with sliding mode control. Furthermore, the automatic tuning of STSMC parameters considerably lightened the control's computational load, while the addition of an integral term sliding surface substantially reduced steady-state errors. Rigorous testing under various conditions, including different trajectories, disturbances, and parameter variations, demonstrated the exceptional control performance. This system not only adeptly tracked desired trajectories but also rejected disturbance and maintained stability even in the face of parameter variation.

In the realm of image processing, the utilization of 3530 drone-captured images, paired with pre-trained weights through YOLOv8 training, provided outstanding results. With a training mean average precision of 97.1% and a validation mean average precision of 96.9%, this model showed impressive ability in automating wheat head detection and counting, a crucial element in estimating wheat yield. By using this data, in conjunction with kernel weight and the number of kernel heads from the ground, the thesis has presented an efficient approach to estimating wheat yield. This innovation not only drastically reduces the time and effort required for manual counting but also mitigates the potential errors associated with human counting. Moreover, it resolves the challenges posed by low-resolution satellite images, thereby enhancing the accuracy and efficiency of wheat yield estimation.

In summary, this thesis has contributed significantly to the fields of control systems and agricultural technology. The fusion of Fuzzy logic and STSMC with PID sliding surfaces has enhance robustness, reduced chattering, and accelerated state convergence while maintaining stability. Simultaneously, the application of image processing has revolutionized wheat yield estimation, making it faster, more accurate, and less labor-intensive. These findings hold immense promise for future research and practical applications in both control engineering and agriculture, underscoring the potential for innovation in these domains.

7.2 Recommendation

The thesis primarily focuses on the examination and evaluation of a quadcopter designed specifically for wheat yield estimation. Emphasis is placed on assessing the robustness and effectiveness of the controller, with the primary goal of minimizing tracking errors for desired trajectories. Additionally, the thesis explores automatic controller gain tuning methods through the implementation of various flight trajectories.

Fellow researchers are strongly encouraged to proactively implement the quadcopter model and control system on physical hardware. Furthermore, by capturing images using the quadcopter and using the trained system for wheat yield estimation, substantial contributions to the advancement and relevance of agricultural technology can be achieved. This will ultimately benefit both the agricultural industry and the research community.

7.3 Future Work

To enhance the quadcopter's capabilities by adding an autonomous path planning mechanism, which involves the integration of advanced sensors and optimization for energy efficiency. Additionally, extending the system to automate the counting of kernels per wheat head represents a significant advancement, enhancing the precision of crop yield estimation. This development aligns with the main goal of automating wheat yield estimation, thereby making the process more efficient and accurate.

References

[1]. I. Becker-Reshef, E. Vermote, M. Lindeman, et al., "A generalized regression-based model for forecasting winter wheat yields in Kansas and Ukraine using MODIS data," Remote Sensing of Environment, vol. 114, no. 6, pp. 1312-1323, 2010.

[2]. M. U. Liaqat, M. J. M. Cheema, W. Huang, et al., "Evaluation of MODIS and Landsat multiband vegetation indices used for wheat yield estimation in irrigated Indus Basin," in *Comput. Electron. Agric.*, vol. 138, pp. 39-47, 2017.

[3]. P. C. Silvestro, S. Pignatti, S. Pascucci, et al., "Estimating wheat yield in China at the field and district scale from the assimilation of satellite data into the Aquacrop and simple algorithm for yield (SAFY) models," *Remote Sensing*, vol. 9, no. 5, pp. 509, 2017.

[4]. S. Skakun, E. Vermote, J. C. Roger, et al., "Combined use of Landsat-8 and Sentinel-2A images for winter crop mapping and winter wheat yield assessment at regional scale," *AIMS Geosci*, vol. 3, no. 2, pp. 163-186, 2017.

[5]. W. Zhuo, J. Huang, L. Li, et al., "Assimilating soil moisture retrieved from Sentinel-1 and Sentinel-2 data into WOFOST model to improve winter wheat yield estimation," *Remote Sensing*, vol. 11, no. 13, pp. 1-17, 2019.

[6]. Y. Fu, G. Yang, J. Wang, X. Song, and H. Feng, "Winter wheat biomass estimation based on spectral indices, band depth analysis and partial least squares regression using hyperspectral measurements," *Comput. Electron. Agric.*, vol. 100, pp. 51-59, 2014.

[7]. Y. Wang, K. Chang, R. Chen, J. Lo, and Y. Shen, "Large-area rice yield forecasting using satellite imageries," *Int. J. Appl. Earth Obs.*, vol. 12, pp. 27-35, 2010.

[8]. M. Du and N. Noguchi, "Multi-temporal monitoring of wheat growth through correlation analysis of satellite images, unmanned aerial vehicle images with ground variable," in *IFAC-PapersOnLine*, vol. 49, pp. 5-9, 2016.

[9]. J. A. J. Berni, P. J. Zarco-Tejada, L. Suarez, and E. Fereres, "Thermal and narrowband multispectral remote sensing for vegetation monitoring from an unmanned aerial vehicle," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, pp. 722-738, 2009.

[10]. N. Yu, L. Li, N. Schmitz, L. F. Tiaz, J. A. Greenberg, and B. W. Diers, "Development of methods to improve soybean yield estimation and predict plant maturity with an unmanned aerial vehicle-based platform," *Remote Sens. Environ.*, vol. 187, pp. 91-101, 2016.

[11]. C. Zhang and J. M. Kovacs, "The application of small unmanned aerial systems for precision

agriculture: A review," *Precis. Agric.*, vol. 13, pp. 693-712, 2012.

[12]. A. Verger, N. Vigneau, C. Cheron, J. Gilliot, A. Comar, and F. Baret, "Green area index from an unmanned aerial system over wheat and rapeseed crops," *Remote Sens. Environ.*, vol. 152, pp. 654-664, 2014.

[13]. S. Candiago, F. Remondino, M. De Giglio, M. Dubbini, and M. Gattelli, "Evaluating multispectral images and vegetation indices for precision farming applications from UAV images," *Remote Sens.*, vol. 7, pp. 4026-4047, 2015.

[14]. S. G. Gupta, M. M. Ghonge, P. M. Jawandhiya, et al., "Review of unmanned aircraft system (UAS)," *Int. J. Adv. Res. Comput. Eng. Technol*, vol. 2, no. 4, pp. 1646-1658, 2013.

[15]. R. Tesfaye, "Modeling and Control of a Quad-rotor Unmanned Aerial Vehicle at Hovering Position," Ph.D. thesis, Addis Ababa University, 2012.

[16]. V. I. Utkin, "Sliding Mode Control: Mathematical Tools, Design and Applications," Lecture note, 2004.

[17]. C. A. Reynolds, M. Yitayew, D. C. Slack, C. F. Hutchinson, A. Huete, and M. S. Petersen, "Estimating crop yields and production by integrating the FAO crop-specific water balance model with real-time satellite data and ground-based ancillary data," *IEEE Trans. Geosci. Remote Sens.*, vol. 21, pp. 3487-3508, 2000.

[18]. H. Greatrex, "The Application of Seasonal Rainfall Forecasts and Satellite Rainfall Estimates to Seasonal Crop Yield Forecasting for Africa," Ph.D. dissertation, University of Reading, Reading, UK, 2012.

[19]. Y. Shtessel, F. Plestan, and M. Taleb, "Lyapunov design of adaptive super-twisting controller applied to a pneumatic actuator," *in Proc. IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 3051-3056, 2011.

[20]. 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.

[21]. V. I. Utkin and A. S. Poznyak, "Adaptive sliding mode control with application to super-twist algorithm: equivalent control method," *Automatica*, vol. 49, no. 1, pp. 39-47, 2013.

[22]. K. Lu and Y. Xia, "Finite-time attitude control for rigid spacecraft based on adaptive super-twisting algorithm," in IET Control Theory and Applications, vol. 8, no. 15, pp. 1465-1477, 2014.

[23]. C.-C. Peng and C.-L. Chen, "Robust chaotic control of Lorenz system by backstepping design," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 55, no. 10, pp. 3220-3228, 2008.

[24]. C.-L. Chen, C. C. Peng, and H.-T. Yau, "High-order sliding mode controller with backstepping design for aeroelastic systems," *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 4, pp. 1813-1823, 2012.

[25]. L.-H. Chen and C.-C. Peng, "Extended backstepping sliding controller design for chattering attenuation and its application for servo motor control," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 3, pp. 1362-1370, 2017.

[26]. A. Levant, "Sliding order and sliding accuracy in sliding mode control," *IEEE Trans. Autom. Control*, vol. 58, no. 6, pp. 1247-1263, 1993.

[27]. S. Bouabdallah A. Noth and R. Siegwart. "PID vs LQ control techniques applied to an indoor micro quadrotor," Intelligent Robots and Systems, 2004.

[28]. S. Bouabdallah and R. Y. Siegwart, "Full control of a quadrotor," in *Proc. of International Conference on Intelligent Robots and Systems, IROS 2007*, San Diego, CA, USA, IEEE, pp. 153-158, Oct. 29-Nov. 2, 2007.

[29]. X. Shi, Y. Cheng, C. Yin, S. Dadras, and X. Huang, "Design of fractional-order backstepping sliding mode control for quadrotor UAV," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 1, pp. 156-171, January 2019.

[30].W. Qi, G. Zong, and H. Karimi, "Sliding mode control for nonlinear stochastic singular semi-Markov jump systems," *IEEE Trans. Autom. Control*, vol. 65, no. 1, pp. 361-368, January 2020.

[31]. W. Qi, G. Zong, and H. R. Karim, "Observer-based adaptive SMC for nonlinear uncertain singular semi-Markov jump systems with applications to DC motor," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 65, no. 9, pp. 2951-2960, February 2018.

[32]. A. Levant, "Sliding order and sliding accuracy in sliding mode control," *IEEE Transactions on Automatic Control*, vol. 58, no. 6, pp. 1247-1263, 1993.

[33]. V. Utkin, "Discussion aspects of high-order sliding mode control," *IEEE Transactions on Automatic Control*, vol. 61, no. 3, pp. 829-833, 2015.

[34]. T. Huang, D. Huang, Z. Wang, and A. Shah, "Robust tracking control of a quadrotor UAV based on adaptive sliding mode controller," *Complexity*, vol. 2019, 2019.

[35]. M. Rabah, A. Rohan, S. A. S. Mohamed, and S. H. Kim, "Autonomous Moving Target-Tracking for a UAV Quadcopter Based on Fuzzy-PI," *IEEE Access*, vol. 7, pp. 38407-38419, 2019.

[36]. Q. Feng and S. Wu, "Processing in crop yield estimating by remote sensing in China," *World*

Sci-Tech RD, vol. 28, no. 3, pp. 32-36, 2006, doi: 10.3969/j.issn.1006-6055.2006.03.006.

[37]. K. R. Thorp, K. C. DeJonge, A. L. Kaleita, W. D. Batchelor, and J. O. J. C. Paz, "Methodology for the use of DSSAT models for precision agriculture decision support," *Comput. Electron Agric.*, vol. 64, pp. 276-285, 2008, doi: 10.1016/j.compag.2008.05.022.

[38]. J. Betbeder, F. Rémy, and F. Baup, "Assimilation of LAI and dry biomass data from optical and SAR images into an agro-meteorological model to estimate soybean yield," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 9, pp. 1-14, 2016, doi: 10.1109/JSTARS.2016.2541169.

[39]. P. K. Kingra, D. Majumder, and S. P. Singh, "Application of remote sensing and GIS in agriculture and natural resource management under changing climatic conditions," Agricultural Research Journal, vol. 53, no. 3, pp. 2016.

[40]. M. Qiao, X. He, X. Cheng, P. Li, H. Luo, L. Zhang, et al., "Crop yield prediction from multi-spectral, multi-temporal remotely sensed imagery using recurrent 3D convolutional neural networks," *Int. J. Appl. Earth Obs. Geoinf.*, vol. 102, p. 102436, 2021, doi: 10.1016/j.jag.2021.102436.

[41]. B. Zhang, Y. Wu, B. Zhao, J. Chanussot, D. Hong, J. Yao, et al., "Progress and challenges in intelligent remote sensing satellite systems," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 15, pp. 1814-1822, 2022, doi: 10.1109/jstars.2022.3148139.

[42]. X. Wang, J. Huang, Q. Feng, and D. Yin, "Winter wheat yield prediction at county level and uncertainty analysis in main wheat-producing regions of China with deep learning approaches," *Remote Sens.*, vol. 12, no. 11, p. 1744, 2020, doi: 10.3390/rs12111744.

[43]. E. Kamir, F. Waldner, and Z. Hochman, "Estimating wheat yields in Australia using climate records, satellite image time series and machine learning methods," *ISPRS J. Photogramm. Remote Sens.*, vol. 160, pp. 124-135, 2020.

[44]. R. L. F. Cunha and B. Silva, "Estimating crop yields with remote sensing and deep learning," in *2020 IEEE Latin American GRSS ISPRS Remote Sensing Conference (LAGIRS)*, 2020, pp. 59-64, doi: 10.5194/isprs-annals-IV-3-W2-2020-59-2020.

[45]. D. E. David, S. Madec, P. Sadeghi-Tehran, H. Aasen, B. Zheng, S. Liu, N. Kirchgessner, G. Ishikawa, K. Nagasawa, M. A. Badhon, and C. Pozniak, "Global Wheat Head Detection (GWHD) dataset: a large and diverse dataset of high-resolution RGB-labelled images to develop and benchmark wheat head detection methods," *Plant Phenomics*, Aug 20, 2020.

[46]. B. Gong, D. Ergu, Y. Cai, and B. Ma, "Real-time detection for wheat head applying deep neural network," *Sensors*, vol. 21, no. 1, p. 191, Dec 30, 2020.

[47]. D. J. Tritton, *Physical Fluid Dynamics*, Van Nostrand Reinhold, 1977 (reprinted 1980),

Section 22.7, "The Coandă Effect."

[48]. (2019). "How is lift generated due to Coandă effect?" *Physics Stack Exchange.* [Online]. Available: https://physics.stackexchange.com/questions/278120/how-is-lift-generated-due-to-coanda-effect.

[49]. T. Editors of Encyclopaedia Britannica, "Propeller," *Encyclopedia Britannica*, Jul. 20, 2012. [Online]. Available: https://www.britannica.com/technology/propeller.

[50]. A. Persson, "How Do We Understand the Coriolis Force?," *Bulletin of the American Meteorological Society*, vol. 79, no. 7, pp. 1373-1386, Jul. 1998, `doi:10.1175/1520-0477(1998)079<1373:HDWUTC>2.0.CO;2`.

[51]. Z. Tahir, W. Tahir, and S. A. Liaqat, "State space system modelling of a quadcopter UAV," arXiv preprint arXiv:1908.07401, Aug 20, 2019.

[52]. S. Ullah, A. Mehmood, Q. Khan, S. Rehman, and J. Iqbal, "Robust integral sliding mode control design for stability enhancement of under-actuated quadcopter," *International Journal of Control, Automation and Systems*, vol. 18, pp. 1671-1678, Jul. 2020.

[53]. S. Rao, "Some Issues in the Sliding Mode Control of Rigid Robotic Manipulators," Master of Science (Computer Science) thesis, School of Information Technology and Mathematics, Edith Cowan University, Perth, Western Australia, Oct. 1995.

[54]. F. Moldoveanu, "Sliding Mode Controller Design for Robot Manipulators," *Bulletin of the Transilvania University of Braşov, Series I: Engineering Sciences*, vol. 7, no. 2, 2014.

[55]. J. Wang et al., "Indirect adaptive fuzzy sliding mode control: Part I: fuzzy switching," *Fuzzy Sets and Systems*, vol. 122, no. 1, Aug. 16, 2001.

[56]. F. Ashrafzadeh, E. P. Nowicki, R. Boozarjomehry, and J. C. Salmon, "Optimal Synthesis of Fuzzy Sliding Mode Controllers," *Proceedings of IEEE*, 1996.

[57]. S. Spurgeon, "Sliding mode control: a tutorial," in *2014 European Control Conference (ECC)*, Jun 24, 2014, pp. 2272-2277.

[58]. I. S. Attah, S. U. Hussien, O. A. Agboola, L. O. Inumoh, H. D. Ibrahim, and M. C. Chukwu, "Neuro-Super Twisting Sliding Mode Control of a Direct Current Motor," in *2021 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, Dec 9, 2021, pp. 1-6.

[59]. L. Tan, J. Gao, Y. Luo, and L. Zhang, "Super-twisting sliding mode control with defined boundary layer for chattering reduction of permanent magnet linear synchronous motor," *Journal of Mechanical Science and Technology*, vol. 35, pp. 1829-1840, May 2021.

[60]. C. C. Lee, "Fuzzy logic in control systems: fuzzy logic controller. I," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 2, pp. 404-418, Mar 1990.

[61]. K. M. Passino, *Fuzzy Control*, 1st ed. Columbus, OH, USA: Department of Electrical Engineering, the Ohio State University, 1998.

[62]. Y. Bai and D. Wang, "Fundamentals of fuzzy logic control—fuzzy sets, fuzzy rules and defuzzifications," in *Advanced fuzzy logic technologies in industrial applications*, 2006, pp. 17-36.

[63]. R. Jiao, W. Chou, Y. Rong, and M. Dong, "Anti-disturbance control for quadrotor UAV manipulator attitude system based on fuzzy adaptive saturation super-twisting sliding mode observer," *Applied Sciences*, vol. 10, no. 11, p. 3719, May 27, 2020.

[64]. "Turnigy D2205-2300KV 28G Brushless Motor CW," Hobbyking, [Online]. Available: `https://hobbyking.com/en_us/brushless-motor-d2205-2300kv-cw.html?___store=en_us`.

[65]. "GoPro Hero9 Black review," *Tom's Guide*, [Online]. Available: `https://www.tomsguide.com/reviews/gopro-hero9-black`.

[66]. G. Boesch, "Image Recognition: The Basics and Use Cases (2023 Guide)," *viso.ai*, [Online]. Available: https://viso.ai/computer-vision/image-recognition/.

[67]. A. Germanov, "How to detect objects in images using the YOLOV8 neural Network," *freeCodeCamp.org*, [Online]. Available: https://www.freecodecamp.org/news/how-to-detect-objects-in-images-using-yolov8/.

[68]. "PyCharm: the Python IDE for Professional Developers by JetBrains," *JetBrains*, [Online]. Available: https://www.jetbrains.com/pycharm/.

[69]. "Image annotation tools to get your labeling project started," *Datagen*, [Online]. Available: https://datagen.tech/guides/image-annotation/image-annotation-tool/#.

Appendices

A Euler Angle Operations

A.1 Rotation Matrix Derivation

The rotation matrix is found by calculating the projection of one axis on the other.

Rotation about the Z-axis

$$\text{Rotation matrix about the Z-axis: } \begin{bmatrix} X_0X_I & Y_0X_I & Z_0X_I \\ X_0Y_I & Y_0Y_I & Z_0Y_I \\ X_0Z_I & Y_0Z_I & Z_0Z_I \end{bmatrix}$$

Where: X_0X_I is the Projection of X_0 on X_I, X_0Y_I is the Projection of X_0 on Y_I, X_0Z_I is the Projection of X_0 on Z_I, Y_0X_I is the Projection of Y_0 on X_I, Y_0Y_I is the Projection of Y_0 on Y_I, Y_0Z_I is the Projection of Y_0 on Z_I, Z_0X_I is the Projection of Z_0 on X_I, Z_0Y_I is the Projection of Z_0 on Y_I, Z_0Z_I is the Projection of Z_0 on Z_I.

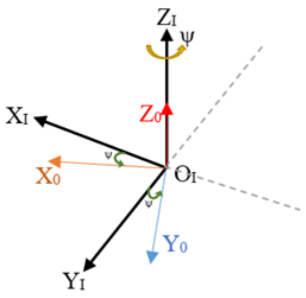
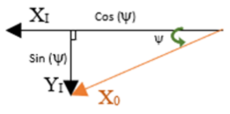
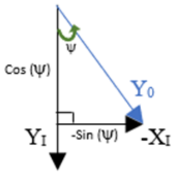


Figure A.1: Rotation about Z axis





$$X_0X_I = \cos(\psi) \qquad Y_0X_I = -\sin(\psi)$$

$$X_0Y_I = \sin(\psi) \qquad Y_0Y_I = \cos(\psi)$$

$$X_0Z_I = 0 \qquad Y_0Z_I = 0$$

$Z_0X_I = Z_0Y_I = 0$ and $Z_0Z_I = 1$ because Z_0 is perpendicular to X_I and Y_I so the projection of Z_I on X_I and Y_I are zero. The length of Z_0 is one, and it lines up exactly with Z_I. The projection of Z_0 on Z_I is one.

$$[R_{Z,\psi}] = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} X_I & Y_I & Z_I \end{bmatrix}^T = [R_{Z,\psi}] \begin{bmatrix} X_0 & Y_0 & Z_0 \end{bmatrix}^T$$

Rotation about Y axis

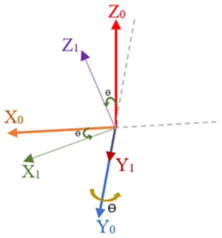
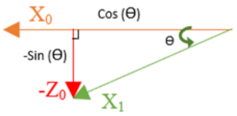
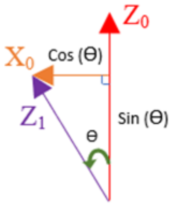


Figure A.2: Rotation about Y axis





$$X_1X_0 = \cos(\theta) \qquad Z_1X_0 = \sin(\theta)$$

$$X_1Y_0 = 0 \qquad Z_1Y_0 = 0$$

$$X_1Z_0 = -\sin(\theta) \qquad Z_1Z_0 = \cos(\theta)$$

$$Y_1X_0 = Y_1Z_0 \quad = 0 \text{ and } Y_1Y_0 = 1$$

$$[R_{Y,\theta}] = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$\begin{bmatrix} X_0 & Y_0 & Z_0 \end{bmatrix}^T = [R_{Y,\theta}] \begin{bmatrix} X_1 & Y_1 & Z_1 \end{bmatrix}^T$$

Rotation about X axis

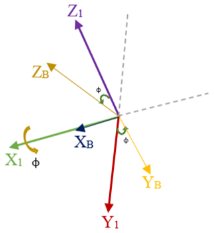
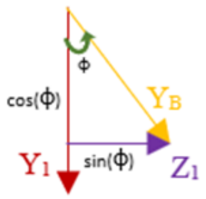
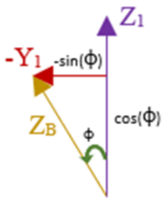


Figure A.3: Rotation about X axis





$$Y_B X_1 = 0 \qquad\qquad Z_B X_1 = 0$$

$$Y_B Y_1 = \cos(\phi) \qquad\qquad Z_B Y_1 = -\sin(\phi)$$

$$Y_B Z_1 = \sin(\phi) \qquad\qquad Z_B Z_1 = \cos(\phi)$$

$$X_B Y_1 = X_B Z_1 = 0 \text{ and } X_B X_1 \quad = 1$$

$$[R_{x,\phi}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}$$

$$\begin{bmatrix} X_1 & Y_1 & Z_1 \end{bmatrix}^T = [R_{x,\phi}] \begin{bmatrix} X_B & Y_B & Z_B \end{bmatrix}^T$$

Multiplying the 3 rotation matrices: ${}^B_I R = R_{z,\psi} R_{y,\theta} R_{x,\phi}$

$${}^B_I R = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

$${}^B_I R = \begin{bmatrix} c(\theta)c(\psi) & c(\theta)s(\psi) & -s(\theta) \\ s(\phi)s(\theta)c(\psi) - c(\phi)s(\psi) & c(\theta)c(\psi) + s(\phi)s(\theta)s(\psi) & s(\phi)c(\theta) \\ s(\phi)s(\psi) + c(\phi)s(\theta)c(\psi) & -s(\phi)c(\psi) + c(\phi)s(\theta)s(\psi) & c(\phi)c(\theta) \end{bmatrix}$$

$$({}^B_I R)^T = {}^I_B R = \begin{bmatrix} c(\theta)c(\psi) & s(\phi)s(\theta)c(\psi) - c(\phi)s(\psi) & s(\phi)s(\psi) + c(\phi)s(\theta)c(\psi) \\ c(\theta)s(\psi) & c(\theta)c(\psi) + s(\phi)s(\theta)s(\psi) & -s(\phi)c(\psi) + c(\phi)s(\theta)s(\psi) \\ -s(\theta) & s(\phi)c(\theta) & c(\phi)c(\theta) \end{bmatrix}$$

A.2 Transfer Matrix Derivation

The rotation is first rotation about the Z axis, then about the Y axis, and then about the X axis. Since the last rotation is about the X axis, X_B and X_1 line up exactly, so the roll rate vector is $\begin{bmatrix} \dot{\phi} & 0 & 0 \end{bmatrix}^T$. To calculate the pitch rate vector, the X axis rotation must be rotated back, so pitch rate vector is $R_{X,\phi}^{-1} \begin{bmatrix} 0 & \dot{\theta} & 0 \end{bmatrix}^T$. To find yaw rate vector, X axis rotation and Y axis rotation must be rotated back, so yaw rate vector is $R_{X,\phi}^{-1} R_{Y,\theta}^{-1} \begin{bmatrix} 0 & 0 & \dot{\psi} \end{bmatrix}^T$.

$$\begin{bmatrix}\Omega_\phi\\ \Omega_\theta\\ \Omega_\psi\end{bmatrix}=\begin{bmatrix}\dot{\phi}\\ 0\\ 0\end{bmatrix}+\left(R_{x,\phi}\right)^{-1}\begin{bmatrix}0\\ \dot{\theta}\\ 0\end{bmatrix}+\left(R_{x,\phi}\right)^{-1}\left(R_{y,\theta}\right)^{-1}\begin{bmatrix}0\\ 0\\ \dot{\psi}\end{bmatrix}$$

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

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

$$\begin{bmatrix}\Omega_\phi\\ \Omega_\theta\\ \Omega_\psi\end{bmatrix}=\begin{bmatrix}-\sin\theta\dot{\psi}\\ \sin\phi\cos\theta\dot{\psi}\\ \cos\phi\cos\theta\dot{\psi}\end{bmatrix}+\begin{bmatrix}0\\ \cos\phi\dot{\theta}\\ -\sin\phi\dot{\theta}\end{bmatrix}+\begin{bmatrix}\dot{\phi}\\ 0\\ 0\end{bmatrix}$$

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

$$T^{-1}=\begin{bmatrix}1 & 0 & -\sin\theta\\ 0 & \cos\phi & \sin\phi\cos\theta\\ 0 & -\sin\phi & \cos\phi\cos\theta\end{bmatrix},T=\begin{bmatrix}1 & \sin\phi\tan\theta & \cos\phi\tan\theta\\ 0 & \cos\phi & -\sin\phi\\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta}\end{bmatrix}$$

B Derivation of Coriolis Force

Using Newton-Euler method, Euler's 1st axiom of Newton's 2nd law $F = ma$, in the body reference frame:

$\sum \vec{F}^B = m\frac{d\mathbf{v}}{dt}, \quad \mathbf{v} = v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k}$

Let $\mathbf{v} = u\mathbf{i_x} + v\mathbf{j_y} + w\mathbf{k_z}$ represent linear velocity in the inertial frame of reference, and $\mathbf{v} = v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k}$ represent linear velocity in the body reference frame.

The derivative of $\mathbf{v}$ in the inertial frame of reference is:

$$\frac{d\mathbf{v}}{dt} = \frac{du}{dt}\mathbf{i_x} + \frac{dv}{dt}\mathbf{j_y} + \frac{dw}{dt}\mathbf{k_z} + \frac{di_x}{dt}u + \frac{dj_y}{dt}v + \frac{dk_z}{dt}w$$

$\frac{di_x}{dt} = \frac{dj_y}{dt} = \frac{dk_z}{dt} = 0$ because the coordinate axes are in the inertial frame so they are constant. There fore:

$$\frac{d\mathbf{v}}{dt} = \frac{du}{dt}\mathbf{i_x} + \frac{dv}{dt}\mathbf{j_y} + \frac{dw}{dt}\mathbf{k_z}$$

The derivative of $\mathbf{v}$ in the body frame of reference is:

$$\frac{d\mathbf{v}}{dt} = \frac{dv_x}{dt}\mathbf{i} + \frac{dv_y}{dt}\mathbf{j} + \frac{dv_z}{dt}\mathbf{k} + \frac{di}{dt}v_x + \frac{dj}{dt}v_y + \frac{dz}{dt}v_z$$

Equating the linear velocity in the two frames of reference:

$$\frac{du}{dt}\mathbf{i_x} + \frac{dv}{dt}\mathbf{j_y} + \frac{dw}{dt}\mathbf{k_z} = \frac{dv_x}{dt}\mathbf{i} + \frac{dv_y}{dt}\mathbf{j} + \frac{dv_z}{dt}\mathbf{k} + \frac{di}{dt}v_x + \frac{dj}{dt}v_y + \frac{dz}{dt}v_z$$

$$\left(\frac{du}{dt}\mathbf{i}_x + \frac{dv}{dt}\mathbf{j}_y + \frac{dw}{dt}\mathbf{k}_z\right) = \left(\frac{dv_x}{dt}\mathbf{i} + \frac{dv_y}{dt}\mathbf{j} + \frac{dv_z}{dt}\mathbf{k}\right) + \left(\frac{di}{dt}v_x + \frac{dj}{dt}v_y + \frac{dz}{dt}v_z\right)$$

$$\left(\frac{du}{dt}\mathbf{i}_x + \frac{dv}{dt}\mathbf{j}_y + \frac{dw}{dt}\mathbf{k}_z\right)_{\text{I-frame}} = \left(\frac{dv_x}{dt}\mathbf{i} + \frac{dv_y}{dt}\mathbf{j} + \frac{dv_z}{dt}\mathbf{k}\right)_{\text{B-frame}} + \text{Effect of rotation}$$

Since, linear velocity v is equal to angular velocity ω times the position vector r, i.e., $v = \omega \times r$, and linear velocity is the derivative of position with respect to time, $v = \frac{dr}{dt}$. In this case, $r = \mathbf{i}, \mathbf{j}, \mathbf{k}$ and $\omega = \Omega_\phi, \Omega_\theta, \Omega_\psi$. So, $v = \frac{dr}{dt} = \boldsymbol{\omega} \times r$. Thus:

$$\frac{di}{dt} = \boldsymbol{\omega} \times i, \quad \frac{dj}{dt} = \boldsymbol{\omega} \times j, \quad \frac{dk}{dt} = \boldsymbol{\omega} \times k$$

$$\frac{d\mathbf{v}}{dt} = \frac{dv_x}{dt}\mathbf{i} + \frac{dv_y}{dt}\mathbf{j} + \frac{dv_z}{dt}\mathbf{k} + (\boldsymbol{\omega} \times \mathbf{i})v_x + (\boldsymbol{\omega} \times \mathbf{j})v_y + (\boldsymbol{\omega} \times \mathbf{k})v_z$$

$$\sum \vec{F}^B = m\frac{d\mathbf{v}}{dt} = m\left(\frac{dv_x}{dt}\mathbf{i} + \frac{dv_y}{dt}\mathbf{j} + \frac{dv_z}{dt}\mathbf{k}\right) + m\Big((\boldsymbol{\omega} \times \mathbf{i})v_x + (\boldsymbol{\omega} \times \mathbf{j})v_y + (\boldsymbol{\omega} \times \mathbf{k})v_z\Big)$$

$m(\boldsymbol{\omega} \times \mathbf{i})v_x + (\boldsymbol{\omega} \times \mathbf{j})v_y + (\boldsymbol{\omega} \times \mathbf{k})v_z$ is the Coriolis force.

Coriolis force = m$(\boldsymbol{\omega}^B \times (v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k}))$

Coriolis force = m$\left(\begin{bmatrix} \Omega_\phi \\ \Omega_\theta \\ \Omega_\psi \end{bmatrix} \times \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}\right)$

The cross product $\boldsymbol{\omega}^B \times \mathbf{V}^B$ is calculated as:

$$\boldsymbol{\omega}^B \times \mathbf{V}^B = \mathbf{i}(\Omega_\theta v_z - \Omega_\psi v_y) - \mathbf{j}(\Omega_\phi v_z - \Omega_\psi v_x) + \mathbf{k}(\Omega_\phi v_y - \Omega_\theta v_x)$$

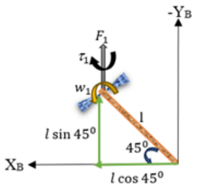
$$\begin{aligned} \vec{F}^B_x &= m\left(\frac{dv_x}{dt} + \Omega_\theta v_z - \Omega_\psi v_y\right) \\ \vec{F}^B_y &= m\left(\frac{dv_y}{dt} + \Omega_\psi v_x - \Omega_\phi v_z\right) \\ \vec{F}^B_z &= m\left(\frac{dv_z}{dt} + \Omega_\phi v_y - \Omega_\theta v_x\right) \end{aligned}$$

$$\sum \vec{F}^B = \begin{bmatrix} m(a_x + \Omega_\theta v_z - \Omega_\psi v_y) \\ m(a_y + \Omega_\psi v_x - \Omega_\phi v_z) \\ m(a_z + \Omega_\phi v_y - \Omega_\theta v_x) \end{bmatrix}$$

C Torque Analysis

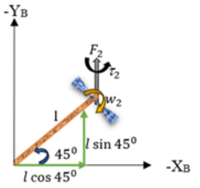
C.1 Torque Derivation

τ_1: Analysis



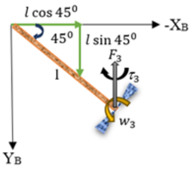
$$\tau_{x1} = \vec{F}_1 l \cos 45^\circ, \quad \tau_{y1} = -\vec{F}_1 l \sin 45^\circ, \quad \tau_{z1} = -\tau_1$$

τ_2: Analysis



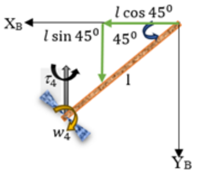
$$\tau_{x2} = -\vec{F}_2 l \cos 45^\circ, \quad \tau_{y2} = -\vec{F}_2 l \sin 45^\circ, \quad \tau_{z2} = \tau_2$$

τ_3: Analysis



$$\tau_{x3} = -\vec{F}_3 l \cos 45^\circ, \quad \tau_{y3} = \vec{F}_3 l \sin 45^\circ, \quad \tau_{z3} = -\tau_3$$

τ_4: Analysis



$$\tau_{x4} = \vec{F}_4 l \cos 45^\circ, \quad \tau_{y4} = \vec{F}_4 l \sin 45^\circ, \quad \tau_{z4} = \tau_4$$

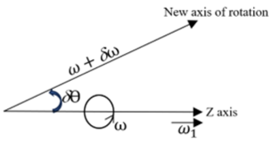
$\sum \tau_x^B = \tau_{x1} + \tau_{x2} + \tau_{x3} + \tau_{x4} = \vec{F}_1\, l \cos(45^\circ) - \vec{F}_2\, l \cos(45^\circ) - \vec{F}_3\, l \cos(45^\circ) + \vec{F}_4\, l \cos(45^\circ) = \frac{l\sqrt{2}}{2}(\vec{F}_1 - \vec{F}_2 - \vec{F}_3 + \vec{F}_4)$

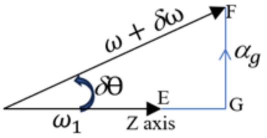
$\sum \tau_y^B = \tau_{y1} + \tau_{y2} + \tau_{y3} + \tau_{y4} = -\vec{F}_1\, l \sin(45^\circ) - \vec{F}_2\, l \sin(45^\circ) + \vec{F}_3\, l \sin(45^\circ) + \vec{F}_4\, l \sin(45^\circ) = \frac{l}{\sqrt{2}}(-\vec{F}_1 - \vec{F}_2 + \vec{F}_3 + \vec{F}_4)$

$\sum \tau_z^B = -\tau_1 + \tau_2 - \tau_3 + \tau_4$

C.2 Gyroscopic Effect Derivation

To find gyroscopic acceleration





From point E to point F there is change in velocity so there is angular acceleration. Assume at point G there is zero angular acceleration.

$$\alpha_g = \lim_{\delta t \to 0} \left(\frac{(\omega + \delta_\omega) \sin \delta_\theta - 0}{\delta t} \right) = \lim_{\delta t \to 0} \left(\frac{\omega \sin \delta_\theta + \delta_\omega \sin \delta_\theta}{\delta t} \right)$$

Since sin of very small angle is the angle itself $\sin(\delta\theta) \to \delta\theta$.

$$\alpha_g = \lim_{\delta t \to 0} \left(\frac{\omega\delta_\theta + \delta\omega\delta_\theta}{\delta t} \right)$$

By neglecting $\delta_\omega \delta_\theta$ because it is a very small number, $\alpha_g = \lim_{\delta t \to 0} \frac{\omega\delta_\theta}{\delta t} = \omega \frac{d_\theta}{dt} = \omega\Omega$. Where: Ω is the angular rate of change of the plane of rotation of the propeller and ω is the propeller angular velocity.

$$\tau_g = J\alpha_g = J(\omega \times \Omega)$$

Where $\omega = \begin{bmatrix} 0 & 0 & \omega \end{bmatrix}^T$ and $\Omega = \begin{bmatrix} \Omega_\phi & \Omega_\theta & \Omega_\psi \end{bmatrix}^T$.

In the horizontal plane of quadcopter there is four propellers and each of them contributes this gyroscopic torque.

$$\tau_{g1} = J \left(\begin{bmatrix} 0 \\ 0 \\ \omega_1 \end{bmatrix} \times \begin{bmatrix} \Omega_\phi \\ \Omega_\theta \\ \Omega_\psi \end{bmatrix} \right)$$

The cross product of ω and Ω is calculated as:

$$\omega \times \Omega = -(\Omega_\theta \omega)i - (-\Omega_\phi \omega)j + 0k = \begin{bmatrix} -\Omega_\theta \omega & \Omega_\phi \omega & 0 \end{bmatrix}^T$$

$$\tau_{g1} = J \begin{bmatrix} -\Omega_\theta \omega_1 & \Omega_\phi \omega_1 & 0 \end{bmatrix}$$

$$\tau_{g2} = J \left(\begin{bmatrix} 0 \\ 0 \\ -\omega_2 \end{bmatrix} \times \begin{bmatrix} \Omega_\phi \\ \Omega_\theta \\ \Omega_\psi \end{bmatrix} \right) = J \begin{bmatrix} \Omega_\theta \omega_2 \\ -\Omega_\phi \omega_2 \\ 0 \end{bmatrix}$$

$$\tau_{g3} = J \left(\begin{bmatrix} 0 \\ 0 \\ \omega_3 \end{bmatrix} \times \begin{bmatrix} \Omega_\phi \\ \Omega_\theta \\ \Omega_\psi \end{bmatrix} \right) = J \begin{bmatrix} -\Omega_\theta \omega_3 \\ \Omega_\phi \omega_3 \\ 0 \end{bmatrix}$$

$$\tau_{g4} = J\left(\begin{bmatrix} 0 \\ 0 \\ -\omega_4 \end{bmatrix} \times \begin{bmatrix} \Omega_\phi \\ \Omega_\theta \\ \Omega_\psi \end{bmatrix}\right) = J\begin{bmatrix} \Omega_\theta\omega_4 \\ -\Omega_\phi\omega_4 \\ 0 \end{bmatrix}$$

$$\tau_{g\ \text{total}} = \tau_{g1} + \tau_{g2} + \tau_{g3} + \tau_{g4}$$

$$\tau_{g\ \text{total}} = J\begin{bmatrix} \Omega_\theta(-\omega_1 + \omega_2 - \omega_3 + \omega_4) \\ \Omega_\phi(\omega_1 - \omega_2 + \omega_3 - \omega_4) \\ 0 \end{bmatrix}$$

$$\tau_{g\ \text{total}} = J\begin{bmatrix} \Omega_\theta(\omega_r) & \Omega_\phi(-\omega_r) & 0 \end{bmatrix}^T, \text{Let } \omega_r = -\omega_1 + \omega_2 - \omega_3 + \omega_4$$

C.3 Euler Equation

$$\tau^B = \boldsymbol{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\boldsymbol{I}\boldsymbol{\omega}) = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}\begin{bmatrix} \dot{\Omega}_\phi \\ \dot{\Omega}_\theta \\ \dot{\Omega}_\psi \end{bmatrix} + \begin{bmatrix} \Omega_\phi \\ \Omega_\theta \\ \Omega_\psi \end{bmatrix} \times \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}\begin{bmatrix} \Omega_\phi \\ \Omega_\theta \\ \Omega_\psi \end{bmatrix}$$

$$\tau^B = \begin{bmatrix} I_{xx}\dot{\Omega}_\phi \\ I_{yy}\dot{\Omega}_\theta \\ I_{zz}\dot{\Omega}_\psi \end{bmatrix} + \begin{bmatrix} \Omega_\phi \\ \Omega_\theta \\ \Omega_\psi \end{bmatrix} \times \begin{bmatrix} I_{xx}\Omega_\phi \\ I_{yy}\Omega_\theta \\ I_{zz}\Omega_\psi \end{bmatrix}$$

The cross product of ω and $I\omega$ is calculated as:

$$\boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} = (\Omega_\theta I_{zz}\Omega_\psi - I_{yy}\Omega_\theta\Omega_\psi)\mathbf{i} - (\Omega_\phi I_{zz}\Omega_\psi - I_{xx}\Omega_\phi\Omega_\psi)\mathbf{j} + (\Omega_\phi I_{yy}\Omega_\theta - I_{xx}\Omega_\phi\Omega_\theta)\mathbf{k}$$

$$\begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} I_{xx}\dot{\Omega}_\phi \\ I_{yy}\dot{\Omega}_\theta \\ I_{zz}\dot{\Omega}_\psi \end{bmatrix} + \begin{bmatrix} \Omega_\theta\Omega_\psi(I_{zz} - I_{yy}) \\ \Omega_\phi\Omega_\psi(I_{xx} - I_{zz}) \\ \Omega_\phi\Omega_\theta(I_{yy} - I_{xx}) \end{bmatrix} = \begin{bmatrix} I_{xx}\dot{\Omega}_\phi + \Omega_\theta\Omega_\psi(I_{zz} - I_{yy}) \\ I_{yy}\dot{\Omega}_\theta + \Omega_\phi\Omega_\psi(I_{xx} - I_{zz}) \\ I_{zz}\dot{\Omega}_\psi + \Omega_\phi\Omega_\theta(I_{yy} - I_{xx}) \end{bmatrix}$$

$$\sum \tau = \tau_T + \tau_g = \boldsymbol{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\boldsymbol{I}\boldsymbol{\omega})$$

$$\sum \tau = \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} + J\begin{bmatrix} \Omega_\theta(\omega_r) \\ \Omega_\phi(-\omega_r) \\ 0 \end{bmatrix} = \begin{bmatrix} I_{xx}(\dot{\Omega}_\phi) + \Omega_\theta\Omega_\psi(I_{zz} - I_{yy}) \\ I_{yy}(\dot{\Omega}_\theta) + \Omega_\phi\Omega_\psi(I_{xx} - I_{zz}) \\ I_{zz}(\dot{\Omega}_\psi) + \Omega_\phi\Omega_\theta(I_{yy} - I_{xx}) \end{bmatrix}$$

$$\begin{bmatrix} \dot{\Omega}_\phi \\ \dot{\Omega}_\theta \\ \dot{\Omega}_\psi \end{bmatrix} = \begin{bmatrix} (U_2 + J\Omega_\theta(\omega_r) - (\Omega_\theta\Omega_\psi(I_{zz} - I_{yy}))/I_{xx}) \\ (U_3 + J\Omega_\phi(-\omega_r) - (\Omega_\phi\Omega_\psi(I_{xx} - I_{zz}))/I_{yy}) \\ (U_4 - (\Omega_\phi\Omega_\theta(I_{yy} - I_{xx}))/I_{zz}) \end{bmatrix}$$

D Desired Roll and Pitch Angles

Define:

$$\ddot{X} = (\cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi))\left(\tfrac{U_1}{m}\right)$$
$$\ddot{Y} = (\cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi))\left(\tfrac{U_1}{m}\right)$$
$$\ddot{Z} = (\cos(\phi)\cos(\theta))\left(\tfrac{U_1}{m}\right) - g$$

From $\ddot{X}$:

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

$$\left(\frac{\ddot{X}m}{U_1} - \sin(\phi)\sin(\psi)\right)\frac{1}{\cos(\phi)\cos(\psi)} = \sin(\theta)$$

$$\theta_d = \sin^{-1}\left[\left(\frac{\ddot{X}m}{U_1} - \sin(\phi)\sin(\psi)\right)\frac{1}{\cos(\phi)\cos(\psi)}\right]$$

From $\ddot{X}$ and $\ddot{Y}$:

$$\left(\frac{\ddot{X}m}{U_1} - \sin(\phi)\sin(\psi)\right)\frac{1}{\cos(\psi)} = \cos(\phi)\sin(\theta)$$

$$\left(\frac{\ddot{Y}m}{U_1} + \sin(\phi)\cos(\psi)\right)\frac{1}{\sin(\psi)} = \cos(\phi)\sin(\theta)$$

Equating the two equations above:

$$\left(\frac{\ddot{X}m}{U_1} - \sin(\phi)\sin(\psi)\right)\frac{1}{\cos(\psi)} = \left(\frac{\ddot{Y}m}{U_1} + \sin(\phi)\cos(\psi)\right)\frac{1}{\sin(\psi)}$$

Multiply by $\cos(\psi)\sin(\psi)$:

$$\frac{\ddot{X}m\sin(\psi)}{U_1} - \sin(\phi)\sin^2(\psi) = \frac{\ddot{Y}m\cos(\psi)}{U_1} + \sin(\phi)\cos^2(\psi)$$

$$\frac{\ddot{X}m\sin(\psi)}{U_1} - \frac{\ddot{Y}m\cos(\psi)}{U_1} = \sin(\phi)\sin^2(\psi) + \sin(\phi)\cos^2(\psi)$$

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

$$\phi_d = \sin^{-1}\left(\frac{m}{U_1}(\ddot{X}\sin(\psi) - \ddot{Y}\cos(\psi))\right)$$

E Square Wave Trajectory

E.1 Optimal Trajectory Generation

The problem of minimum distance trajectory generation is formulated as a constrained optimization problem. For Z-trajectory:

Cost function: $Z(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$

Boundary condition: $Z(t_0) = \text{given},\ Z(t_f) = \text{given},\ (Z(t_0))' = 0,\ \text{and } (Z(t_f))' = 0$

Given: $t_0 = 0$ sec, $t_f = 10$ sec, $Z(t_0) = 0$ m, and $Z(t_f) = 20$ m.

$$\begin{aligned} Z(t_0) &= a_0 + a_1 t_0 + a_2 t_0^2 + a_3 t_0^3 \\ (Z(t_0))' &= a_1 + 2a_2 t_0 + 3a_3 t_0^2 \\ Z(t_f) &= a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 \\ (Z(t_f))' &= a_1 + 2a_2 t_f + 3a_3 t_f^2 \end{aligned}$$

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 0 & 1 & 2t_0 & 3t_0^2 \\ 1 & t_f & t_f^2 & t_f^3 \\ 0 & 1 & 2t_f & 3t_f^2 \end{bmatrix}^{-1} \begin{bmatrix} Z(t_0) \\ (Z(t_0))' \\ Z(t_f) \\ (Z(t_f))' \end{bmatrix}$$

By the same method, X and Y trajectories are generated.

```
%% square wave trajectory code
function [Xd,Yd,Zd,psi_d] = fcn(t)
if t<2.5
Xd=0;
Yd=0;
Zd=2.4*t^2-0.64*t^3;
psi_d=0;
elseif (2.5<=t)&&(t<=11.5)
Xd=0;
Yd=4.9382716049382717-4.2592592592592596*t+1.0370370370370372*t
    ^2-0.049382716049382716*t^3;
Zd=5;
psi_d=0;
elseif (11.5<=t)&&(t<=61.5)
Xd=18.3034-3.3948*t+0.1752*t^2-0.0016*t^3;
```

```
Yd=18;
Zd=5;
psi_d=0;
elseif (61.5<t)&&(t<=79.5)
Xd=100;
Yd=4150.458333333333-181.08333333333333348*t+2.61111111111111132*t
   ^2-0.0123456790123456788*t^3;
Zd=5;
psi_d=0;
elseif (79.5<t)&&(t<=129.5)
Xd=-1462.3658+49.4172*t-0.5016*t^2+0.0016*t^3;
Yd=54;
Zd=5;
psi_d=0;
elseif (129.5<t)&&(t<=147.5)
Xd=0;
Yd=32455.7793209876544-707.453703703703664*t+5.12962962962962968*t
   ^2-0.0123456790123456788*t^3;
Zd=5;
psi_d=0;
elseif (147.5<t)&&(t<=197.5)
Xd=7745.225-139.83*t+0.828*t^2-0.0016*t^3;
Yd=90;
Zd=5;
psi_d=0;
elseif (197.5<t)&&(t<=200)
Xd=100;
Yd=90;
Zd=-5024000+75840*t-381.6*t^2+0.64*t^3;
psi_d=0;
else
Xd=100;
Yd=90;
Zd=0;
psi_d=0;
end
```

F Stability Analysis

F.1 For SMC with PID Surface

Stability analysis for X

Defining Lyapunov candidate

$$V(S_1) = \frac{1}{2}S_1^2, \text{ since } V(0) = 0 \text{ and } V(S_1) > 0 \text{, it is positive definite.}$$

Taking the derivative

$$\dot{V}(S_1) = S_1\dot{S}_1 < 0$$

$$\dot{V}(S_1) = S_1(K_{px}\dot{e}_1 + K_{ix}e_1 - K_{dx}\ddot{X}_d + K_{dx}/m\left[\frac{m}{K_{dx}}(-Q_x\text{Sign}(S_1) + K_{dx}\ddot{X}_d - K_{px}\dot{e}_1 - K_{ix}e_1)\right])$$

$$\dot{V}(S_1) = S_1(-Q_x\text{Sign}(S_1)) = -S_1Q_x\text{Sign}(S_1)$$

If $S_1 > 0, Q_x > 0$

If $S_1 < 0, Q_x > 0$, So Q_x must be positive to be stable.

Stability analysis for Y

Defining Lyapunov candidate

$$V(S_3) = \frac{1}{2}S_3^2, \text{ since } V(0) = 0 \text{ and } V(S_3) > 0 \text{, it is positive definite.}$$

Taking the derivative

$$\dot{V}(S_3) = S_3\dot{S}_3 < 0$$

$$\dot{V}(S_3) = S_3(K_{py}\dot{e}_3 + K_{iy}e_3 - K_{dy}\ddot{Y}_d + K_{dy}/m\left[\frac{m}{K_{dy}}(-Q_y\text{Sign}(S_3) + K_{dy}\ddot{y}_d - K_{py}\dot{e}_3 - K_{iy}e_3)\right])$$

$$\dot{V}(S_3) = S_3(-Q_y\text{Sign}(S_3)) = -S_3Q_y\text{Sign}(S_3)$$

If $S_3 > 0, Q_y > 0$

If $S_3 < 0, Q_y > 0$, So Q_y must be positive to be stable.

Stability analysis for Z

Defining Lyapunov candidate

$$V(S_5) = \frac{1}{2}S_5^2, \text{ since } V(0) = 0 \text{ and } V(S_5) > 0 \text{, it is positive definite.}$$

Taking the derivative

$$\dot{V}(S_5) = S_5\dot{S}_5 < 0$$

$$\dot{V}(S_5) = S_5(K_{pz}\dot{e}_5 + K_{iz}e_5 - K_{dz}\ddot{Z}_d + K_{dz}/m\left[\frac{m}{K_{dz}}(-Q_z\text{Sign}(S_5) + K_{dz}\ddot{Z}_d - K_{pz}\dot{e}_5 - K_{iz}e_5)\right])$$

$$\dot{V}(S_5) = S_5(-Q_z\text{Sign}(S_5)) = -S_5Q_z\text{Sign}(S_5)$$

If $S_5 > 0, Q_z > 0$

If $S_5 < 0, Q_z > 0$, So Q_z must be positive to be stable.

Stability analysis for Phi

Defining Lyapunov candidate

$$V(S_7) = \frac{1}{2}S_7^2, \text{ since } V(0) = 0 \text{ and } V(S_7) > 0\text{, it is positive definite.}$$

Taking the derivative

$$\dot{V}(S_7) = S_7\dot{S}_7 < 0$$

$$\begin{aligned}\dot{V}(S_7) = S_7(&K_{p\phi}\dot{e}_7 + K_{i\phi}e_7 + K_{d\phi}a_1X_{10}X_{12} + K_{d\phi}a_2X_{10} - K_{d\phi}\ddot{\phi}_d \\ &+ K_{d\phi}b_1\left(\frac{1}{K_{d\phi}b_1}(-Q_\phi\text{Sign}(S_7) + K_{d\phi}\ddot{\phi}_d - K_{p\phi}\dot{e}_7 - K_{i\phi}e_7 - K_{d\phi}a_1X_{10}X_{12} - a_2X_{10}K_{d\phi})\right))\end{aligned}$$

$$\dot{V}(S_7) = S_7(-Q_\phi\text{Sign}(S_7)) = -S_7Q_\phi\text{Sign}(S_7)$$

If $S_7 > 0, Q_\phi > 0$

If $S_7 < 0, Q_\phi > 0$, So Q_ϕ must be positive to be stable.

Stability analysis for Theta

Defining Lyapunov candidate

$$V(S_9) = \frac{1}{2}S_9^2, \text{ since } V(0) = 0 \text{ and } V(S_9) > 0\text{, it is positive definite.}$$

Taking the derivative

$$\dot{V}(S_9) = S_9\dot{S}_9 < 0$$

$$\begin{aligned}\dot{V}(S_9) = S_9(&K_{p\theta}\dot{e}_9 + K_{i\theta}e_9 + K_{d\theta}a_3X_8X_{12} - K_{d\theta}a_4X_8 - K_{d\theta}\ddot{\theta}_d \\ &+ K_{d\theta}b_2\left(\frac{1}{K_{d\theta}b_2}(-Q_\theta\text{Sign}(S_9) + K_{d\theta}\ddot{\theta}_d - K_{p\theta}\dot{e}_9 - K_{i\theta}e_9 - K_{d\theta}a_3X_8X_{12} + a_4X_8K_{d\phi})\right))\end{aligned}$$

$$\dot{V}(S_9) = S_9(-Q_\theta\text{Sign}(S_9)) = -S_9Q_\theta\text{Sign}(S_9)$$

If $S_9 > 0, Q_\theta > 0$

If $S_9 < 0, Q_\theta > 0$, So Q_θ must be positive to be stable.

Stability analysis for Psi

Defining Lyapunov candidate

$$V(S_{11}) = \frac{1}{2}S_{11}^2, \text{ since } V(0) = 0 \text{ and } V(S_{11}) > 0\text{, it is positive definite.}$$

Taking the derivative

$$\dot{V}(S_{11}) = S_{11} \cdot (\dot{S}_{11}) < 0$$

$$\begin{aligned}\dot{V}(S_{11}) = S_{11} \cdot (&K_p\psi(\dot{e}_{11}) + K_i\psi e_{11} + K_d\psi a_5 X_8 X_{10} - K_d\psi\ddot{\psi} \\ &+ K_d\psi b_3 \left(\frac{1}{K_d\psi b_3}(-Q_\psi \text{Sign}(S_{11}) + K_d\psi\ddot{\psi} - K_p\psi(\dot{e}_{11}) - K_i\psi e_{11} - K_d\psi a_5 X_8 X_{10})\right))\end{aligned}$$

$$\dot{V}(S_{11}) = -S_{11} Q_\psi \text{Sign}(S_{11})$$

If $S_{11} > 0$, $Q_\psi > 0$

If $S_{11} < 0$, $Q_\psi > 0$, So Q_ψ must be positive to be stable

F.2 For STSMC with PID Surface

Stability analysis for X

Defining Lyapunov candidate

$$V(S_1) = \frac{1}{2}s_1^2, \text{ since } V(0) = 0 \text{ and } V(S_1) > 0, \text{ it is positive definite.}$$

Taking the derivative,

$$\dot{V}(S_1) = S_1\dot{S}_1 < 0$$

$$\begin{aligned}\dot{V}(S_1) = S_1 \Bigg(&K_{px}\dot{e}_1 + K_{ix}e_1 - K_{dx}\ddot{X}_d + \frac{K_{dx}}{m}\left(\frac{m}{K_{dx}}\left(-k_1|S_1|^{1/2}\text{sign}(S_1)\right.\right. \\ &\left.\left. - \; k_2\int \text{sign}(S_1)\,dt + K_{dx}\ddot{X}_d - K_{px}\dot{e}_1 - K_{ix}e_1\right)\right)\Bigg)\end{aligned}$$

$$\dot{V}(S_1) = S_1\left(-k_1|S_1|^{1/2}\text{sign}(S_1) - k_2\int \text{sign}(S_1)\,dt\right)$$

$$\dot{V}(0) = 0 \text{ and } \dot{V}(S_1) < 0$$

If $S_1 > 0$, to make $\dot{V}(S_1) < 0$, k_1 and k_2 must be positive. If $S_1 < 0$, to make $\dot{V}(S_1) < 0$, k_1 and k_2 must be positive.

Stability analysis for Y

Defining Lyapunov candidate

$$V(S_3) = \frac{1}{2}s_3^2, \text{ since } V(0) = 0 \text{ and } V(S_3) > 0, \text{ it is positive definite.}$$

Taking the derivative,

$$\dot{V}(S_3) = S_3\dot{S}_3 < 0$$

$$\begin{aligned}\dot{V}(S_3) = S_3\Bigg(&K_{py}\dot{e}_3 + K_{iy}e_3 - K_{dy}\ddot{Y}_d + \frac{K_{dy}}{m}\left(\frac{m}{K_{dy}}\left(-k_1|S_3|^{1/2}\text{sign}(S_3) - k_2\int \text{sign}(S_3)\,dt\right.\right. \\ &\left.\left. + K_{dy}\ddot{y}_d - K_{py}\dot{e}_3 - K_i y e_3\right)\right)\Bigg)\end{aligned}$$

$$\dot{V}(S_3) = S_3\left(-k_1|S_3|^{1/2}\text{sign}(S_3) - k_2\int \text{sign}(S_3)\,dt\right)$$

$$\dot{V}(0) = 0 \text{ and } \dot{V}(S_3) < 0$$

If $S_3 > 0$, to make $\dot{V}(S_3) < 0$, k_1 and k_2 must be positive. If $S_3 < 0$, to make $\dot{V}(S_3) < 0$, k_1 and k_2 must be positive.

Stability analysis for Z

Defining Lyapunov candidate

$V(S_5) = \frac{1}{2}S_5^2$, since $V(0) = 0$ and $V(S_5) > 0$, it is positive definite.

Taking the derivative

$$\dot{V}(S_5) = S_5 \cdot \dot{S}_5 < 0$$

$$\dot{V}(S_5) = S_5\left(K_{pz}\dot{e}_5 + K_{iz}e_5 - K_{dz}\ddot{Z}_d + \frac{K_{dz}}{m}\left(\frac{m}{K_{dz}}\left(-k_1\,|S_5|^{1/2}\,\text{sign}(S_5) - k_2\int \text{sign}(S_5)\,dt \right.\right.\right.$$
$$\left.\left.\left. + K_{dz}\ddot{Z}_d - K_{pz}\dot{e}_5 - K_{iz}e_5\right)\right)\right)$$

$$\dot{V}(S_5) = S_5\left(-k_1\,|S_5|^{1/2}\,\text{sign}(S_5) - k_2\int \text{sign}(S_5)\,dt\right)$$

If $S_5 > 0$, to make $\dot{V}(S_5) < 0$, k_1 and k_2 must be positive. If $S_5 < 0$, to make $\dot{V}(S_5) < 0$, k_1 and k_2 must be positive.

Stability analysis for Phi

Defining Lyapunov candidate

$V(S_7) = \frac{1}{2}S_7^2$, since $V(0) = 0$ and $V(S_7) > 0$, it is positive definite.

Taking the derivative

$V(S_7) = S_7 \cdot \dot{S}_7 < 0$

$$\dot{V}(S_7) = S_7\Big(K_{p\phi}\dot{e}_7 + K_{i\phi}e_7 + K_{d\phi}a_1X_{10}X_{12} + K_{d\phi}a_2X_{10} - K_{d\phi}\ddot{\phi}_d$$
$$+ K_{d\phi}b_1\left(\frac{1}{K_{d\phi}b_1}(-k_1|S_7|^{1/2}\text{sign}(S_7) - k_2\int \text{sign}(S_7)dt\right.$$
$$+K_{d\phi}\ddot{\phi}_d - K_{p\phi}\dot{e}_7 - K_{i\phi}e_7 - K_{d\phi}a_1X_{10}X_{12} - a_2X_{10}K_{d\phi}))\Big)$$

$$\dot{V}(S_7) = S_7\left(-k_1|S_7|^{1/2}\text{sign}(S_7) - k_2\int \text{sign}(S_7)\,dt\right)$$

If $S_7 > 0$, to make $\dot{V}(S_7) < 0$, k_1 and k_2 must be positive. If $S_7 < 0$, to make $\dot{V}(S_7) < 0$, k_1 and k_2 must be positive.

Stability analysis for Theta

Defining Lyapunov candidate

V(S$_9$) $= \frac{1}{2}S_9^2,$ since $V(0) = 0$ and $V(S_9) > 0,$ it is positive definite.

Taking the derivative

$$\dot{V}(S_9) = S_9\dot{S}_9 < 0$$

$$\begin{aligned}\dot{V}(S_9) = S_9\,(K_{p\theta}\dot{e}_9 + K_{i\theta}e_9 + K_{d\theta}a_3X_8X_{12} - K_{d\theta}a_4X_8 \\ -K_{d\theta}\ddot{\theta}_d + K_{d\theta}b_2\left(\frac{1}{K_{d\theta}b_2}(-k_1|S_9|^{1/2}\text{sign}(S_9) - k_2\int \text{sign}(S_9)\,dt)\right. \\ \left.\left. + K_{d\theta}(\ddot{\theta_d}) - K_{p\theta}(\dot{e_9}) - K_{i\theta}e_9 - K_{d\theta}a_3X_8X_{12} + a_4X_8K_{d\phi}\right)\right)\end{aligned}$$

$$\dot{V}(S_9) = S_9\left(-k_1|S_9|^{1/2}\text{sign}(S_9) - k_2\int \text{sign}(S_9)\,dt\right)$$

If $S_9 > 0$, to make $\dot{V}(S_9) < 0$, k_1 and k_2 must be positive. If $S_9 < 0$, to make $\dot{V}(S_9) < 0$, k_1 and k_2 must be positive.

Stability analysis for Psi

Defining Lyapunov candidate

V(S$_{11}$) $= \frac{1}{2}S_{11}^2,$ since $V(0) = 0$ and $V(S_{11}) > 0,$ it is positive definite.

Taking the derivative

$$\dot{V}(S_{11}) = S_{11}\dot{S}_{11} < 0$$

$$\begin{aligned}\dot{V}(S_{11}) = S_{11}\left(K_{p\psi}\dot{e}_{11} + K_{i\psi}e_{11} + K_{d\psi}a_5X_8X_{10} - K_{d\psi}\ddot{\psi}_d\right. \\ +K_{d\psi}b_3\left(\frac{1}{K_{d\psi}b_3}(-k_1|S_{11}|^{1/2}\text{sign}(S_{11}) - k_2\int \text{sign}(S_{11})dt\right. \\ \left.\left. +K_{d\psi}\ddot{\psi}_d - K_{p\psi}\dot{e}_{11} - K_{i\psi}e_{11} - K_{d\psi}a_5X_8X_{10}))\right)\right.\end{aligned}$$

$$\dot{V}(S_{11}) = S_{11}\left(-k_1|S_{11}|^{1/2}\text{sign}(S_{11}) - k_2\int \text{sign}(S_{11})\,dt\right)$$

If $S_{11} > 0$, to make $\dot{V}(S_{11}) < 0$, k_1 and k_2 must be positive. If $S_{11} < 0$, to make $\dot{V}(S_{11}) < 0$, k_1 and k_2 must be positive.

G Fuzzy Membership Functions

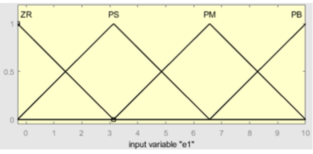


Figure G.1: Input Membership Function for X-PID

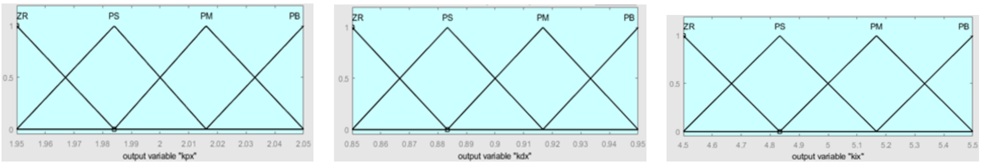


Figure G.2: Output Membership Function for X-PID

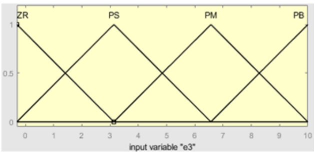


Figure G.3: Input Membership Function for Y-PID

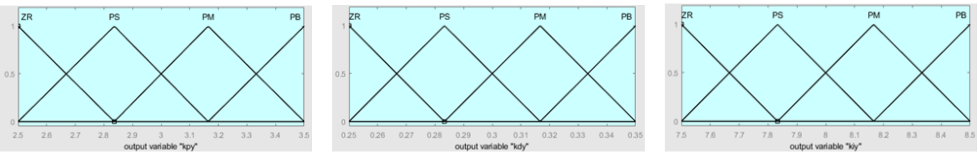


Figure G.4: Output Membership Function for Y-PID

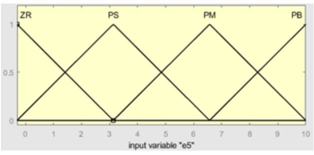


Figure G.5: Input Membership Function for Z-PID

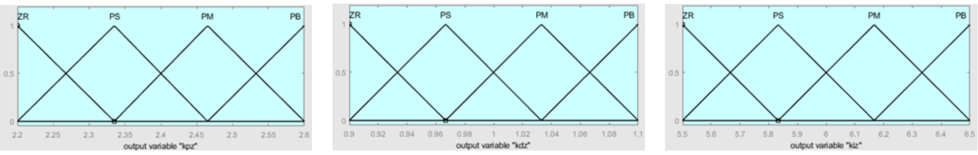


Figure G.6: Output Membership Function for Z-PID

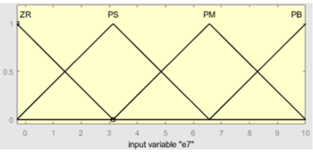


Figure G.7: Input Membership Function for Phi-PID

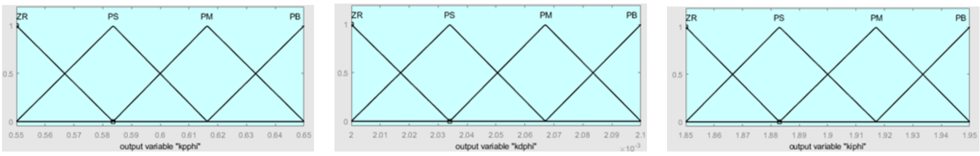


Figure G.8: Output Membership Function for Phi-PID

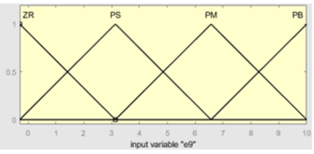


Figure G.9: Input Membership Function for Theta-PID

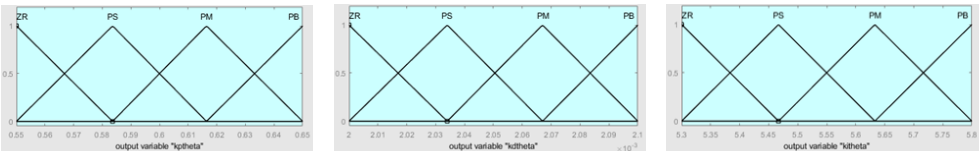


Figure G.10: Output Membership Function for Theta-PID

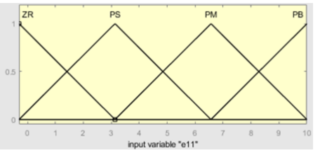


Figure G.11: Input Membership Function for Psi-PID

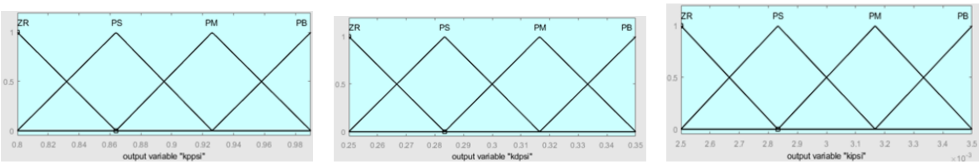


Figure G.12: Output Membership Function for Psi-PID

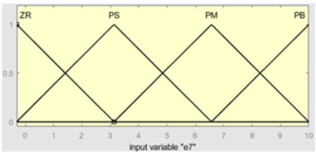


Figure G.13: Input Membership Function for Phi-PID (Spiral Infinity and Square Wave)

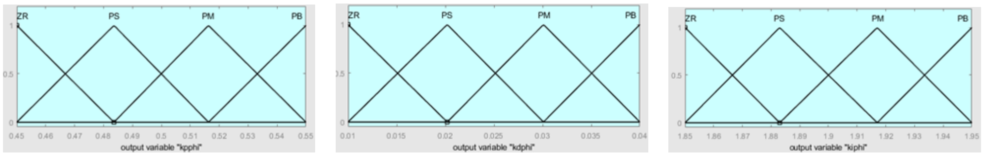


Figure G.14: Output Membership Function for Phi-PID (Spiral Infinity Square Wave)

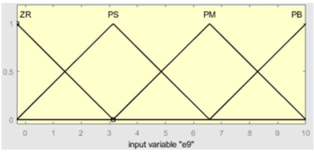


Figure G.15: Input Membership Function for Theta-PID (Spiral Infinity and Square Wave)

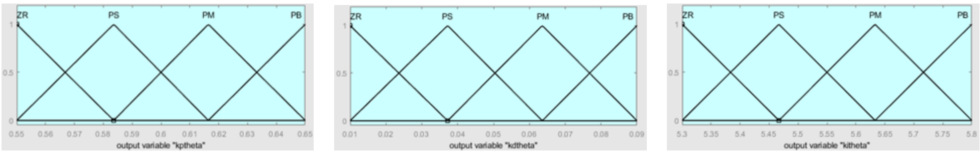


Figure G.16: Output Membership Function for Theta-PID (Spiral Infinity and Square Wave)

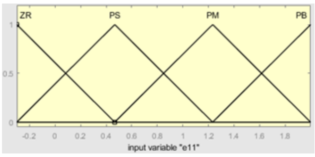


Figure G.17: Input Membership Function for Psi-PID (Spiral Cylindrical Trajectory)

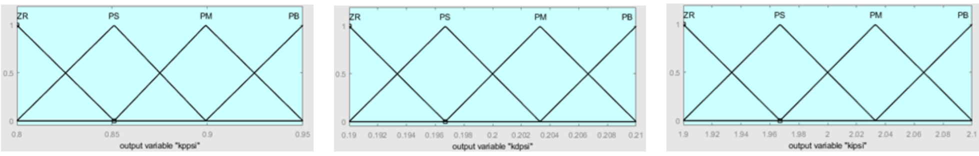


Figure G.18: Output Membership Function for Psi-PID (Spiral Cylindrical Trajectory)

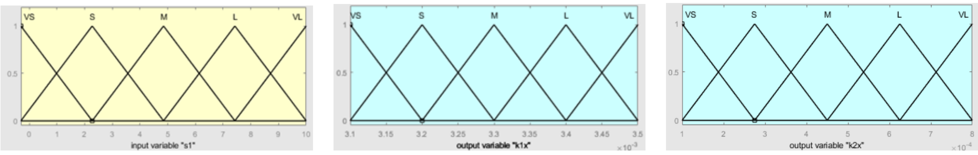


Figure G.19: Input and Output Membership Function for X-ST

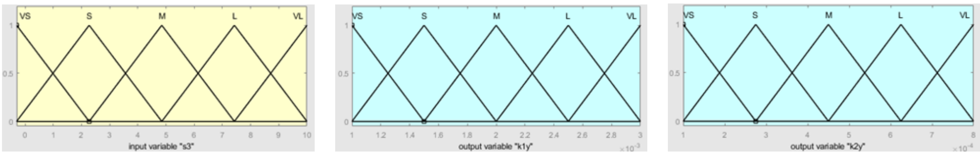


Figure G.20: Input and Output Membership Function for Y-ST

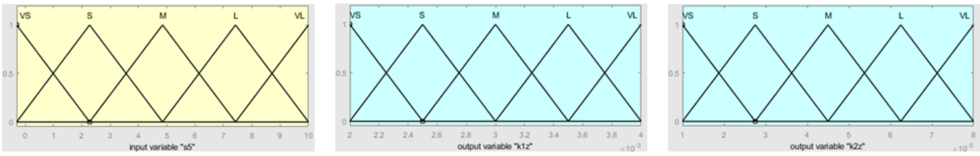


Figure G.21: Input and Output Membership Function for Z-ST

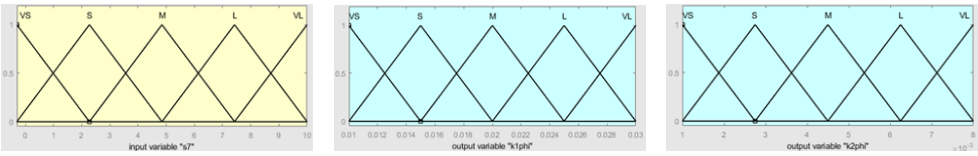


Figure G.22: Input and Output Membership Function for Phi-ST

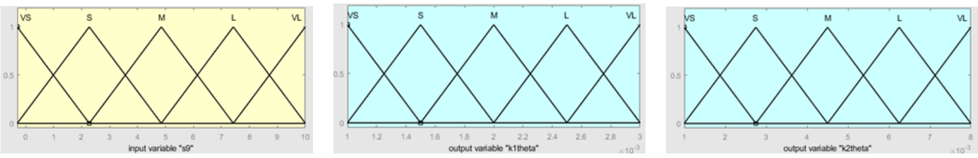


Figure G.23: Input and Output Membership Function for Theta-ST

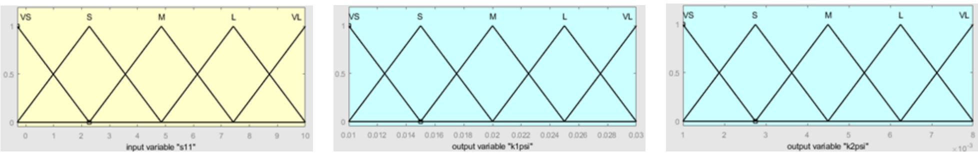


Figure G.24: Input and Output Membership Function for Psi-ST

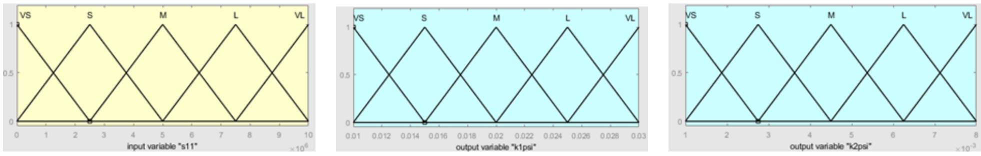


Figure G.25: Input and Output Membership Function for Psi-ST (Spiral Cylindrical)

H Wheat Yield Estimation

H.1 Wheat Head Training Code

Main code

```
from ultralytics import YOLO
# Load a model
model = YOLO("yolov8n.pt")  # load a pretrained model
# Use the model
results = model.train(data="path.yaml", epochs=70)  # train the model
metrics = model.val()
```

Path

```
path: C:\Users\elisa\Desktop\Image ProcessingFinal\New folder\WheatHeadTraining # dataset location
train: images\train # train images (relative to 'path')
val: images\val # val images (relative to 'path')
predict: images\test
# Classes
names:
  0: wheat_head
```

H.2 Wheat Head Count Code

```
from ultralytics import YOLO
import cv2
import glob
# Load a model
model = YOLO('best.pt')  # pretrained YOLOv8n model
images_folder = glob.glob("images/*.png")  # Load images from a folder
WheatHead_folder_count = 0
for path in images_folder:
    print("Img path", path)
    results = model(path)
    for result in results:
        boxes = result.boxes
        WheatHead_count = len(boxes)
        WheatHead_folder_count += WheatHead_count
    results_plotted = results[0].plot()
    cv2.imshow("result", results_plotted)
    cv2.waitKey(1)
print("Total current count", WheatHead_folder_count)
```

H.3 Wheat Yield Estimation Code

For one image

```
from ultralytics import YOLO
import cv2
model = YOLO('best.pt')
```

```
WheatHead_folder_count = 0
results = model('yeboklawheat.jpg')
for result in results:
    boxes = result.boxes
    WheatHead_count = len(boxes)
    WheatHead_folder_count += WheatHead_count
    results_plotted = results[0].plot()
    cv2.imshow("result", results_plotted)
    cv2.waitKey(1)
print("Total current count", WheatHead_folder_count)
A = input('Enter Kernels Per Head ')
B = input('Enter Weight of Kernel in Kg ')
C = input('Enter the Area in meter square')
KernelsPerHead = int(A)
WeightOfKernel = float(B)
Area = float(C)
WheatYield = ((WheatHead_folder_count) * (KernelsPerHead) * (WeightOfKernel)) / (Area)
print("wheat yield estimation ", WheatYield, "Kg/m2")
```

For more than one image

```
from ultralytics import YOLO
import cv2
import glob
# Load a model
model = YOLO('best.pt')  # pretrained YOLOv8n model
images_folder = glob.glob("images/*.png")    # Load images from a folder
WheatHead_folder_count = 0
for path in images_folder:
    print("Img path", path)
    results = model(path)
    for result in results:
      boxes = result.boxes
      WheatHead_count = len(boxes)
      WheatHead_folder_count += WheatHead_count
    results_plotted = results[0].plot()
    cv2.imshow("result", results_plotted)
    cv2.waitKey(1)
print("Total current count", WheatHead_folder_count)
A = input('Enter Kernels Per Head ')
B = input('Enter Weight of Kernel in Kg ')
C = input('Enter the Area in in meter square')
KernelsPerHead = int(A)
WeightOfKernel = float(B)
Area = float(C)
WheatYield = ((WheatHead_folder_count)*(KernelsPerHead)*(WeightOfKernel))/(Area)
print("wheat yield estimation ", WheatYield, "Kg/m2")
```