Addis Ababa University

College of Technology and Built Environment

School of Electrical and Computer Engineering

Adaptive Sliding Mode Control Design for Quadrotor Position and Attitude Tracking in Desert Locust Management

---

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

By Muluemebet Semagn

Advisor

Chala Merga (PhD)

May 11, 2026

Addis Ababa, Ethiopia

Addis Ababa University

College of Technology and Built Environment

School of Electrical and Computer Engineering

Adaptive Sliding Mode Control Design for Quadrotor Position and Attitude Tracking in Desert Locust Management

By

Muluemebet Semagn

APPROVED BY BOARD OF EXAMINER

Name	Signature	Date
Dean,School of Graduate Committe		
Advisor		
Internal Examiner		
External Examiner		

Declaration

I, Muluemebet Semagn, hereby declare that: this thesis is my original work. All research was conducted independently under supervision. No part of this work has been plagiarized. Proper citations are provided for all referenced materials. Findings and conclusions are based on my analysis. This work has not been submitted for any other degree. I take full responsibility for the content. Any collaborative contributions are clearly acknowledged.

Signature: _____ Date: _____

Name: Muluemebet Semagn Place: Addis Ababa, Ethiopia

Acknowledgment

First and foremost, praises and thanks to GOD, the Almighty, for all the showers of blessings bestowed in my life.

I would like to express my deep and sincere gratitude to my advisor Dr. Chala Merga(phd) Assistant Professor, School of Electrical and Computer Engineering, Addis Ababa Institute of Technology (AAiT), Addis Ababa University (AAU) for his mentorship, guidance, friendship, empathy and great sense of humor and support throughout the MSc program.

Special thanks go to those, present themselves to hand for me.

Abstract

This paper presents a robust adaptive sliding mode control (ASMC) design integrated with a Kalman filter for the position and attitude control of a quadrotor unmanned aerial vehicle (UAV) aimed at desert locust identification and management. The proposed control system addresses the challenges posed by desert conditions, including external disturbances such as wind, and the non-linear dynamics of the quadrotor. The sliding mode control technique ensures robustness and stability, while the adaptive mechanism updates control parameters. The Kalman filter provides accurate state estimation in the presence of noise, enhancing the controller's performance.the control objectives are dynamically adjusted based on control parameters. Simulation demonstrate the effectiveness of the proposed control system in maintaining stable flight and accurate positioning, making it a valuable tool for efficient desert locust monitoring and management.

Keywords: ASMC, Quadrotor UAV, MATLAB, Simulink

Contents

Acknowledgment ii

Abstract . . . . . ii iii

List of Figures 8

List of Tables 9

List of Acronyms 10

1 Introduction 2

1.1 General Background 2

1.2 Statement of the Problem 4

1.3 Objective of the Thesis 5

1.3.1 General Objective 5

1.3.2 Specific Objectives 5

1.4 Significance of the Thesis 5

1.5 Scope and Limitation of Thesis 7

1.6 Methodology 8

1.7 Thesis Outline 9

2 Literature Review 10

2.1 Application of Quadrotor UAVs 11

2.2 Control Problem of Quadrotor UAVs 12

2.2.1 Conventional Control of Quadrotor UAVs 13

2.2.2 Adaptive Control of Quadrotor UAVs 13

2.2.3 Robust Control of Quadrotor UAVs 14
2.3 State Estimation and Noise Reduction 17

3 Quadrotor UAV Mathematical Modeling 18
3.1 Working Principle 18
3.2 Quadrotor UAV Configuration Setup 19
3.3 Reference Frame Definitions 20
3.4 Quaternion Based Modeling 21
3.4.1 Coordinate Transformation 22
3.4.2 Rotation Transformation Matrices in Space 24
3.5 Rigid Body Mathematical Modeling 25
3.5.1 Kinematic Modeling of Quadrotor UAV 26
3.5.2 Dynamic Modeling of Quadrotor UAV 27
3.5.2.1 Quadrotor Translational Equations of Motion 27
3.5.2.2 Thrust Generation 27
3.5.2.3 Gyroscopic Effect Compensation 28
3.5.2.4 The Force of Gravity 28
3.5.2.5 The effect of Drag in Quadrotor 29
3.5.2.6 Quadrotor Rotational Equations of Motion 30
3.6 State Space Model of Attitude 32
3.7 Model Verification of Quadrotor UAV Mathematical Modeling 36
3.7.1 Simulation Framework 36
3.7.2 Position Verification 37
3.7.3 Attitude Verification Using Quaternions 37
3.7.4 Discussion 39
3.7.5 Conclusion 39

4 Quadrotor UAV Controller and State Estimator Design 40
4.1 State Estimation Design 40
4.1.1 Kalman Filter Overview 40

4.2 Adaptive Sliding Mode Controller Design 42

4.2.1 Sliding Mode Control Design for Position 46

4.2.2 Sliding Mode Control Design for Quaternion 48

4.2.3 Stability Analysis 51

4.2.3.1 Stability Analysis for U_2 (Torque Control About the x-Axis) 52 4.2.3.2 Stability Analysis for U_3 (Torque Control About the y-Axis) 53

4.2.3.3 Stability Analysis for U_4 (Torque Control About the z-Axis) 54

4.3 Adaptive Switching Gains in SMC 55

4.3.1 Fuzzy Inference System Description 55

4.4 Enhanced Controller Design with Fuzzy Logic and Kalman Filter 58

4.4.1 Integrated Control Architecture 58

4.4.2 Fuzzy Logic Adaptation Mechanism 58

4.4.3 Kalman Filter Implementation 58

4.5 Simulation Results and Comparative Analysis 59

4.5.1 Performance with and without Kalman Filter 59

4.5.2 Comparison with Other Control Strategies 59

4.5.3 Disturbance Rejection Analysis 59

5 Simulation Results and Analysis 61

5.1 Helical Trajectory Tracking 62

5.2 BowTie Trajectory Tracking 67

5.3 Comparative Analysis of Controller Performance 71

5.3.1 Position Tracking Performance 72

5.3.2 Velocity Tracking Performance 73

5.3.3 Conclusion 73

5.4 Implications for Desert Locust Management 74

5.4.1 Application in Locust Monitoring 74

5.4.2 Integration with Imaging and Machine Learning 74

5.4.3 Enhancement of Monitoring Strategies 74

5.5 Desert Locust Identification Using Convolutional Neural Networks 75
5.5.1 Overview 75
5.5.2 Dataset Acquisition and Preprocessing 75
5.5.3 CNN Architecture 75
5.5.4 Training Procedure 76
5.5.5 Evaluation and Results 77
5.5.6 Conclusion 78
5.5.7 Integrated System Overview 78

6 Conclusion and Recommendation 80

References 81

List of Figures

1.1 Drone Spraying Locusts [35] 3

1.2 Methodology 8

3.1 Thrust Generation and Weight in Quadrotor 19

3.2 Reference Frame Definition 21

3.3 Rotation in Quaternion Operation 25

3.4 Position Verification under Translational and Rotational Motion. Each subplot compares model and simulation outputs for X, Y , and Z positions. The close alignment between curves confirms the fidelity of the translational kinematics. 37

3.5 Quaternion-Based Attitude Verification under Translational and Rotational Motion. Model and simulation outputs are compared for yaw, roll, and pitch dynamics. The close match across all plots validates the quaternion propagation and attitude modeling. 38

4.1 Cascade Control System Architecture 43

4.2 Membership Functions for Sliding Surface S. 56

4.3 Membership Functions for Sliding Surface Derivative $\dot{S}$ 57

4.4 Adaptive Switching SMC Gain k_i. 57

4.5 Integrated control architecture showing ASMC, fuzzy adaptation, and Kalman filter 58

4.6 Comparison of roll angle tracking with and without Kalman filter 59

4.7 Performance comparison of ASMC, SMC, and PID controllers 59

4.8 Disturbance rejection performance of ASMC 60

5.1 3D Position Plot in Helical Trajectory Tracking 62

5.2 Position and Attitude Plot in Helical Trajectory Tracking 63

5.3 Quaternion (q_1) Plot in Helical Trajectory Tracking 63
5.4 Quaternion (q_2) Plot in Helical Trajectory Tracking 64
5.5 Quaternion (q_3) Plot in Helical Trajectory Tracking 65
5.6 Adaptive Gains for x and y Control in Helical Trajectory Tracking . . . 65
5.7 Adaptive Gains for z and q_1 Control in Helical Trajectory Tracking . . . 66
5.8 Adaptive Gains for q_2 and q_3 Control in Helical Trajectory Tracking . . 66
5.9 3D Position Plot in BowTie Trajectory Tracking 67
5.10 Position and Attitude Plot in BowTie Trajectory Tracking 68
5.11 Quaternion (q_1) Plot in Bow-Tie Trajectory Tracking 68
5.12 Quaternion (q_2) Plot in Bow-Tie Trajectory Tracking 69
5.13 Quaternion (q_3) Plot in Bow-Tie Trajectory Tracking 70
5.14 Adaptive Gains for x and y Control in BowTie Trajectory Tracking . . . 70
5.15 Adaptive Gains for z and q_1 Control in BowTie Trajectory Tracking . . 71
5.16 Adaptive Gains for q_2 and q_3 Control in BowTie Trajectory Tracking . . 71
5.17 Position tracking performance and error comparison for ASMC-KF, ASMC Raw, SMC, and PID controllers. 72
5.18 Velocity tracking performance and error comparison for ASMC-KF, ASMC Raw, SMC, and PID controllers. 73
5.19 CNN architecture for desert locust image classification. 76
5.20 Training loss curve over 10 epochs. 77
5.21 Confusion matrix of CNN classifier on validation set. 77
5.22 Integrated system for desert locust identification and management 79

List of Tables

4.1 Fuzzy rule base for switching gain adaptation 58
4.2 RMSE comparison with and without Kalman filter 59
4.3 Performance metrics comparison of different controllers 60

List of Acronyms

UAV	Unmanned Aerial Vehicle
COG	Center of Gravity
DOF	Degree of Freedom
MRAC	Model Reference Adaptive Controller
SLTV	system Linear Time Variant system
DMRAC	Direct Model Reference Adaptive Controller
InMRAC	Indirect Model Reference Adaptive Controller
PSO	Particle Swarm Optimization
ITAE	Integral Time Absolute Error
CNN	Convolutional Neural Networks
VGGNet	Visual Geometry Community Network
PID-gains	Proportion - Derivative - Integral gains

Chapter 1 Introduction

1.1 General Background

Unmanned Aerial Vehicles (UAVs), particularly quadrotors, have become essential tools in various applications due to their versatility and ability to operate in diverse environments. Their capability to hover, take off, and land vertically makes them particularly suited for tasks that require precise positioning and maneuverability [9]. In recent years, UAVs have seen extensive use in agriculture, surveillance, and environmental monitoring. The integration of advanced control strategies and real-time data processing enhances their performance, particularly in complex and dynamic environments.

Desert locusts (Schistocerca gregaria) pose a significant threat to agriculture and food security in many parts of the world [35, 45]. Their swarming behavior can lead to devastating crop losses, impacting the livelihoods of millions of people. Effective management of locust swarms requires timely detection and monitoring, which can be challenging due to the vast and often inaccessible areas they inhabit. UAVs offer a promising solution for this problem by providing aerial surveillance capabilities that can cover large areas quickly and efficiently.

Figure 1.1: Drone Spraying Locusts [35]

The control of quadrotor UAVs involves managing both their position and attitude, which is a complex task due to the inherent non-linearity and coupling of their dynamics [8, 12, 18, 19, 27]. Position control deals with maintaining a specific location in space, while attitude control involves managing the orientation of the UAV. These tasks become more challenging in the presence of external disturbances such as wind and dust, which are prevalent in desert environments. To address these challenges, robust and adaptive control strategies are essential.

Sliding Mode Control (SMC) is a well-established method for achieving robustness in control systems [7, 32, 38]. It is known for its ability to handle uncertainties and disturbances by driving the system states to a predefined sliding surface. However, traditional SMC methods may not be sufficient when dealing with real-world applications where model uncertainties and external disturbances are prevalent. To enhance the robustness of the control system, an adaptive mechanism can be incorporated, allowing the control parameters to adjust in real-time based on observed conditions.

The Kalman filter is a powerful tool for state estimation in systems affected by noise and measurement errors [4, 14, 44]. It provides a means to combine multiple sources of information to produce an estimate of the system's state that is more accurate than any single measurement alone. In the context of UAV control, the Kalman filter can improve the accuracy of position and attitude estimates, thereby enhancing the overall performance of the control system.

Combining SMC with an adaptive mechanism and Kalman filtering offers a promising approach to addressing the challenges of UAV control in desert environments. This integration aims to create a control system that is not only robust and stable but also capable of adapting to changing conditions and providing accurate state estimations. The proposed solution aims to advance the capabilities of UAVs in desert locust management, ultimately contributing to more effective pest control and improved agricultural practices.

1.2 Statement of the Problem

Desert locust swarms pose a significant threat to agriculture and food security in affected regions. Effective identification and management of these swarms are critical to mitigate their devastating impact. However, controlling and monitoring desert locusts in vast and challenging desert environments present numerous difficulties. Harsh desert conditions, including extreme temperatures, wind, and dust, complicate the operation of UAVs and sensor systems. Wind gusts and other environmental disturbances affect the stability and control of the quadrotor UAV, making precise navigation and positioning difficult. The non-linear dynamics of the quadrotor UAV necessitate advanced control strategies to achieve stable flight and accurate positioning. Additionally, the control system must adapt in real-time to changing conditions and uncertainties in the environment. Accurate state estimation is challenging due to noise and errors in sensor measurements. Finally, the UAV must be equipped with reliable sensors and algorithms to effectively identify and map locust swarms in real-time. To address these challenges, this thesis aims to design a robust adaptive sliding mode control system integrated with a Kalman filter for the position and attitude control of a quadrotor UAV, specifically for the purpose of desert locust identification and management.

1.3 Objective of the Thesis

1.3.1 General Objective

The general objective of the thesis is to design a control system for a quadrotor UAV that can maintain stable flight and accurately position itself for desert locust identification and management.

1.3.2 Specific Objectives

The specific objective of the research are:

- To derive the mathematical model of the quadrotor UAV, including translational and rotational dynamics.
- To design a sliding mode control (SMC) system for robust and stable position and attitude control of the quadrotor UAV.
- To integrate an adaptive mechanism into the SMC to update control parameters in real-time, addressing system uncertainties and external disturbances.
- To implement a Kalman filter for accurate state estimation in the presence of noise.
- To create a simulation environment that mimics desert conditions and locust swarms for testing the controller's performance.
- To validate the designed control system through simulations, analyzing its effectiveness in maintaining stable flight and accurate positioning.

1.4 Significance of the Thesis

This research has the following significant contributions:

- Development of a robust control system that significantly improves the accuracy of position and attitude control for quadrotor UAVs in challenging environments.

- Implementation of an adaptive mechanism that allows the control system to adjust in real-time, providing robustness against system uncertainties and external disturbances.
- Integration of a Kalman filter that enhances the state estimation accuracy in the presence of sensor noise, leading to more reliable control performance.
- Provision of an effective tool for desert locust identification and management, which can help mitigate the impact of locust swarms on agriculture and food security.
- Contribution to the advancement of UAV control technology, particularly in the area of robust adaptive control methods.
- Potential to reduce the economic losses caused by desert locust infestations by providing a more efficient and effective means of monitoring and managing locust populations.
- Support for environmentally friendly pest control methods by enabling precise targeting of locust swarms, reducing the need for widespread pesticide use.

1.5 Scope and Limitation of Thesis

The scope of this thesis encompasses the development and validation of a robust adaptive sliding mode control (SMC) system integrated with a Kalman filter for the precise position and attitude control of a quadrotor unmanned aerial vehicle (UAV). This control system is specifically designed for the identification and management of desert locusts. The primary focus includes the derivation of the quadrotor's mathematical model, the design and implementation of the SMC and adaptive mechanisms. The thesis also covers the simulation of desert conditions to test the controller's performance.

However, several limitations should be acknowledged. The control design primarily considers standard environmental disturbances, and extreme or unpredictable conditions may still pose challenges to the UAV's stability and control. The scope is limited to the hardware and sensors used in this study, and different UAV models or sensor types may require additional adjustments to the control algorithms. Real-time adaptation and state estimation rely on the accuracy of the sensor data, which may be affected by noise and

measurement errors. Furthermore, while simulations provide valuable insights, the results may vary under different real-world conditions. This thesis does not delve into the economic analysis or broader implementation strategies for large-scale locust management, focusing instead on the technical aspects of control system design and validation.

1.6 Methodology

The basic procedures to complete the thesis and meet specified objectives of research are shown in the following flowchart, 1.2.

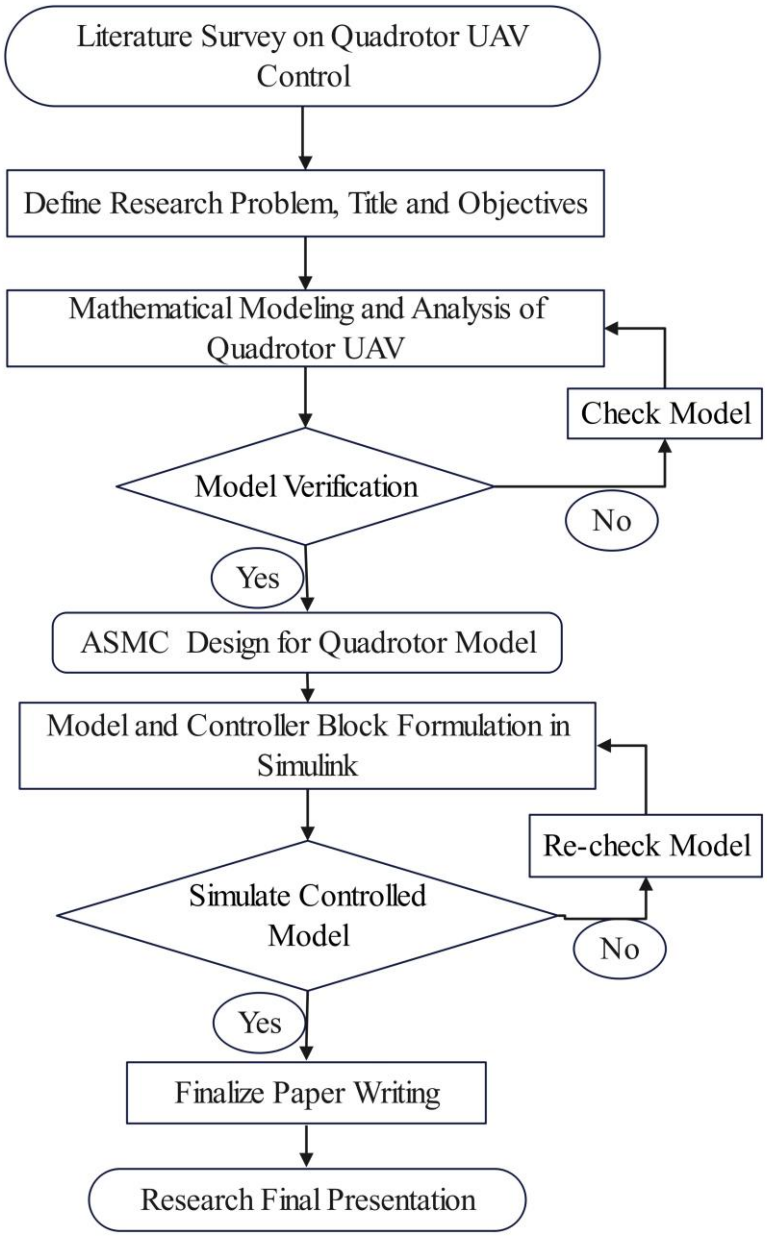


Figure 1.2: Methodology

1.7 Thesis Outline

The thesis outline is organized into six chapters.

Chapter 1 presents a general introduction to the position and attitude control of quadrotor UAVs, emphasizing the challenges in desert locust identification and management. Chapter 2

discusses a review of various research papers that are significant and provide background concepts for this research. This includes studies on sliding mode control, adaptive control mechanisms, and Kalman filters.

Chapter 3 presents the dynamic modeling of the quadrotor UAV. The mathematical model, including translational and rotational dynamics, is discussed thoroughly.

Chapter 4 presents the design of the robust adaptive sliding mode control system and the integration of the Kalman filter for state estimation. The control scheme design principles and the adaptive mechanism are explained in detail.

Chapter 5 discusses the simulation results and the analysis based on these simulations. The performance of the control system in various scenarios, including desert conditions, is evaluated.

Chapter 6 concludes the overall analysis performed, and recommendations for future work are presented. This includes potential improvements and additional applications for desert locust management.

Chapter 2

Literature Review

In recent years, the application of Unmanned Aerial Vehicles (UAVs), particularly quadrotors, has garnered significant attention due to their potential in various domains. One of the critical areas where quadrotors are making an impact is in environmental monitoring and pest management. The use of quadrotors for desert locust identification and management has emerged as a promising approach due to their ability to cover large areas and provide real-time data. As noted in [4, 22, 41?] the integration of UAV technology in agricultural management allows for efficient monitoring and control of pest outbreaks, which is crucial for mitigating the damage caused by desert locusts.

The design and control of quadrotor UAVs for such specialized applications involve complex challenges, particularly in maintaining precise control over both position and attitude. The unique dynamics of quadrotors, combined with the need to operate effectively in varying environmental conditions, necessitate advanced control strategies. Researches under [9, 13, 18, 33] highlight the importance of adaptive control techniques in addressing these challenges, as they allow the UAV to adjust its parameters in real-time to handle dynamic and unpredictable conditions.

Adaptive Sliding Mode Control (ASMC) has been identified as a powerful approach for managing the nonlinear dynamics of quadrotor UAVs, especially in applications requiring high precision and robustness [18, 19, 27, 31]. ASMC combines the benefits of adaptive control and sliding mode control, offering a solution to the issues of model uncertainties and external disturbances. These literatures discuss how ASMC can be applied

to enhance the stability and performance of quadrotor systems, making it suitable for complex tasks as pest identification and management.

Hence, throughout this literature a comprehensive exploration of quadrotor UAVs from their application area, design, and control is done in the following sections. This literature review covers the advancements in UAV quaternion-based-modeling, development of control strategies, and the specific implementation of Adaptive Sliding Mode Control. The subsequent sections delve into the state-of-the-art research and methodologies, providing insights into how these technologies can be optimized for effective pest management and environmental monitoring.

2.1 Application of Quadrotor UAVs

Quadrotor Unmanned Aerial Vehicles (UAVs) have found a wide range of applications across various fields due to their versatility, maneuverability, and ease of control. The following literatures highlight some of the significant applications of quadrotor UAVs.

Quadrotors are extensively used in aerial photography and videography, providing unique perspectives and high-quality imagery [?]. Their ability to hover and maneuver precisely makes them ideal for capturing detailed images and videos from various angles. Advances in camera technology and stabilization systems have further enhanced their effectiveness in this domain. The literature [?] discusses the integration of highresolution cameras with quadrotors to achieve identification of Inboch plant in Ethiopia.

In surveillance and monitoring applications, quadrotor UAVs are employed for tasks such as security surveillance, environmental monitoring, and disaster management. Their ability to access hard-to-reach areas and cover large areas efficiently makes them valuable tools for real-time monitoring .

Quadrotors have been increasingly utilized in precision agriculture for tasks such as crop monitoring, pest detection, and precision spraying [22, 35, 45]. They enable farmers to gather data on crop health and optimize the use of resources. Researches show how quadrotors equipped with multispectral sensors can assess crop health and detect issues early, leading to more efficient and sustainable farming practices.

In search and rescue operations, quadrotors provide valuable assistance by delivering real-time aerial views and locating missing persons in challenging environments [11]. Their

ability to operate in diverse terrains and adverse weather conditions enhances their utility in rescue missions. Furthermore, quadrotor UAVs are increasingly used for inspecting and maintaining infrastructure such as bridges, power lines, and pipelines, and in delivery and logistics as an emerging field with significant potential.

Furthermore, quadrotor UAVs have proven to be highly versatile and effective tools across various applications. Their continued development and integration into different fields promise to expand their potential and impact, offering innovative solutions to complex problems.

2.2 Control Problem of Quadrotor UAVs

The control of quadrotor UAVs presents significant challenges due to the inherent complexity and nonlinearity of their models [13, 15, 19, 24, 27, 31, 33, 42]. The dynamics of a quadrotor are governed by nonlinear equations of motion that account for the effects of aerodynamic forces, motor dynamics, and rotor interactions. This nonlinearity arises from the dependence of the quadrotor's thrust and torque on rotor speeds, resulting in a complex set of nonlinear differential equations.

One of the primary challenges in controlling quadrotors is ensuring stability [16]. The nonlinear nature of the quadrotor's dynamics makes it difficult to maintain stability under various flight conditions. Secondly, precise control over position, orientation, and velocity is crucial, especially in tasks that require accurate maneuvers [1, 7, 34]. Environmental disturbances, such as wind, further complicate the control problem, necessitating robust control strategies.

To address these challenges, various control methods have been explored, ranging from conventional to robust approaches in the following sections. Research in quadrotor control continues to advance, exploring new strategies that combine robustness with adaptability. Hybrid controllers and learning-based methods are examples of emerging approaches that aim to improve control performance and address the challenges of quadrotor UAVs.

2.2.1 Conventional Control of Quadrotor UAVs

Conventional control methods for quadrotor UAVs are foundational techniques that have been widely used due to their simplicity and ease of implementation. These methods include Proportional-Integral-Derivative (PID) control and Linear Quadratic Regulator (LQR) control. Each approach offers distinct advantages and limitations when applied to the dynamics of quadrotors.

Proportional-Integral-Derivative (PID) control is one of the most commonly employed control strategies in quadrotor UAVs. PID controllers work by calculating an error value as the difference between a desired set-point and a measured process variable. PID control is often used for regulating altitude, velocity, and attitude [32, 34]. These literatures demonstrate the application of PID control for quadrotor altitude and attitude stabilization, noting its effectiveness in simple, well-defined environments. However, while PID controllers are straightforward and effective for many scenarios, they struggle with handling the complex, nonlinear dynamics of quadrotors, particularly in the presence of external disturbances and dynamic changes.

LQR control is applied to manage linearized models of the quadrotor dynamics around a stable operating point. Research by [38] highlights the use of LQR for quadrotor position and attitude control, showing its effectiveness in scenarios where the system can be approximated as linear. LQR controllers offer robust performance in terms of stability and optimality within the linear regime. Nonetheless, LQR control is inherently limited to linear systems and may not perform well under significant nonlinearities or when the system deviates substantially from the linear model. This limitation necessitates the use of more advanced or hybrid control methods to address the full range of dynamic behaviors encountered in real-world quadrotor operations.

2.2.2 Adaptive Control of Quadrotor UAVs

Adaptive control is a sophisticated approach designed to handle systems with uncertain or time-varying dynamics, making it particularly relevant for quadrotor UAVs. Unlike conventional methods, adaptive control adjusts its parameters in real-time to cope with

changes in system dynamics or external disturbances, thereby enhancing performance in dynamic environments.

One popular adaptive control method is Model Reference Adaptive Control (MRAC). MRAC designs a control system based on a reference model that specifies desired system behavior. The adaptive controller adjusts its parameters to ensure that the actual system performance matches the reference model. This approach is useful for quadrotors, as it allows the system to adapt to changes in dynamics and external conditions. [3, 20, 28, 43] demonstrates the application of MRAC for quadrotor control, showing how it can effectively handle variations in payload and environmental disturbances.

2.2.3 Robust Control of Quadrotor UAVs

Robust control methods are essential for managing the complex and uncertain dynamics of quadrotor UAVs, providing the ability to maintain performance despite variations in system parameters and external disturbances [16, 18, 27]. Techniques such as Sliding Mode Control (SMC) and H-infinity control are widely used to address these challenges. SMC is particularly effective due to its ability to handle model uncertainties and disturbances by driving the system trajectories to a predefined sliding surface, thereby ensuring robustness against external disturbances. H-infinity control, on the other hand, optimizes the worst-case performance by minimizing the impact of disturbances and uncertainties on system performance.

Hassani et al. (2020), proposed a Robust Adaptive Sliding Mode Controller (RASMC) based on a new adaptive reaching law for quadrotor UAV flight [18] in order to satisfy the desired performances. The main goal of the proposed RASMC was to keep the quadrotor behaviors intact from the impact of uncertainties and disturbances. The effectiveness of the proposed method was validated through simulation experiments and shows high performances compared to the conventional sliding mode controller (CSMC) and the proportional-integral-derivative controller (PID).

Robust adaptive sliding mode control (RASMC) has emerged as an effective approach for enhancing trajectory tracking in quadrotor UAVs, particularly in the presence of model uncertainties and external disturbances. This method integrates the robustness of sliding

mode control with adaptive mechanisms to dynamically adjust control parameters, ensuring accurate tracking performance even under varying flight conditions. RASMC forces the system to adhere to a predefined sliding surface, thus mitigating the effects of uncertainties and disturbances, while the adaptive component continuously fine-tunes the control parameters to address real-time variations in the UAV's dynamics. Islam et al. (2015) [19] demonstrates that RASMC significantly improves trajectory tracking accuracy and stability compared to conventional control methods, making it a powerful tool for achieving precise and reliable quadrotor performance in complex environments.

Quaternion-based position control for quadrotor UAVs, when combined with robust nonlinear third-order sliding mode control (SMC) and disturbance cancellation, offers a sophisticated solution for enhancing control precision and robustness. This approach leverages quaternions to represent the UAV's orientation, avoiding singularities and providing smooth rotational control. The third-order sliding mode control improves the system's response by considering higher-order dynamics and effectively addressing nonlinearities. Additionally, the integration of disturbance cancellation techniques ensures that external perturbations and unmodeled dynamics have minimal impact on the control performance. Sanwale et al. (2020) [37] demonstrates that this robust nonlinear control strategy significantly enhances position control accuracy and stability, making it well-suited for complex and dynamic operational environments.

Eltayeb et al. (2022) [13], Integral Adaptive Sliding Mode Control (IASMC) is proposed to enhance the performance of quadcopter UAVs under varying payloads and external disturbances. The IASMC approach integrates integral action into the adaptive sliding mode control framework, which allows the system to effectively manage changes in payload and address disturbances with improved robustness. The integral component helps in eliminating steady-state errors and enhancing overall control accuracy, making the system more resilient to fluctuations in payload and environmental conditions. The effectiveness of the IASMC method was validated through simulations, which demonstrated its superior performance in maintaining stable and accurate control compared to traditional control strategies. This approach provides a robust solution for managing dynamic and uncertain conditions in

quadcopter operations, offering significant improvements in control precision and adaptability.

Nguyen et al. (2021) proposed an Adaptive Sliding Mode Controller (ASMC) enhanced with neural network techniques for managing the attitude and altitude control of quadcopter UAVs [31]. The proposed control strategy integrates an adaptive mechanism with sliding mode control to handle uncertainties and external disturbances effectively. The neural network component is employed to learn and adjust control parameters in real-time, improving the system's ability to maintain desired performance despite dynamic changes. The study demonstrated through extensive simulations that the ASMC significantly outperforms conventional sliding mode controllers (CSMC) and proportional-integral-derivative (PID) controllers, showcasing enhanced stability and accuracy in quadcopter flight control. This approach addresses the limitations of traditional methods by providing robust performance and adaptability, validating its effectiveness in complex flight scenarios.

Adaptive neural network-based sliding mode control (NN-SMC) represents an advanced approach to managing the altitude of quadrotor UAVs, combining the strengths of neural networks and sliding mode control. This technique integrates a neural network to adaptively learn and approximate the nonlinear dynamics of the quadrotor, thereby enhancing the control system's ability to handle uncertainties and external disturbances. Sliding mode control provides robustness by enforcing system trajectories to adhere to a predefined sliding surface, while the neural network dynamically adjusts the control parameters to optimize performance and stability. Razmi et al. (2018) [33] demonstrates the effectiveness of NN-SMC in maintaining precise altitude control in the presence of model uncertainties and changing flight conditions, highlighting its potential to improve the reliability and adaptability of quadrotor UAV systems in complex environments.

Analyzing the existing literatures thoroughly, for the quadrotor UAV control of a nonlinear model, SMC is becoming a more robust and advanced control method. However, there are two main problems to overcome when using the SMC method. The first problem is dealing the effects of unknown internal and external disturbances on the system, which can be solved using adaptive and fuzzy control respectively. The second problem is the chattering and it is solved through the fuzzy, and higher order SMC [15, 33, 44].

Therefore, ASMC is developed for the quadrotor UAV control designed for both high and low level control loops of quadrotor UAV that handles both disturbance and uncertainty.

2.3 State Estimation and Noise Reduction

State estimation and noise reduction are crucial aspects of quadrotor UAV operation, ensuring accurate and reliable control performance. State estimation involves using sensors and algorithms to estimate the internal states of the UAV, such as position, velocity, and orientation, which are not directly measurable [14, 23, 44]. Techniques of Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) are commonly employed to fuse sensor data and provide accurate state estimates despite measurement noise and system uncertainties. Noise reduction is equally important, as sensor measurements often include various sources of noise that can degrade control performance and stability. Methods such as adaptive filtering and smoothing algorithms help to mitigate the impact of noise, ensuring that the state estimates remain reliable and the control system performs optimally. Effective state estimation and noise reduction are essential for enhancing the precision and robustness of quadrotor UAVs, particularly in complex and dynamic environments where accurate data is critical for successful operation. In this thesis EKF is utilized for the quadrotor attitude estimation and noise cancellation method.

Chapter 3

Quadrotor UAV Mathematical Modeling

3.1 Working Principle

Quadrotor UAVs, also known as quadcopters, operate using four rotors arranged in a cross configuration of different angle between axis along the flight direction [8, 26]. Each rotor generates lift and torque, which allow the UAV to maneuver in three-dimensional space. The basic principles governing the operation of quadrotor UAVs involve the generation of thrust, torque, and the subsequent control of the vehicle's attitude and position.

The mechanical design of the quad-rotor features four identical and equally spaced rotors, each with a propeller directly coupled to the rotor's shaft. The propellers are arranged in counter-rotating pairs, meaning that each opposing pair of rotors spins in the same direction [31, 43]. The motor-propeller system is responsible for generating both thrust and torque. The thrust force produced by the propellers acts along the axes of the motors, directed from the motors to the propellers, as illustrated in Figure 3.1. Opposing this propulsion force is the gravitational force, which always points towards the Earth's center of gravity. It is important to note that the thrust forces are generated at the rotating tips of the quad-rotor. This, in turn, creates turning moments known as roll, pitch, and yaw torques, which occur around the vehicle's center of gravity, critical for the three channel attitude control.

Further, each rotating propeller generates reaction torques along its axis, opposite to the direction of the produced torques. Proper arrangement and distribution of the propeller forces and the generated turning moments are crucial for the quad-rotor's navigation [24]. Hence, in this thesis, a standard model-based design approach is used and the analysis of the

quad-rotor control system starts with the adoption of a high-fidelity nonlinear model. This model captures the intricate dynamics of the quad-rotor by incorporating detailed descriptions of all relevant parameters. To ensure the model's accuracy and reliability, it is verified through a series of carefully selected test cases. These tests are designed to confirm that the model's behavior aligns closely with real-world performance.

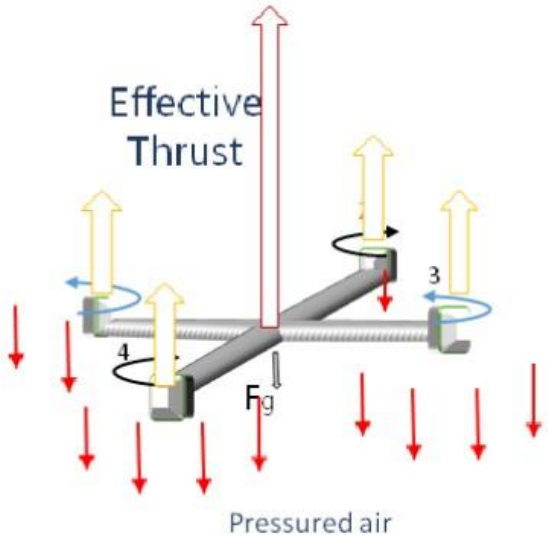


Figure 3.1: Thrust Generation and Weight in Quadrotor

3.2 Quadrotor UAV Configuration Setup

Quad-rotors feature various construction schemes, each providing a range of functional capabilities. The overall motion of a quad-rotor is primarily driven by the action of its propellers [3]. Depending on the orientation of the blades relative to the body coordinates of the vehicle, quad-rotors can be broadly classified into different categories based on their design and functionality of blades as follows.

The X-configuration for quad-rotors arranges the four rotors in an "X" shape, with rotors positioned at the ends of the arms extending from the center of the frame. This setup is known for its aerodynamic efficiency and is commonly used in both commercial and hobbyist drones [34]. It strikes a good balance between stability and maneuverability, making it a popular choice for many applications.

The Plus (+) configuration positions the rotors in a "+" shape, with each rotor mounted at the end of an arm extending outward from the central body [38]. This arrangement simplifies control algorithms due to its symmetric rotor placement, making it suitable for drones requiring straightforward control dynamics. The Plus configuration is often utilized in research and experimental drones due to its simplicity.

The Coaxial configuration stacks two rotors on the same axis, with one rotor spinning in the opposite direction of the other. This design reduces the overall size of the drone while enhancing the thrust-to-weight ratio. The coaxial setup improves aerodynamic efficiency but can complicate control due to the interactions between the stacked rotors.

The H-configuration features rotors arranged in an "H" shape, with two sets of rotors placed on either side of the central body [7]. This less common configuration offers unique stability and control benefits depending on the design. Though not as prevalent as the X and Plus configurations, it can be advantageous for specific applications requiring such an arrangement. Finally, some quad-rotors are designed with an omnidirectional configuration, incorporating additional rotors or mechanisms to achieve movement in any direction without altering the drone's orientation. This advanced setup involve complex control systems and additional rotor types or gimbal mechanisms, making it suitable for specialized robotics and high-precision applications.

3.3 Reference Frame Definitions

Modeling of the quadrotor dynamics, involves the definition of the reference coordinate frames of the system. There are two coordinate frames, as shown in Figure 3.2, that the quadrotor will operate in: inertial frame (x_i,y_i,z_i) and body frame (x_b,y_b,z_b) [27]. Inertial Frame (IF) is defined with respect to the ground with a positive z-axis pointing in the opposite direction of gravity, whereas Body Frame (BF) is defined with respect to the Center of Gravity (CoG) of the quadrotor with its axes fixed to the body. The necessity of defining those reference frames is because of:

- Aerodynamics forces and torques are manipulated with respect to BF

- On-board builtin sensors measure quantities with their respective reference frame, for instance, Inertial Measurement Unit (IMU) gives readings with respect to BF whereas GPS gives readings with respect to IF and Newton laws are valid in IF.

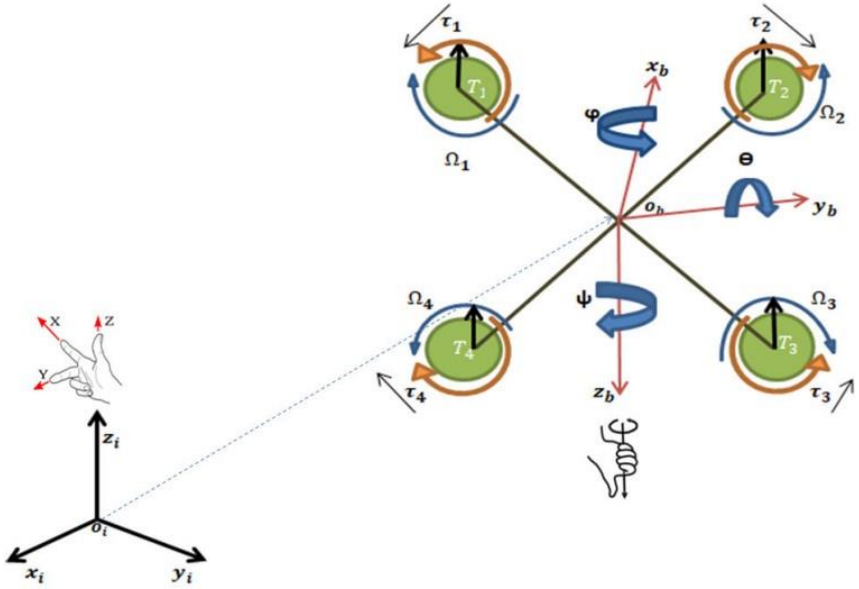


Figure 3.2: Reference Frame Definition

3.4 Quaternion Based Modeling

The first step in designing control schemes for dynamical systems is to develop a mathematical model [27]. This model should accurately reflect the behavior and operation of the system in real-world scenarios. For quad-rotors, modeling approaches are based on the underlying physics of the system. The analysis involves breaking down the entire system into smaller subsystems to facilitate easier analysis and design. In this section, Newton and Quaternion modeling approach is used to derive the equations of motion that describe both the dynamics and kinematics of the quadrotor.

Every mathematical modeling seeks simplification or engineering assumptions to the real physical system. So do so in this paper the following engineering assumptions are considered, [10, 18, 19, 27, 32, 33].

- The saturation and parameter changes are neglected.

- The quad-rotor has a symmetrical structure, resulting in a diagonal inertia matrix about this symmetry to exist only i.e. (J_x,J_y,J_z).
- The propellers of the quad-rotor are considered rigid.
- The center of gravity (CoG) of the vehicle is assumed to coincide with the origin of the body reference frame.
- The physical structure of the quad-rotor is treated as a rigid frame equipped with four rotors.
- The vehicle's mass is assumed to be constant and does not change during motion.
- Aerodynamic effects, such as blade flapping and non-zero free stream velocity, are disregarded for the purpose of the analysis.

3.4.1 Coordinate Transformation

The two reference frames were defined in the previous section Figure 3.2. Hence coordinate frame transformations are essential for converting quantities between different reference frames. To circumvent the issues of nonlinearity, geometric singularities, and high computational costs typically associated with Euler angles [15, 27], the quaternion method is utilized for these transformations.

Quaternions were first introduced by Hamilton as an extension of two-dimensional complex numbers into three-dimensional space. Unlike Euler angles, which describe attitude using multiple angles and axes, quaternion representation uses a single axis and an angle of rotation to describe orientation in three dimensions [25]. Although quaternions may lack the intuitive clarity provided by Euler angles, they are advantageous in avoiding singularity issues, making them valuable for certain applications. Quaternions offer a computationally efficient way to describe three-dimensional rotations [1].

A quaternion is defined as a hyper-complex number consisting of four dimensions: a scalar component q_o and a three-dimensional vector component $\vec{q} = [q_i,q_j,q_k]$, with orthonormal bases $[i,j,k]$ [19]. Therefore, the quaternion expressed by equation 3.1.

$$q = q_0 + q_1\hat{i} + q_2\hat{j} + q_3\hat{k} = [q_0, q_1, q_2, q_3] = (q_0, \vec{q}) \tag{3.1}$$

To accurately represent a 3D orientation or compute valid vector rotations about different axis, a quaternion must be normalized [16, 17]. Normalizing the quaternion ensures it is ideally suited for representing and transforming coordinates between different reference frames. Quaternion normalization can be performed as described in 3.2.

$$q = \frac{q_0 + q_1\hat{i} + q_2\hat{j} + q_3\hat{k}}{\sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}} \tag{3.2}$$

A unit quaternion provides a convenient mathematical notation for representing orientations in 3D space. Thus, the concept of the unit quaternion is used to derive the quadrotor's dynamic model. The mathematical algebra of two quaternions, $q_a = (q_{oa}, \vec{q}_a)$ and $q_b = (q_{ob}, \vec{q}_b)$, is illustrated as follows, including several key operations.

The product of two quaternions q_a and q_b is given by equation 3.3:

$$q_a \cdot q_b = (q_{oa}q_{ob} - \vec{q}_a \cdot \vec{q}_b,\ q_{oa}\vec{q}_b + q_{ob}\vec{q}_a + \vec{q}_a \times \vec{q}_b) \tag{3.3}$$

where $\vec{q}_a \cdot \vec{q}_b$ is the dot product and $\vec{q}_a \times \vec{q}_b$ is the cross product of the vectors.

The conjugate of a quaternion $q_a = (q_{oa}, \vec{q}_a)$ is given by equation 3.4:

$$q_a^* = (q_{oa}, -\vec{q}_a) \tag{3.4}$$

The conjugate is used to compute the inverse of the quaternion.

A quaternion $q_a = (q_{oa}, \vec{q}_a)$ is normalized by dividing by its norm. The norm $\|q_a\|$ is given by 3.5 where as the normalized quaternion is 3.6.

$$\|q_a\| = \sqrt{q_{oa}^2 + \|\vec{q}_a\|^2} \tag{3.5}$$

$$\hat{q}_a = \frac{q_a}{\|q_a\|} \tag{3.6}$$

The inverse of a quaternion q_a is given by 3.7.

$$q_a^{-1} = \frac{q_a^*}{\|q_a\|^2} \tag{3.7}$$

These operations are fundamental for working with quaternions in 3D space and are essential for applications in robotics, computer graphics, and aerospace engineering.

3.4.2 Rotation Transformation Matrices in Space

The concept of rotation using quaternions is based on the idea that any point or coordinate frame can be rotated from an arbitrary initial orientation to a desired final orientation using a single rigid rotation around a unit-length axis $\mathbf{u} = u_x\mathbf{i}+u_y\mathbf{j}+u_z\mathbf{k}$ by an angle θ [?]. Recalling unit quaternion, in three-dimensional (3D) space, a vector $\mathbf{v}$ can be represented as a pure quaternion 3.8, where $v_0 = 0$.

$$\mathbf{v} = 0 + v_1\mathbf{i} + v_2\mathbf{j} + v_3\mathbf{k} \tag{3.8}$$

To find the rotated version of the vector, $\mathbf{v}'$, quaternion multiplication is used. This involves pre-multiplying and post-multiplying by the rotation quaternion q and its inverse q^{-1}, or equivalently by q and its transpose q^T . This procedure is given by equation 3.9 and illustrated with Figure 3.3.

$$\mathbf{v}' = q\mathbf{v}q^{-1} = q^{-1}\mathbf{v}q = q\mathbf{v}q^T = q^T\mathbf{v}q \tag{3.9}$$

Figure 3.3: Rotation in Quaternion Operation

Furthermore, by applying the matrix expansions for quaternion multiplication described in earlier, the quaternion rotation operation between the two reference frames of inertial (IF)

and body frame (BF) is expressed as a rotation matrix of single instant operation, details listed at Appendix ??.

The rotation matrix R_E^B from IF to BF is given by equation 3.4.2 [27].

$$R_{EB} = \begin{bmatrix} 2(q_0^2 + q_1^2) - 1 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & 2(q_0^2 + q_2^2) - 1 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & 2(q_0^2 + q_3^2) - 1 \end{bmatrix}$$

The inverse rotation matrix R_B^E is given by 3.4.2.

$$R_B^E = \begin{bmatrix} 2(q_0^2 + q_1^2) - 1 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & 2(q_0^2 + q_2^2) - 1 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & 2(q_0^2 + q_3^2) - 1 \end{bmatrix}$$

3.5 Rigid Body Mathematical Modeling

There are a kind of different mathematical modeling approaches for the quadrotor UAV. These are classical mechanics of Newtonian and Lagrangian approaches, Euler and Quaternion. In this thesis the Newton-Euler method of modeling approached with enhanced attitude orientation description by Quaternion [17, 27, 30, 37]. In this process the UAV is assumed tobe a rigid body of finite unit in space.

3.5.1 Kinematic Modeling of Quadrotor UAV

Kinematic modeling of a quadrotor UAV involves defining its position and velocity, using quaternions to represent its orientation, and relating angular velocity to quaternion derivatives. The rotation matrix derived from quaternions is essential for transforming between the body frame and the inertial frame. The kinematics is all about the configuration and positional orientation description of the UAV without including the forces and moments generating the motion [2, 6, 39]. Hence, lets start with defining the kinematic variables, the state variables in this analysis are defined as follows.

- The three linear positions (x,y,z) are defined in IF.
- The linear and angular velocities defined in BF.
- The attitude representations defined in the Quaternion.

By utilizing the quaternion product as described earlier, a 3D rotation matrix $\mathbf{R}^{E}_{B}$ that transforms position vectors from the BF to the IF given by equation 3.10 [15, 21, 27, 33].

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{z}_i \end{bmatrix} = \begin{bmatrix} 2(q_0^2 + q_1^2) - 1 & 2(q_1q_3 + q_0q_2) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 + q_0q_3) & 2(q_0^2 + q_2^2) - 1 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_2 - q_0q_3) & 2(q_2q_3 - q_0q_1) & 2(q_0^2 + q_3^2) - 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (3.10)$$

The ordinary body frame rates of (p,q,r) are projected onto the quaternion rates of (q_0,q_1,q_2,q_3) using equation 3.11 [7, 16, 25, 27].

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \begin{bmatrix} 0 \\ p \\ q \\ r \end{bmatrix} \quad (3.11)$$

This orientation of the quadrotor is also called quaternion representation. A quaternion $\mathbf{q}(t)$ representing the orientation is has $q_0(t)$ the scalar part and $(q_1(t),q_2(t),q_3(t))^T$ the vector part. Where the quaternion must satisfy the normalization condition 3.12 [16, 17].

$$q_0^2(t) + q_1^2(t) + q_2^2(t) + q_3^2(t) = 1 \quad (3.12)$$

3.5.2 Dynamic Modeling of Quadrotor UAV

The dynamic model of the quadrotor is developed residing on two basic principles of equation of motion: Newton for translational equations of motion and Euler for rotational equations of motion [5, 36?].

3.5.2.1 Quadrotor Translational Equations of Motion

The translational dynamics of the quadrotor is developed based on Newton's 2nd law of translational motion. Newton's laws are valid only in the IF, hence the transformation matrix of R_B^E is used to transform the forces defined in the BF to the IF. The Newton's 2nd law in IF is given by equation 3.13 [6, 27, 30, 37].

$$\text{X}_{F_i} = m[\ddot{x}, \ddot{y}, \ddot{z}]^T = F_{thrust} - F_g - F_d \tag{3.13}$$

Where F_{thrust}, the total thrust force along z-axis defined in BF, F_g the gravitational force along negative z-axis defined in IF, F_d the drag force along the three channels in the defined in IF. Now let's see each component force separately as follows.

3.5.2.2 Thrust Generation

Thrust is generated by the principle that all motors generate an upward force along the z-axis direction and denoted with F_{thrust} [20, 28], defined in the BF. It balances the force of gravity i.e. weight of the quadrotor. The thrust generated by each motor of the rotor denoted with T_i is proportional to the square of the rotor angular speed ω_i, where i is the number 1 to 4 of motors given by equation 3.14:

$$T_i = k_t \omega_i^2 \tag{3.14}$$

where k_t is the thrust coefficient. The total thrust denoted with T is the sum of the thrusts produced by all four rotors given by equation 3.15.

$$T = \sum_{i=1}^{4} T_i \tag{3.15}$$

3.5.2.3 Gyroscopic Effect Compensation

Quadrotor has a stronger ground effect up to the altitude of $z = 6R$, so it's considered a gyroscopic effect up to this altitude value [1, 6, 27]. Hence, the region where $z \geq 0.5R$ and $z \leq 6R$ is considered as under Ground Effect (GE), and the region where $z < 0.5R$ and $z > 6R$ is

considered as free from the GE, where GE is typically doesn't exist. Experiments show that at constant power, the GE is given by equation 3.16.

$$f_{GE}(z) = \begin{cases} \frac{1}{\left(\frac{R}{4z}\right)^2 1-\gamma} & \text{for } 0.5R \le z \le 6R \\ 1 & \text{else} \end{cases} \quad (3.16)$$

Hence, including the gyroscopic effect the thrust generation is updated accordingly. Adjust the Thrust Force. This calculated compensation factor to adjust the thrust force is given by equation 3.17.

$$T_{GE}(z) = T \cdot f_{GE}(z) \quad (3.17)$$

3.5.2.4 The Force of Gravity

In a quadrotor, gravity is a fundamental force that affects its vertical motion. The force of gravity acts downward along z-axis and is equal to the weight of the quadrotor [19, 33, 40]. This force is a constant and calculated using the following equation 3.18. This force is defined in the IF and along negative z-axis for further vector dynamic equation derivation.

$$F_g = m \cdot g \quad (3.18)$$

where F_g is the gravitational force (weight) acting on the quadrotor, m is the mass of the quadrotor, and g is the acceleration due to gravity (approximately 9.81m/s^2).

3.5.2.5 The effect of Drag in Quadrotor

Drag is a resistive force that opposes the motion of the quadrotor through the air [9, 12] which is defined in the IF. It plays a significant role in the dynamics of a quadrotor, affecting its speed, energy consumption, and stability. The most common air drag or friction force in IF which is proportional to linear velocity along each axes x-y-z is given by equation 3.19. Where C_d is drag coefficient along each axis.

$$F_d = \begin{bmatrix} C_{dx}\dot{x} \\ C_{dy}\dot{y} \\ \end{bmatrix} \quad (3.19)$$

$$C_{dz}\dot{z}$$

where $(\dot{x}, \dot{y}, \dot{z})$ is the velocity of the quadrotor relative to the air. The drag force increases with the square of the velocity, which means that as the quadrotor accelerates, the drag force grows significantly. This effect can lead to a higher power requirement for maintaining a certain speed or altitude.

Now the individual component forces are separately analyzed. Hence, lets transform the thrust from BF to the IF using transformation matrix R_B^E and simplify the translational equation of motion given in equation 3.13 earlier with state variables acceleration along the three axes of x-y-z.

$$\begin{bmatrix} \\ \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} \ddot{x} \; 0 \; 0 \; C_{dx}\dot{x}$$

$$\begin{bmatrix} \\ \\ \end{bmatrix} \; {}_E \begin{bmatrix} \\ \\ \\ \\ \end{bmatrix} \qquad \begin{bmatrix} \\ \\ \\ \\ \end{bmatrix} \qquad \begin{bmatrix} \\ \\ \\ \\ \end{bmatrix} \tag{3.20}$$

$$m\begin{bmatrix} \\ \ddot{y} \\ \\ \end{bmatrix} = R_B \begin{bmatrix} \\ \\ \\ \end{bmatrix} \; 0 \; \begin{bmatrix} \\ \\ \\ \end{bmatrix} - \begin{bmatrix} \\ \\ \\ \end{bmatrix} \; 0 \; \begin{bmatrix} \\ \\ \\ \end{bmatrix} - \begin{bmatrix} \\ \\ \\ \end{bmatrix} C_{dy}\dot{y} \begin{bmatrix} \\ \\ \\ \end{bmatrix}$$

$$\begin{bmatrix} \\ \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} \ddot{z} \; F_{thrust} \; mg \; C_{dz}\dot{z}$$

Where $(\ddot{x}, \ddot{y}, \ddot{z})$ is the linear acceleration of quadrotor in IF, C_d is air drag or friction force coefficient in IF along the three axes of x-y-z i.e. (C_{dx}, C_{dy}, C_{dz}), g is the acceleration due-to gravity, and m is the mass of the quadrotor. Upon simplification the second order translational equation of motion is given by equation 3.21.

$$\begin{aligned} \ddot{x} &= 2(q_1q_3 - q_0q_2)\frac{F_{thrust}}{m} - \frac{C_{dx}}{m}\dot{x} \\ \ddot{y} &= 2(q_2q_3 + q_0q_1)\frac{F_{thrust}}{m} - \frac{C_{dy}}{m}\dot{y} \\ \ddot{z} &= (2(q_0^2 + q_3^2) - 1)\frac{F_{thrust}}{m} - \frac{C_{dz}}{m}\dot{z} - g \end{aligned} \tag{3.21}$$

3.5.2.6 Quadrotor Rotational Equations of Motion

The torques generated by the rotors are due to aerodynamic drag and gyroscopic effects. The torque τ_i produced by each rotor expressed as equation 3.22 which is defined in BF.

$$\tau_i = k_\tau \omega_i^2 \tag{3.22}$$

where k_τ is the torque coefficient. The total torque acting on the quadrotor is the combination of individual torques from each rotor and it will be about the three axis.

Newton's 2nd law of rotational motion for a fully actuated quadrotor rotational dynamics in BF is given by equation 3.23 [6, 27, 30, 37]. This description is usually called Newton-Euler equation.

$$\text{X}_{\tau_i} = J\alpha = J(\dot{p}, \dot{q}, \dot{r})^T = \tau_i - M_{gy} - F_{aer} \tag{3.23}$$

where J is the inertia matrix of the quadrotor, (p,q,r) is the body angular speed vector, M_{gy} is the moment due to gyroscopic effect, M_{aer} is the moment due to drag force, and α is the rotational acceleration. Newton-Euler equation.

$$J(\dot{p}, \dot{q}, \dot{r})^T = (\tau_1, \tau_2, \tau_3)^T - \omega \times (J\omega) - J_p\Omega \times \omega - (C_{ap}p, C_{aq}q, C_{ar}r)^T \tag{3.24}$$

where J is the moment of inertia matrix, Ω is the net angular velocity vector, and τ is the total torque vector, C_a is drag coefficient. Rearranging equation 3.24, rotational dynamics of the quadrotor is given by equation 3.25.

$$\begin{aligned}
\dot{p} &= \frac{1}{J_x}\tau_1 + \frac{J_y - J_z}{J_x}qr - \frac{J_p}{J_x}q\Omega_r - \frac{C_{ap}}{J_x}p \\
\dot{q} &= \frac{1}{J_y}\tau_2 + \frac{J_z - J_x}{J_y}pr - \frac{J_p}{J_y}p\Omega_r - \frac{C_{aq}}{J_y}q \\
\dot{r} &= \frac{1}{J_z}\tau_3 + \frac{J_x - J_y}{J_z}pq - \frac{C_{ar}}{J_z}rw
\end{aligned} \tag{3.25}$$

This rotational dynamics isn't expressed with what we need of quaternion representation. Hence, to design a controller for a second-order system represented quaternion model, a small quaternion approach is employed to reformulate the quadrotor's rotational dynamics in terms of quaternions [8, 13, 27, 43]. This approach is valid for stabilizing the quadrotor assuming it's near a hovering state. The unit quaternion representation is used to describe the quadrotor's orientation, hence the following conditions are consistently satisfied during near hovering maneuvers: $q_0 \approx 1$ and $q_1 = q_2 = q_3 \approx 0$. The relationship between body frame (BF) rates and quaternion rates expressed as in equation 3.11 becomes simplified as equation 3.26.

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = 0.5 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -p & -p \\ 0 & p & 0 & -p \\ 0 & p & p & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (3.26)$$

Rearranging this equation we will get equation 3.27 and both side time derivative will result equation 3.28.

$$\begin{bmatrix} q_0 \\ q_1 \end{bmatrix} \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \end{bmatrix} \begin{bmatrix} \dot{q}_1 p \\ \dot{q}_2 q \\ \dot{q}_3 r \end{bmatrix} \quad (3.27)$$

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \end{bmatrix} = 0.5 \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \end{bmatrix} \begin{bmatrix} \dot{q}_1 p + \ddot{q}_1 p \\ \dot{q}_2 q \\ \dot{q}_3 r \end{bmatrix}$$

$$\begin{bmatrix} \ddot{q}_0 \\ \ddot{q}_1 \end{bmatrix} = 0.5 \begin{bmatrix} -\dot{q}_1 & -\dot{q}_2 & -\dot{q}_3 \\ \dot{q}_0 & -\dot{q}_3 & \dot{q}_2 \end{bmatrix} \begin{bmatrix} \dot{q}_1 p + \ddot{q}_1 p \\ \dot{q}_2 q \\ \dot{q}_3 r \end{bmatrix} \quad (3.28)$$

Finally, the 2nd order rotational dynamics of the quadrotor in terms of a quaternion expressed by equation 3.29, through substituting equations 3.27 and 3.28 into equation 3.24 and further mathematical simplification.

$$\begin{aligned} \ddot{q}_1 &= \frac{1}{J_x}(0.5\tau_1 - C_{dp}\dot{q}_1) \\ \ddot{q}_2 &= \frac{1}{J_y}(0.5\tau_2 - C_{dq}\dot{q}_2) \\ \ddot{q}_3 &= \frac{1}{J_z}(0.5\tau_3 - C_{dr}\dot{q}_3) \end{aligned} \quad (3.29)$$

The nonlinear and complex mathematical quaternion based models are completed. For the controller and state estimator design the second order dynamic models are used. Hence, lets rewrite the 2nd order system modeling defining the main actual control signals as force of thrust F_{thrust} by U_1, roll channel control moment τ_1 by U_2, pitch channel control moment τ_2 by U_3, and yaw channel control moment τ_3 by U_4. The complete 2nd order dynamic model given by 3.30.

$$
\begin{aligned}
\ddot{x} &= 2(q_1q_3 - q_0q_2)\frac{U_1}{m} - \frac{C_{dx}}{m}\dot{x} \\
\ddot{y} &= 2(q_2q_3 + q_0q_1)\frac{U_1}{m} - \frac{C_{dy}}{m}\dot{y} \\
\ddot{z} &= (2(q_0^2 + q_3^2) - 1)\frac{U_1}{m} - \frac{C_{dz}}{m}\dot{z} - g \\
\ddot{q}_1 &= \frac{1}{J_x}(0.5U_2 - C_{dp}\dot{q}_1) \\
\ddot{q}_2 &= \frac{1}{J_y}(0.5U_3 - C_{dq}\dot{q}_2) \\
\ddot{q}_3 &= \frac{1}{J_z}(0.5U_4 - C_{dr}\dot{q}_3)
\end{aligned}
\tag{3.30}
$$

3.6 State Space Model of Attitude

One of the main tasks in this thesis is the design of state estimators based on Kalman filter and make it resilient to avoid system generated and sensor measurement noise. Hence, the design of Kalman filter both for estimation and noise cancellation relays on the state space model representation of the system. To apply the Kalman filter in MATLAB Simulink it's needed that the conventional state matrices are passed though. Hence conventional A, B, C, and D matrices are needed explicitly in this section. To help this in-progress the model with defined control signals in equation 3.30 are used for each quaternion states state-space modeling.

State space models are a mathematical framework used to describe dynamic systems. The state-space representation comprises matrices A, B, C, and D, which collectively capture the dynamics of the system, the influence of control inputs, and the relationship between the system's internal state and its outputs [7, 12, 38]. The model provides a compact and flexible way to represent linear time-invariant (LTI) systems, making it particularly suitable for analysis and design of control systems. For the given system equations, the state-space form allows to describe the evolution of each attitude state variable and their interactions under the influence of control inputs.

In the design of a Kalman filter, the state-space model plays a crucial role. The Kalman filter is an optimal estimator that combines measurements from sensors with a model of the system's dynamics to estimate the true state of the system [4, 14, 23, 46]. By utilizing the matrices A, B, C, and D, the Kalman filter algorithm predicts the future state and updates the

estimates based on new measurements. This process is essential for applications requiring accurate state estimation in the presence of noise and uncertainties. The state space model provides the necessary structure for implementing the Kalman filter, facilitating robust and efficient state estimation in attitude control of quadrotor UAV in the quaternion approach.

The given system equations are given by equation 3.30, hence it's needed to express each of the states in state space form as follows. Lets define the state variables and inputs for the three attitude channels. For each system q_i the state variables and inputs are defined by equations 3.31, and 3.32 respectively where x_i denotes the quaternion state to be controlled.

$$\begin{aligned} x_{i1} &= q_i \\ x_{i2} &= \dot{q}_i \end{aligned} \tag{3.31}$$

$$u_i = U_{i+1} \quad \text{(assuming } i \text{ ranges from 1 to 3)} \tag{3.32}$$

The state-space form is given by the general form of equation 3.33, where y is the output of each state of quaternion (q_1,q_2,q_3) not the position y here.

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{3.33}$$

For roll channel about x-axis q_1 control, the state variables are defined by equation 3.34 and state space equation given by 3.35.

$$\begin{aligned} x_{11} &= q_1 \\ x_{12} &= \dot{q}_1 \end{aligned} \tag{3.34}$$

$$\begin{aligned} \dot{x}_{11} &= x_{12} \\ \dot{x}_{12} &= \frac{1}{J_x}(0.5U_2 - C_{dp}\dot{q}_1) = \frac{1}{J_x}(0.5U_2 - C_{dp}x_{12}) \end{aligned} \tag{3.35}$$

State-space matrices (for roll channel):

$$A_1 = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{C_{dp}}{J_x} \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 \\ \frac{0.5}{J_x} \end{bmatrix}, \quad C_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad D_1 = \begin{bmatrix} 0 \end{bmatrix} \tag{3.36}$$

For pitch channel about y-axis q_2 control, the state variables are defined by equation 3.37 and state space equation given by 3.38.

$$x_{21} = q_2$$
$$x_{22} = \dot{q}_2 \tag{3.37}$$

$$\dot{x}_{21} = x_{22}$$
$$\dot{x}_{22} = \frac{1}{J_y}(0.5U_3 - C_{dq}\dot{q}_2) = \frac{1}{J_y}(0.5U_3 - C_{dq}x_{22}) \tag{3.38}$$

State-space matrices (for pitch channel):

$$A_2 = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{C_{dq}}{J_y} \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 \\ \frac{0.5}{J_y} \end{bmatrix}, \quad C_2 = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad D_2 = \begin{bmatrix} 0 \end{bmatrix} \tag{3.39}$$

For yaw channel about z-axis q_3 control, the state variables are defined by equation 3.40 and state space equation given by 3.41.

$$x_{31} = q_3$$
$$x_{32} = \dot{q}_3 \tag{3.40}$$

$$\dot{x}_{31} = x_{32}$$
$$\dot{x}_{32} = \frac{1}{J_z}(0.5U_4 - C_{dr}\dot{q}_3) = \frac{1}{J_z}(0.5U_4 - C_{dr}x_{32}) \tag{3.41}$$

State-space matrices (for yaw channel):

$$A_3 = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{C_{dr}}{J_z} \end{bmatrix}, \quad B_3 = \begin{bmatrix} 0 \\ \frac{0.5}{J_z} \end{bmatrix}, \quad C_3 = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad D_3 = \begin{bmatrix} 0 \end{bmatrix} \tag{3.42}$$

3.7 Model Verification of Quadrotor UAV Mathematical Modeling

To validate the mathematical model developed for the quadrotor UAV in Chapter Three, a simulation-based verification was conducted using MATLAB. The verification process evaluates both translational and rotational dynamics by comparing model-predicted outputs with simulated trajectories. This section presents the results of position and attitude verification, with particular emphasis on the use of quaternion-based orientation modeling.

3.7.1 Simulation Framework

The simulation was performed over a 10-second time horizon with a sampling interval of 0.01 seconds. Two distinct motion profiles were applied:

- Translational Motion: Modeled as low-frequency sinusoidal drift in the x and y directions, with constant vertical velocity in z. This simulates hovering with minor drift.
- Rotational Motion: Modeled as oscillatory angular velocities around all three axes, inducing sinusoidal attitude changes.

The quadrotor's orientation was represented using unit quaternions, which offer a singularity-free and computationally efficient method for modeling 3D rotations. Unlike Euler angles, quaternions avoid gimbal lock and provide smooth interpolation of rotational states. In this verification, quaternion propagation was performed using a first-order differential update based on angular velocity inputs. The resulting quaternion states were normalized at each time step and converted to Euler angles (roll, pitch, yaw) for interpretability and visualization.

3.7.2 Position Verification

Figure 3.4 presents the position verification results under translational and rotational motion. Each subplot compares the model-predicted position (solid black line) with the simulation output (dashed red line) across the X, Y, and Z axes.

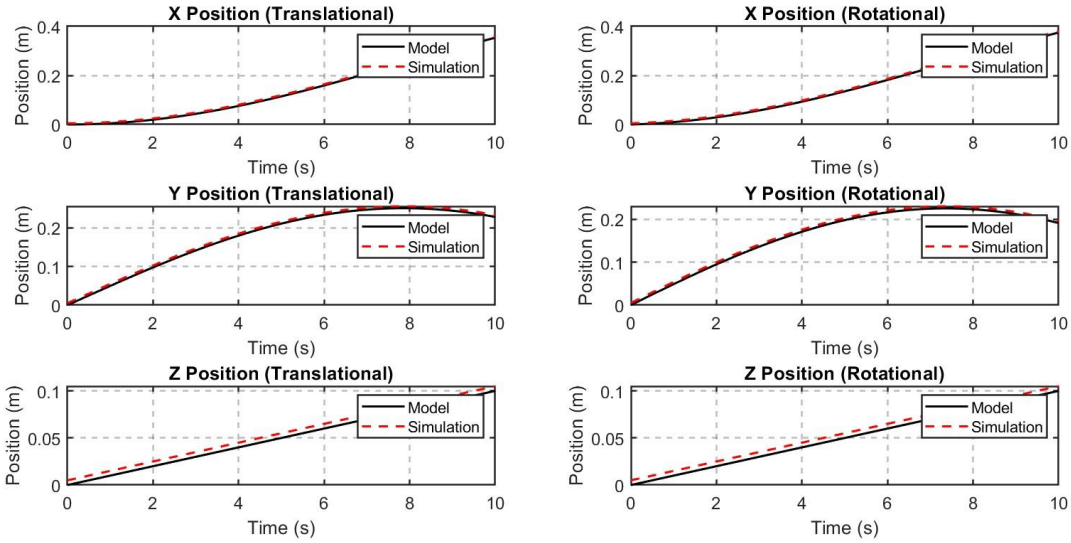


Figure 3.4: Position Verification under Translational and Rotational Motion. Each subplot compares model and simulation outputs for X, Y, and Z positions. The close alignment between curves confirms the fidelity of the translational kinematics.

The results demonstrate excellent agreement between the model and simulation outputs. The translational motion plots show smooth sinusoidal trajectories with minimal deviation, while the rotational motion plots reflect consistent positional behavior despite the presence of angular perturbations. This confirms that the position propagation equations derived in Chapter Three are both accurate and robust under varying motion conditions.

3.7.3 Attitude Verification Using Quaternions

Figure 3.5 illustrates the verification of the quadrotor's attitude dynamics using quaternionbased propagation. The model and simulation outputs are compared for four key parameters:

- Yaw Angle: Reflects heading changes due to rotation around the vertical axis.
- Yaw Rate: Derived from the time derivative of the yaw angle.

- Roll Angle: Represents rotation around the longitudinal axis.
- Pitch Angle: Represents rotation around the lateral axis.

Each parameter is plotted under both translational and rotational motion. The solid black lines represent model outputs, while the dashed red lines denote simulation results.

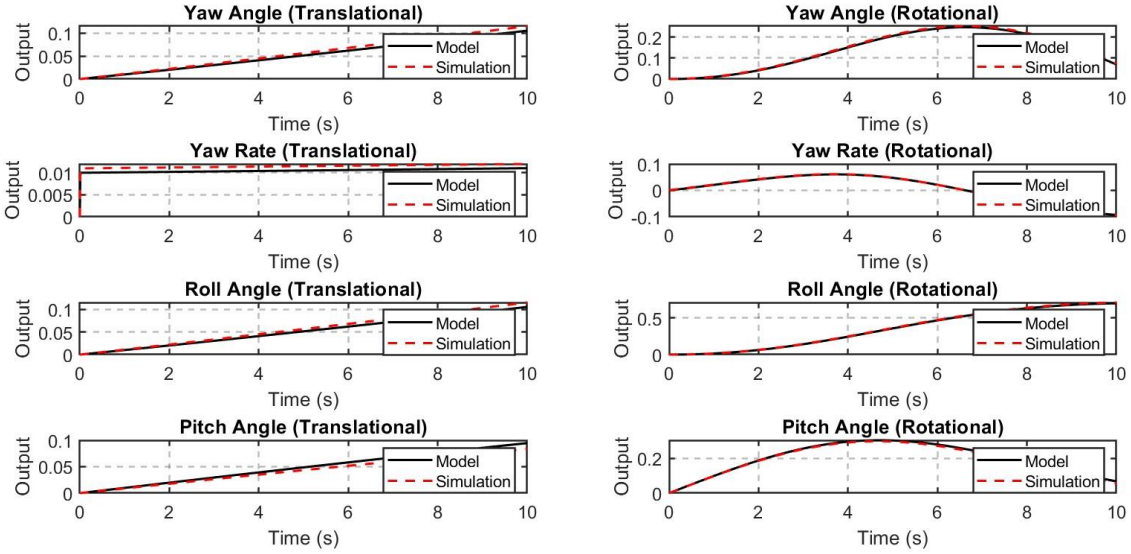


Figure 3.5: Quaternion-Based Attitude Verification under Translational and Rotational Motion. Model and simulation outputs are compared for yaw, roll, and pitch dynamics. The close match across all plots validates the quaternion propagation and attitude modeling.

Under translational motion, the attitude angles exhibit linear growth, consistent with constant angular velocity inputs. Under rotational motion, the angles and rates follow sinusoidal trajectories, reflecting the oscillatory nature of the input angular velocities. The close alignment between model and simulation curves across all subplots confirms the correctness of the quaternion propagation algorithm and the accuracy of the attitude dynamics derived in the modeling phase.

3.7.4 Discussion

The verification results provide strong evidence that the mathematical model accurately captures both the translational and rotational behavior of the quadrotor UAV. The position plots confirm the integrity of the kinematic equations, while the attitude plots validate the quaternion-based orientation tracking and its conversion to Euler angles.

The use of quaternions in this context ensures robust and singularity-free representation of rotational motion, which is critical for UAV dynamics. Their successful application in this verification supports their continued use in subsequent control and estimation algorithms.

3.7.5 Conclusion

This model verification establishes the reliability of the quadrotor UAV mathematical model developed in Chapter Three. The validated position and attitude dynamics — particularly the quaternion-based orientation framework — form a robust foundation for the adaptive sliding mode control (ASMC) strategies and Kalman filtering techniques introduced in later chapters. The simulation framework also provides a reusable testbed for future enhancements and controller validation.

Chapter 4

Quadrotor UAV Controller and State Estimator Design

4.1 State Estimation Design

State estimation is crucial in control systems to accurately infer the internal state of a system from noisy measurements [4, 14, 23]. One popular approach for state estimation is the Kalman filter, which is widely used due to its ability to optimally estimate the state of a linear dynamic system. This section describes the design of the Kalman filter for the attitude subsystem, focusing on the process noise covariance matrix Q and the measurement noise covariance matrix R.

4.1.1 Kalman Filter Overview

The Kalman filter provides a recursive solution to the optimal state estimation problem. It works in two main steps: prediction and correction [14]. During the prediction step, the filter uses the system's dynamic model to predict the state and its uncertainty at the next time step. During the correction step, the filter updates these predictions based on new measurements. The accuracy of the Kalman filter's state estimates depends significantly on the proper selection and tuning of its covariance matrices, Q and R.

The process noise covariance matrix Q represents the uncertainty in the system's

dynamics model [29]. It quantifies the extent to which the actual process deviates from the model predictions due to unmodeled dynamics or external disturbances. For each channel let

the q_1 system, the process noise affects both the position (q_1) and velocity ($\dot{q}_1$) states. Hence, the process noise covariance matrix Q for the q_1 system is given by equation 4.1.

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \tag{4.1}$$

Where:

- Q_{11} is the variance of the process noise affecting the position state q_1. It represents the uncertainty in the model's prediction of the position due to process noise.
- Q_{22} is the variance of the process noise affecting the velocity state $\dot{q}_1$. It reflects the uncertainty in the velocity prediction due to process noise.
- Q_{12} and Q_{21} are the variance of the process noise affecting the cross relation between the position q_1 and velocity state $\dot{q}_1$. This cross relation is usually neglected and will not be considered in this thesis.

Generally the choice of covariance matrix Q values is based on empirical measurements and domain knowledge. Accurate tuning of Q ensures that the filter correctly balances between trusting the model and incorporating new measurements.

The measurement noise covariance matrix R characterizes the uncertainty in the measurements obtained from sensors [4, 29]. It reflects how much the measured values deviate from the true values due to measurement errors. Hence, it directly associated with the measured output value. It has the dimension of the output state variable. So it will be a single constant value of vector with one by one dimension. In each channel only the quaternion positions are needed tobe measured. Therefore, for the q_1 system, where only the position q_1 is measured, the measurement noise covariance matrix R is given by equation 4.2.

$$R = R_1 \tag{4.2}$$

Where: R_1 is the variance of the measurement noise for the position measurement q_1. It represents the uncertainty in the measurement of the position due to sensor inaccuracies or external disturbances. The measurement noise covariance R is crucial for determining the

filter's trust in the measurements [35]. If R is set too high, the filter rely too much on the model predictions and ignore the measurements. Conversely, if R is set too low, the filter overly depend on the noisy measurements, leading to suboptimal performance. Hence, very critical selection and tuning is vital; for control system design. When designing the covariance matrices Q and R, factors to-be considered are: systems with more significant unmodeled dynamics require larger values in Q and high-quality sensors with low noise will have smaller values for R.

4.2 Adaptive Sliding Mode Controller Design

The state space model of the quadrotor UAV is nonlinear, coupled, complex, multivariable, and time varying system. So to handle this problem a robust control scheme is needed. One of these controllers is the Sliding Mode Control (SMC) method, which is a class of variable structure system that targets decreasing the complexity of high-order systems to reduced-order state variables, defined as a sliding function and its derivative [18, 27]. SMC has advantageous for order reduction, disturbance rejection, insensitivity to parameter variations, decoupling design parameters and hence robust, and handles all system non-linearity and complexity.

One of the key features of SMC is its ability to force the system dynamics to slide along a predetermined surface, known as the sliding surface, once the system states reach this surface [10, 21]. This ensures that the system behavior is governed by a lower-dimensional dynamic system, which significantly simplifies the control problem. The design of the sliding surface is critical and is usually based on the desired performance specifications such as stability, convergence rate, and robustness against disturbances.

Moreover, the robustness of SMC comes from its inherent ability to reject disturbances and handle model uncertainties. By driving the system states to the sliding surface and maintaining them there, SMC can effectively counteract the effects of external disturbances and unmodeled dynamics [9, 15]. This makes it particularly suitable for controlling quadrotor UAVs, which are often subjected to unpredictable environmental conditions and payload variations. The discontinuous control action of SMC, however, can lead to chattering, a

phenomenon characterized by high-frequency oscillations. To mitigate this issue, various methods such as boundary layer techniques and higher-order sliding modes have been proposed [7, 9, 18, 21, 27, 38].

Adaptive Sliding Mode Controller (ASMC) further enhances the robustness and performance by dynamically adjusting the control parameters in response to changes in the system and environment [18, 19, 21, 33]. This adaptability is crucial for quadrotor UAVs, which operate in highly dynamic and uncertain environments. The adaptive mechanism designed to estimate the discontinuous switching gains to compensate for unmodeled dynamics, thereby improving the overall system performance [18]. The combination of SMC's robustness with adaptive techniques results a powerful control strategy capable of achieving high precision and stability in complex, and nonlinear system of quadrotors.

Furthermore, the ASMC design offers a robust and efficient solution for the control of quadrotor UAVs. By reducing the system's order, rejecting disturbances, and being insensitive to parameter variations, SMC provides a strong foundation for dealing with the inherent complexities and nonlinearities of quadrotor dynamics. The adaptive component ensures that the controller remains effective even in the face of varying operational conditions, making it an ideal choice for real-world applications where reliability and performance are paramount. Therefore, ASMC is going to-be designed and implemented in the following sections. The cascade control system is given by Figure 4.1.

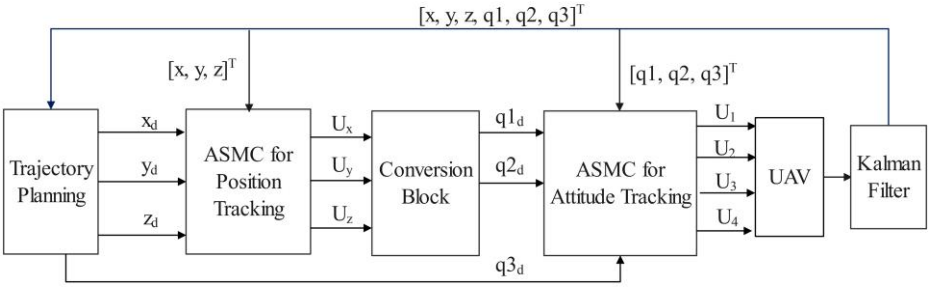


Figure 4.1: Cascade Control System Architecture

The complete mathematical model has two layers of control loops: position and attitude control loops. The control algorithm should guarantee the precise tracking of the desired reference value of the controlled state variables. To design the quadrotor UAV controller the

state variables defined as vector as in equation 4.3. Where the states are linear positions (x,y,z), linear speeds ($\dot{x},\dot{y},\dot{z}$), orientation in quaternion (q_1,q_2,q_3), and quaternion rates along roll, pitch, and yaw direction ($\dot{q}_1,\dot{q}_2,\dot{q}_3$), respectively.

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= \frac{1}{m}(U_x - C_{dx}x_2) \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= \frac{1}{m}(U_y - C_{dy}x_4) \\
\dot{x}_5 &= x_6 \\
\dot{x}_6 &= \frac{1}{m}(U_z - C_{dz}x_6) \\
\dot{x}_7 &= x_8 \\
\dot{x}_8 &= \frac{1}{2J_x}U_2 - \frac{C_{ap}}{J_x}x_8 \\
\dot{x}_9 &= x_{10} \\
\dot{x}_{10} &= \frac{1}{2J_y}U_3 - \frac{C_{aq}}{J_y}x_{10} \\
\dot{x}_{11} &= x_{12} \\
\dot{x}_{12} &= \frac{1}{2J_z}U_4 - \frac{C_{ar}}{J_z}x_{12}
\end{aligned} \tag{4.3}$$

Where the virtual control signals are defined by equation 4.4.

$$\begin{aligned}
U_x &= 2(q_1q_3 - q_0q_2)U_1 \\
U_y &= 2(q_2q_3 + q_0q_1)U_1 \\
U_z &= (2(q_0^2 + q_3^2) - 1)U_1 - mg
\end{aligned} \tag{4.4}$$

From these virtual control commands lets find the desired actual control thrust command and desired quaternion attitudes. To obtain the desired quaternion corresponding to a desired rotation around the z-axis ψ_d, the axis-angle relationship defined in equation 4.5 is utilized [27].

$$q_{3d} = sin(\psi_d) \tag{4.5}$$

Hence, if we do have the desired quaternion about z-axis q_{3d}, we can solve for the desired quaternions q_{1d}, and q_{2d} and the desired thrust force U_1 rearranging equation 4.4. Because we have three known variables of (U_x,U_y,U_z), and q_{3d}, it's possible to solve for the U_1, q_{1d}, and q_{2d}. Through mathematical inversion and simplification of matrices it results the following equation 4.6 for desired quaternions of first and second axis [18, 21, 27].

$$\begin{aligned} q_{0d} &= \frac{1}{\sqrt{2}}\sqrt{\frac{U_z + mg}{\sqrt{U_x^2 + U_y^2 + (U_z + mg)^2}} - 2q_{3d}^2 + 1} \\ q_{1d} &= \frac{q_{3d}U_x + U_y q_{0d}}{U_z + mg + \sqrt{U_x^2 + U_y^2 + (U_z + mg)^2}} \\ q_{2d} &= \frac{q_{3d}U_y - U_x q_{0d}}{U_z + mg + \sqrt{U_x^2 + U_y^2 + (U_z + mg)^2}} \end{aligned} \tag{4.6}$$

The thrust force from this virtual control signals is solved and given by equation 4.7. Where its modified equation is underneath due to the gyroscopic effect from the previous chapter 3.

$$\begin{aligned} U_1 &= \sqrt{U_x^2 + U_y^2 + (U_z + mg)^2} \\ &= \frac{1}{f_{GE}}\sqrt{U_x^2 + U_y^2 + (U_z + mg)^2} \end{aligned} \tag{4.7}$$

Due-to the virtual control inputs, now quadrotor dynamics become fully actuated since there are six control inputs such as U_x,U_y,U_z,U_2,U_3, and U_4 for controlling six degrees of freedom quadrotor maneuvering along x,y,z,q_1,q_2,&q_3 direction, respectively. Hence, a new adaptive reaching law is proposed in this work to enhance the robustness of the SMC against unmodeled dynamics and to exceed the obligatory requirement of disturbance upper limit previous knowledge [19, 33].

4.2.1 Sliding Mode Control Design for Position

The position control starts with designing virtual control U_x design for the position state x on equation 4.8.

$$\begin{aligned} \dot{x}_1 &= x = x_2 \\ \dot{x}_2 &= \ddot{x} = \frac{1}{m}(U_x - C_{dx}x_2) \end{aligned} \tag{4.8}$$

Let the desired position be denoted with x_d then the error dynamics and its derivative are defined by equation 4.9.

$$\begin{aligned} \dot{e}_x &= x_d - x \\ \ddot{e}_x &= \ddot{x}_d - \ddot{x} = \ddot{x}_d - (\frac{1}{m}(U_x - C_{dx}x_2)) \end{aligned} \tag{4.9}$$

The sliding manifold S_x is selected as a function of the error dynamics for trajectory tracking control design as equation 4.10 where $a_1 > 0$ for asymptotic stability.

$$S_x = a_1 e_x + \dot{e}_x \tag{4.10}$$

To design the overall discontinuous and equivalent control U_x, lets consider the constant rate reaching law design. Hence, setting the dynamics of the sliding surface to zero, solve for the controller U_x 4.11.

$$\dot{S}_x = a_1\dot{e}_x + \ddot{e}_x = a_1(\dot{x}_d - \dot{x}) + \ddot{x}_d - (\frac{1}{m}(U_x - C_{dx}x_2)) = -k_1 sign(S_x) \tag{4.11}$$

Solving for U_x it results equation 4.12.

$$U_x = m(a_1(\dot{x}_d - \dot{x}) + \ddot{x}_d) + C_{dx}x_2 + k_1 sign(S_x) \tag{4.12}$$

Now lets design position control U_y for the position state y as described in equation 4.13.

$$\begin{aligned} \dot{x}_3 &= \dot{y} = x_4 \\ \dot{x}_4 &= \ddot{y} = \frac{1}{m}(U_y - C_{dy}x_4) \end{aligned} \tag{4.13}$$

Let the desired position be denoted with y_d; then the error dynamics and its derivative are defined by equation 4.14.

$$\dot{e}_y = y_d - y$$
$$\ddot{e}_y = \ddot{y}_d - \ddot{y} = \ddot{y}_d - \left(\frac{1}{m}(U_y - C_{dy}x_4)\right) \quad (4.14)$$

The sliding manifold S_y is selected as a function of the error dynamics given in equation 4.15, where $a_2 > 0$ ensures asymptotic stability.

$$S_y = a_2 e_y + \dot{e}_y \quad (4.15)$$

To design the overall discontinuous and equivalent control U_y, consider the constant rate reaching law design. Setting the dynamics of the sliding surface to zero, solve for the controller U_y as shown in equation 4.16.

$$\dot{S}_y = a_2\dot{e}_y + \ddot{e}_y = a_1(\dot{y}_d - \dot{y}) + \ddot{y}_d - \left(\frac{1}{m}(U_y - C_{dy}x_4)\right) = -k_2\text{sign}(S_y) \quad (4.16)$$

Solving for U_y, we get:

$$U_y = m(a_2(\dot{y}_d - \dot{y}) + \ddot{y}_d) + C_{dy}x_4 + k_2\text{sign}(S_y) \quad (4.17)$$

The third position control signal U_z for the position state z as described in equation 4.18 will be designed.

$$\dot{x}_5 = \dot{z} = x_6$$
$$\dot{x}_6 = \ddot{z} = \frac{1}{m}(U_z - C_{dz}x_6) \quad (4.18)$$

Let the desired position be denoted with z_d; then the error dynamics and its derivative are defined by equation 4.19.

$$\dot{e}_z = z_d - z$$
$$\ddot{e}_z = \ddot{z}_d - \ddot{z} = \ddot{z}_d - \left(\frac{1}{m}(U_z - C_{dz}x_6)\right) \quad (4.19)$$

The sliding manifold S_z is selected as a function of the error dynamics for trajectory tracking control design, as given in equation 4.20, where $a_3 > 0$ ensures asymptotic stability.

$$S_z = a_3 e_z + \dot{e}_z \tag{4.20}$$

To design the overall discontinuous and equivalent control U_z, consider the constant rate reaching law design. Setting the dynamics of the sliding surface to zero, solve for the controller U_z as shown in equation 4.21.

$$\dot{S}_z = a_3 \dot{e}_z + \ddot{e}_z = a_3(\dot{z}_d - \dot{z}) + \ddot{z}_d - \left(\frac{1}{m}(U_z - C_{dz} x_6)\right) = -k_3 \text{sign}(S_z) \tag{4.21}$$

Solving for U_z, we get:

$$U_z = m(a_3(\dot{z}_d - \dot{z}) + \ddot{z}_d) + C_{dz} x_6 + k_3 \text{sign}(S_z) \tag{4.22}$$

4.2.2 Sliding Mode Control Design for Quaternion

The attitude control in quaternion form starts with designing a torque control signal about the x-axis, U_2 for the quaternion state q_1 as described in equation 4.23.

$$\begin{aligned} \dot{x}_7 &= \dot{q}_1 = x_8 \\ \dot{x}_8 &= \ddot{q}_1 = \frac{1}{2J_x} U_2 - \frac{C_{ap}}{J_x} x_8 \end{aligned} \tag{4.23}$$

; then the error dynamics

Let the desired quaternion about the x-axis be denoted with q_{1d} and its derivative are defined by equation 4.24.

$$\begin{aligned} \dot{e}_{q1} &= \dot{q}_{1d} - \dot{q}_1 \\ \ddot{e}_{q_1} &= \ddot{q}_{1d} - \ddot{q}_1 = \ddot{q}_{1d} - (\frac{1}{2J_x}U_2 - \frac{C_{ap}}{J_x}x_8) \end{aligned} \tag{4.24}$$

The sliding manifold S_{q1} is selected as a function of the error dynamics for trajectory tracking control design, as given in equation 4.25, where $a_4 > 0$ ensuring asymptotic stability.

$$S_{q1} = a_3 e_{q1} + \dot{e}_{q1} \tag{4.25}$$

To design the total discontinuous and equivalent control U_2, consider the constant rate reaching law design. Setting the dynamics of the sliding surface to zero, solve for the controller U_2 as shown in equation 4.26.

$$\dot{S}_{q_1} = a_4\dot{e}_{q_1} + \ddot{e}_{q_1} = a_4(\dot{q}_{1d} - \dot{q}_1) + \ddot{q}_{1d} - \left(\frac{1}{2J_x}U_2 - \frac{C_{ap}}{J_x}x_8\right) = -k_4\text{sign}(S_{q_1}) \tag{4.26}$$

Solving for U_2, we get:

$$U_2 = 2J_x\left(a_4(\dot{q}_{1d} - \dot{q}_1) + \ddot{q}_{1d}\right) + 2C_{ap}x_8 + k_4\text{sign}(S_{q1}) \tag{4.27}$$

The second attitude control in quaternion form is designing a torque control signal about the y-axis, U_3 for the quaternion state q_2 as described in equation 4.28.

$$\begin{aligned} \dot{x}_9 &= \dot{q}_2 = x_{10} \\ \dot{x}_{10} &= \ddot{q}_2 = \frac{1}{2J_y}U_3 - \frac{C_{aq}}{J_y}x_{10} \end{aligned} \tag{4.28}$$

Let the desired quaternion about the y-axis be denoted with q_{2d} and its derivative are defined by equation 4.29.

; then the error dynamics

$$\dot{e}_{q_2} = \dot{q}_{2d} - \dot{q}_2$$
$$\ddot{e}_{q_2} = \ddot{q}_{2d} - \ddot{q}_2 = \ddot{q}_{2d} - \left(\frac{1}{2J_y}U_3 - \frac{C_{aq}}{J_y}x_{10}\right) \tag{4.29}$$

The sliding manifold S_{q2} is selected as a function of the error dynamics for trajectory tracking control design, as given in equation 4.30, where $a_5 > 0$ ensures asymptotic stability.

$$S_{q2} = a_5 e_{q2} + \dot{e}_{q2} \tag{4.30}$$

To design the total discontinuous and equivalent control U_3, consider the constant rate reaching law design. Setting the dynamics of the sliding surface to zero, solve for the controller U_3 as shown in equation 4.31.

$$\dot{S}_{q_2} = a_5\dot{e}_{q_2} + \ddot{e}_{q_2} = a_4(\dot{q}_{2d} - \dot{q}_2) + \ddot{q}_{2d} - \left(\frac{1}{2J_y}U_3 - \frac{C_{aq}}{J_y}x_{10}\right) = -k_5\text{sign}(S_{q_2}) \tag{4.31}$$

Solving for U_3, we get:

$$U_3 = 2J_y\left(a_5(\dot{q}_{2d} - \dot{q}_2) + \ddot{q}_{2d}\right) + 2C_{aq}x_{10} + k_5\text{sign}(S_{q2}) \tag{4.32}$$

The final and third attitude control in quaternion form is designing a torque control signal about the z-axis, U_4, for the quaternion state q_3 as described in equation 4.33.

$$\dot{x}_{11} = \dot{q}_3 = x_{11}$$
$$\dot{x}_{12} = \ddot{q}_3 = \frac{1}{2J_z}U_4 - \frac{C_{ar}}{J_z}x_{12} \tag{4.33}$$

Let the desired quaternion about the z-axis be denoted with q_{3d} and its derivative are defined by equation 4.34.

$$\dot{e}_{q_3} = \dot{q}_{3d} - \dot{q}_3$$
$$\ddot{e}_{q_3} = \ddot{q}_{3d} - \ddot{q}_3 = \ddot{q}_{3d} - \left(\frac{1}{2J_z}U_4 - \frac{C_{ar}}{J_z}x_{12}\right) \tag{4.34}$$

; then the error dynamics

The sliding manifold S_{q3} is selected as a function of the error dynamics for trajectory tracking control design, as given in equation 4.35, where $a_6 > 0$ ensures asymptotic stability.

$$S_{q3} = a_6 e_{q3} + \dot{e}_{q3} \tag{4.35}$$

To design the total discontinuous and equivalent control U_4, consider the constant rate reaching law design. Setting the dynamics of the sliding surface to zero, solve for the controller U_4 as shown in equation 4.36.

$$\dot{S}_{q3} = a_6 \dot{e}_{q3} + \ddot{e}_{q3} = a_6(\dot{q}_{3d} - \dot{q}_3) + \ddot{q}_{3d} - \left(\frac{1}{2J_z}U_4 - \frac{C_{ar}}{J_z}x_{12}\right) = -k_6 \text{sign}(S_{q3}) \tag{4.36}$$

Solving for U_4, we get:

$$U_4 = 2J_z\left(a_6(\dot{q}_{3d} - \dot{q}_3) + \ddot{q}_{3d}\right) + 2C_{ar}x_{12} + k_6 \text{sign}(S_{q3}) \tag{4.37}$$

4.2.3 Stability Analysis

Stability analysis is crucial in control systems to ensure that the designed controllers will bring the system to a desired state and maintain that state over time [19, 36, 38]. For the torque control signals U_2, U_3, and U_4, which are designed to control the attitude of the system about the x, y, and z axes respectively, verifying stability ensures that the error between the desired and actual states will converge to zero and not diverge. By analyzing the stability of each control law, we confirm that the control signals will effectively drive the system to the desired attitude and maintain stability in the presence of disturbances or variations in system parameters.

The Lyapunov function, a mathematical tool used in this analysis, helps demonstrate whether the chosen control laws will lead to a stable equilibrium [17, 26, 45]. For each control law, we calculate the derivative of the Lyapunov function to ensure it is negative, which indicates that the system's energy is decreasing and the state is moving towards stability. Thus, stability analysis not only validates the effectiveness of the controllers but also provides assurance that the system will operate reliably and robustly in practical scenarios.

4.2.3.1 Stability Analysis for U_2 (Torque Control About the x-Axis) Let the Lyapunov function $V(S_{q1})$ for the roll channel system is:

$$V(S_{q_1}) = \frac{1}{2}S_{q_1}^2 \tag{4.38}$$

To ensure the system is globally asymptotically stable, it is needed to show that $\dot{V} < 0$. Hence, it becomes:

$$\dot{V} = S_{q1}\dot{S}_{q1} \tag{4.39}$$

Substitute $\dot{S}_{q1}$ from the control law it results:

$$\dot{S}_{q_1} = a_3\dot{e}_{q_1} + \ddot{e}_{q_1} = a_3(\dot{q}_{1d} - \dot{q}_1) + \ddot{q}_{1d} - \left(\frac{1}{2J_x}U_2 - \frac{C_{ap}}{J_x}x_8\right) \tag{4.40}$$

Now lets substitute U_2 from the control law:

$$U_2 = 2J_x\,(a_3(\dot{q}_{1d} - \dot{q}_1) + \ddot{q}_{1d}) + 2C_{ap}x_8 + k_5\text{sign}(S_{q1}) \tag{4.41}$$

$$\dot{S}_{q_1} = a_3(\dot{q}_{1d} - \dot{q}_1) + \ddot{q}_{1d} - \left(\frac{1}{2J_x}\left[2J_x\,(a_3(\dot{q}_{1d} - \dot{q}_1) + \ddot{q}_{1d}) + 2C_{ap}x_8 + k_5\text{sign}(S_{q_1})\right] - \frac{C_{ap}}{J_x}x_8\right) \tag{4.42}$$

And we have also,

$$\dot{S}_{q1} = -k_5\text{sign}(S_{q1}) \tag{4.43}$$

Finally the Lyapunov dynamics becomes,

$$\dot{V} = S_{q1}\,(-k_5\text{sign}(S_{q1})) = -k_5|S_{q1}| \tag{4.44}$$

To ensure $\dot{V} < 0$, k_5 must be positive. Hence: $k_5 > 0$.

4.2.3.2 Stability Analysis for U_3 (Torque Control About the y-Axis) Consider the Lyapunov function $V(S_{q2})$ for the pitch channel system:

$$V(S_{q_2}) = \frac{1}{2}S_{q_2}^2 \tag{4.45}$$

To ensure the system is globally asymptotically stable it must be $\dot{V} < 0$.Hence, lets compute $\dot{V}$ as follows:

$$\dot{V} = S_{q2}\dot{S}_{q2} \tag{4.46}$$

Substitute $\dot{S}_{q2}$ from the control law:

$$\dot{S}_{q_2} = a_5\dot{e}_{q_2} + \ddot{e}_{q_2} = a_5(\dot{q}_{2d} - \dot{q}_2) + \ddot{q}_{2d} - \left(\frac{1}{2J_y}U_3 - \frac{C_{aq}}{J_y}x_{10}\right) \tag{4.47}$$

Substitute U_3 from the control law:

$$U_3 = 2J_y\,(a_5(\dot{q}_{2d} - \dot{q}_2) + \ddot{q}_{2d}) + 2C_{aq}x_{10} + k_5\text{sign}(S_{q2}) \tag{4.48}$$

$$\dot{S}_{q_2} = a_5(\dot{q}_{2d} - \dot{q}_2) + \ddot{q}_{2d} - \left(\frac{1}{2J_y}\left[2J_y\,(a_5(\dot{q}_{2d} - \dot{q}_2) + \ddot{q}_{2d}) + 2C_{aq}x_{10} + k_5\text{sign}(S_{q_2})\right] - \frac{C_{aq}}{J_y}x_{10}\right) \tag{4.49}$$

$$\dot{S}_{q2} = -k_5\text{sign}(S_{q2}) \tag{4.50}$$

Finally the Lyapunov function time derivative will be:,

$$\dot{V} = S_{q2}\,(-k_5\text{sign}(S_{q2})) = -k_5|S_{q2}| \tag{4.51}$$

To ensure $\dot{V} < 0$, k_5 must be positive. Hence: $k_5 > 0$

4.2.3.3 Stability Analysis for U_4 (Torque Control About the z-Axis) Let the Lyapunov function $V(S_{q3})$ for the yaw channel system be:

$$V(S_{q3}) = \frac{1}{2}S_{q3}^2 \tag{4.52}$$

To ensure the system is globally asymptotically stable it must satisfy $\dot{V} < 0$.

$$\dot{V} = S_{q3}\dot{S}_{q3} \tag{4.53}$$

Substitute $\dot{S}_{q3}$ from the control law:

$$\dot{S}_{q3} = a_6\dot{e}_{q3} + \ddot{e}_{q3} = a_6(\dot{q}_{3d} - \dot{q}_3) + \ddot{q}_{3d} - \left(\frac{1}{2J_z}U_4 - \frac{C_{ar}}{J_z}x_{12}\right) \tag{4.54}$$

Substitute U_4 from the control law:

$$U_4 = 2J_z\,(a_6(\dot{q}_{3d} - \dot{q}_3) + \ddot{q}_{3d}) + 2C_{ar}x_{12} + k_6\text{sign}(S_{q3}) \tag{4.55}$$

$$\dot{S}_{q3} = a_6(\dot{q}_{3d} - \dot{q}_3) + \ddot{q}_{3d} - \left(\frac{1}{2J_z}\left[2J_z\,(a_6(\dot{q}_{3d} - \dot{q}_3) + \ddot{q}_{3d}) + 2C_{ar}x_{12} + k_6\text{sign}(S_{q3})\right] - \frac{C_{ar}}{J_z}x_{12}\right) \tag{4.56}$$

$$\dot{S}_{q3} = -k_6\text{sign}(S_{q3}) \tag{4.57}$$

As a result;

$$\dot{V} = S_{q3}(-k_6\text{sign}(S_{q3})) = -k_6|S_{q3}| \tag{4.58}$$

To ensure $\dot{V} < 0$, k_6 must be positive. Hence: $k_6 > 0$

Similarly, the stability analysis for the virtual control signals were done, and the switching control signals were selected to be positive for asymptotic stability, i.e. k_1, k_2, k_3 must be positive numbers.

4.3 Adaptive Switching Gains in SMC

Adaptive Sliding Mode Control (SMC) leverages the inherent robustness of sliding mode techniques while incorporating adaptability to deal with uncertainties and varying system dynamics. One of the critical components in SMC is the switching gain, which determines the control action's aggressiveness. To improve the performance and robustness of the control system, fuzzy logic is employed to adaptively adjust the switching gains based on real-time system behavior [19, 31].

Fuzzy logic provides a flexible approach to handle uncertainties and nonlinearity by using linguistic rules and membership functions [13]. In the thesis, fuzzy logic is used to adjust the switching gains dynamically, depending on the system's state and error characteristics. The fuzzy logic system utilizes input variables the sliding surface and its derivative. Based on these inputs, the fuzzy inference system determines the appropriate switching gain to maintain the desired system performance and robustness.

Implementing fuzzy logic for adaptive switching gains in SMC offers several benefits. It enhances the controller's ability to handle uncertainties and variations in system dynamics by adjusting the control effort based on real-time feedback. Additionally, it reduces the risk of excessive chattering, a common issue in traditional SMC approaches, by dynamically adjusting the switching gain. The adaptive nature of the fuzzy-based approach ensures that the system remains robust and performs optimally across different operating conditions.

4.3.1 Fuzzy Inference System Description

The fuzzy inference system named was a Mamdani-type system, version 2.0, designed with two input variables and a single output variable. It employs a total of seven rules to govern the system's behavior. The system uses the minimum method for both conjunction (AND) and implication operations, while the maximum method is applied for disjunction (OR) and aggregation operations. The defuzzification process is carried out using the centroid method, which determines the crisp output value by calculating the center of gravity of the aggregated output fuzzy set. This setup is typical for Mamdani systems, providing a balance between simplicity and interpretability in fuzzy logic control.

The fuzzy system incorporates two input variables, S and $\dot{S}$, and one output variable, k_i, each spanning the range $[-1.5, 1.5]$. Each variable is characterized by seven Gaussian membership functions with a standard deviation (σ) of 0.2123 and centers (c) distributed across the range. For both inputs and the output, these membership functions are labeled from NB (Negative Big) to PB (Positive Big), with centers varying between -1.5 and 1.5. This consistent setup ensures a smooth and nuanced representation of the inputoutput

relationships within the fuzzy logic system. The membership functions for each variable are plotted below 4.2, 4.3, and 4.4.

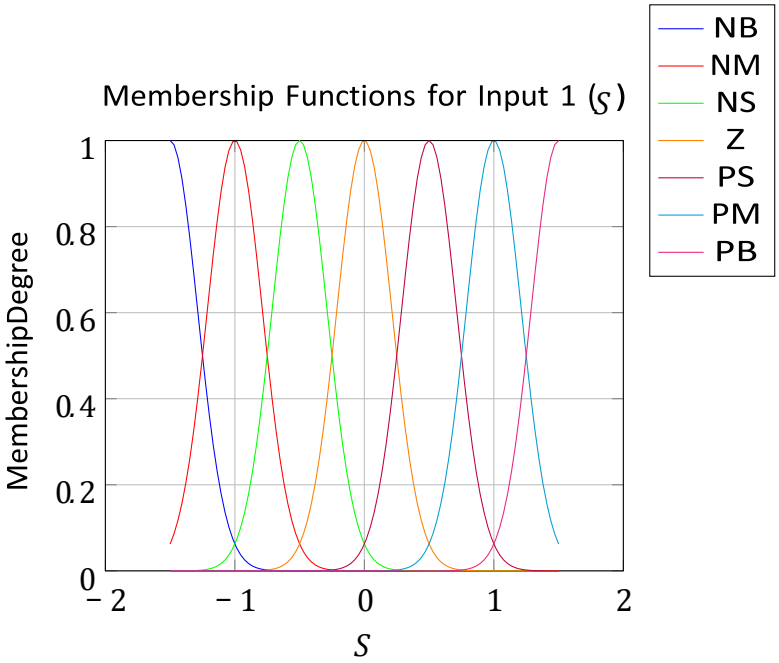


Figure 4.2: Membership Functions for Sliding Surface S.

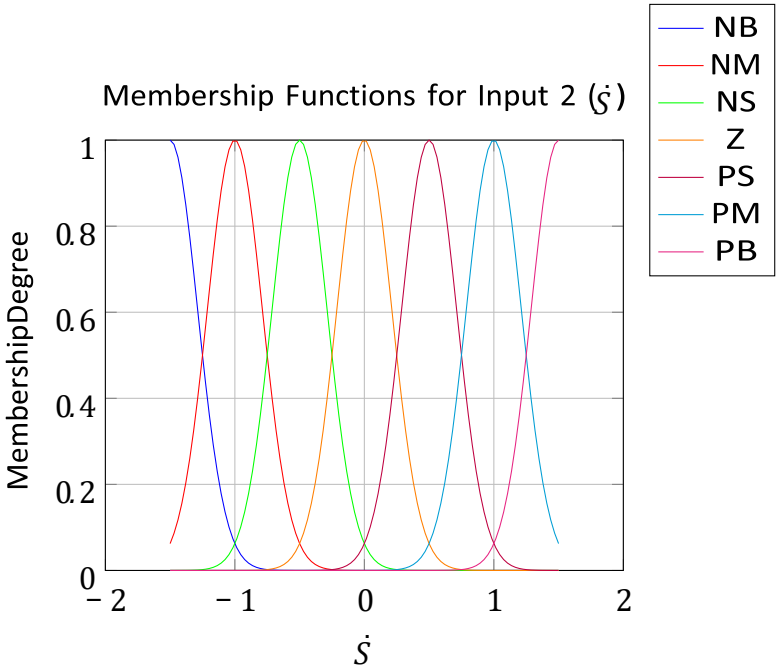


Figure 4.3: Membership Functions for Sliding Surface Derivative $\dot{S}$.

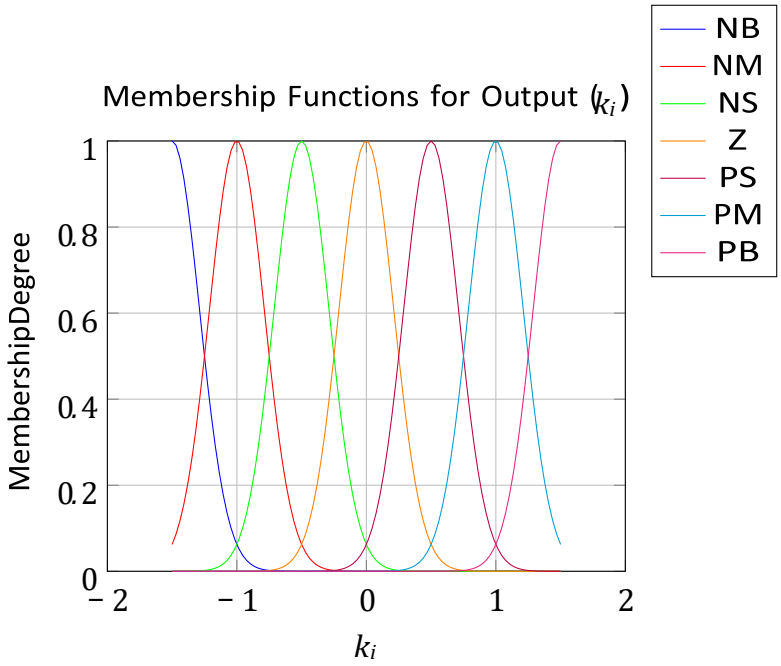


Figure 4.4: Adaptive Switching SMC Gain k_i.

4.4 Enhanced Controller Design with Fuzzy Logic and Kalman Filter

4.4.1 Integrated Control Architecture

The complete control architecture combines ASMC with fuzzy logic adaptation and Kalman filter state estimation as shown in Figure 4.5.

Figure 4.5: Integrated control architecture showing ASMC, fuzzy adaptation, and Kalman filter

4.4.2 Fuzzy Logic Adaptation Mechanism

The fuzzy inference system dynamically adjusts the switching gains (k_i) based on the sliding surface (S) and its derivative ($\dot{S}$). The rule base is designed as:

Table 4.1: Fuzzy rule base for switching gain adaptation

$S\dot{S}$	NB	NM	NS	Z	PS	PM	PB
NB	PB	PB	PM	PM	PS	Z	Z
NM	PB	PB	PM	PS	PS	Z	NS

NS	PM	PM	PM	PS	Z	NS	NS
Z	PM	PM	PS	Z	NS	NM	NM
PS	PS	PS	Z	NS	NS	NM	NM
PM	PS	Z	NS	NM	NM	NM	NB
PB	Z	Z	NM	NM	NM	NB	NB

The membership functions are Gaussian with $\sigma = 0.2123$ and centers distributed across $[-1.5, 1.5]$.

4.4.3 Kalman Filter Implementation

The Kalman filter is implemented for each attitude channel with the following parameters:

$$Q = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}, \quad R = 0.1 \tag{4.59}$$

The state-space model for each channel follows equations (3.36), (3.39), and (3.42) from Chapter 3.

4.5 Simulation Results and Comparative Analysis

4.5.1 Performance with and without Kalman Filter

Figure 4.6 shows the attitude tracking performance comparison.

(a) Roll angle without Kalman filter (b) Roll angle with Kalman filter

Figure 4.6: Comparison of roll angle tracking with and without Kalman filter

The root mean square error (RMSE) values demonstrate the improvement:

Table 4.2: RMSE comparison with and without Kalman filter

Metric	Without KF	With KF
Roll RMSE (rad)	0.085	0.032
Pitch RMSE (rad)	0.079	0.029
Yaw RMSE (rad)	0.092	0.035

4.5.2 Comparison with Other Control Strategies

The ASMC is compared with conventional SMC and PID controllers under the same conditions:The PID controller used for comparative analysis was designed based on conventional tuning methods for quadrotor stabilization as established in [33, 35]. The gains were optimized for nominal flight conditions to provide a standard baseline for evaluating the robustness of the ASMC under external disturbances and model uncertainties Figure 4.7: Performance comparison of ASMC, SMC, and PID controllers

4.5.3 Disturbance Rejection Analysis

The system was tested with wind gusts modeled as:

Table 4.3: Performance metrics comparison of different controllers

Metric	PID	SMC	ASMC
Settling time (s)	4.2	2.8	1.5
Overshoot (%)	12.5	8.3	3.2
RMSE (position)	0.45	0.32	0.18
Chattering index	-	0.25	0.08

$$F_{wind} = 0.5\sin(0.5t) + 0.3\text{randn}(t) \quad [\text{N}] \tag{4.60}$$

Figure 4.8: Disturbance rejection performance of ASMC

The results show that ASMC maintains stable tracking with less than 5% deviation from the reference despite the disturbances.

Chapter 5

Simulation Results and Analysis

This chapter details the simulation results of the Adaptive Sliding Mode Control (ASMC) designed for the position and attitude control of a quadrotor UAV. The primary focus is on the UAV's ability to follow both helical and bowtie trajectories, which are essential for effective desert locust identification and management. Through these trajectories, the performance of the ASMC is rigorously evaluated to demonstrate its capabilities in real-world applications.

5.1 Helical Trajectory Tracking

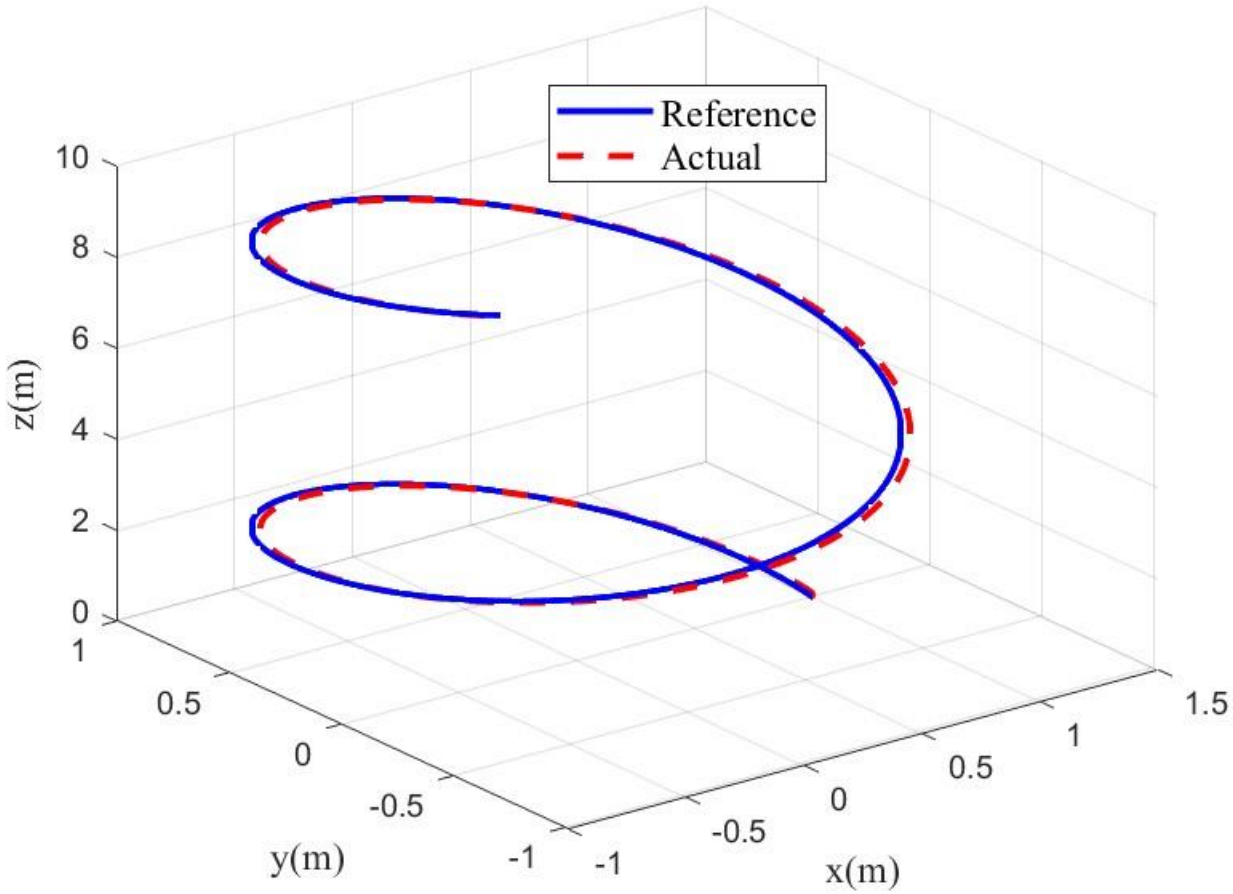
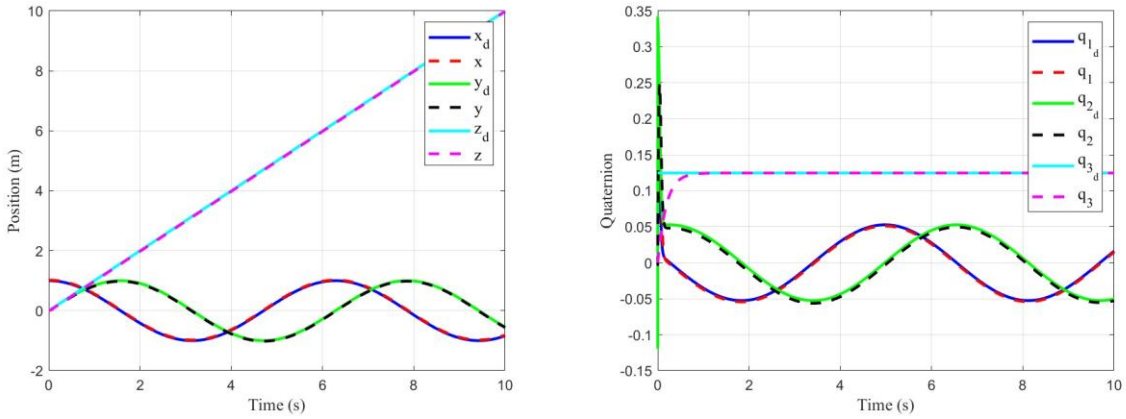


Figure 5.1: 3D Position Plot in Helical Trajectory Tracking



(a) Position Tracking (b) Attitude Tracking

Figure 5.2: Position and Attitude Plot in Helical Trajectory Tracking

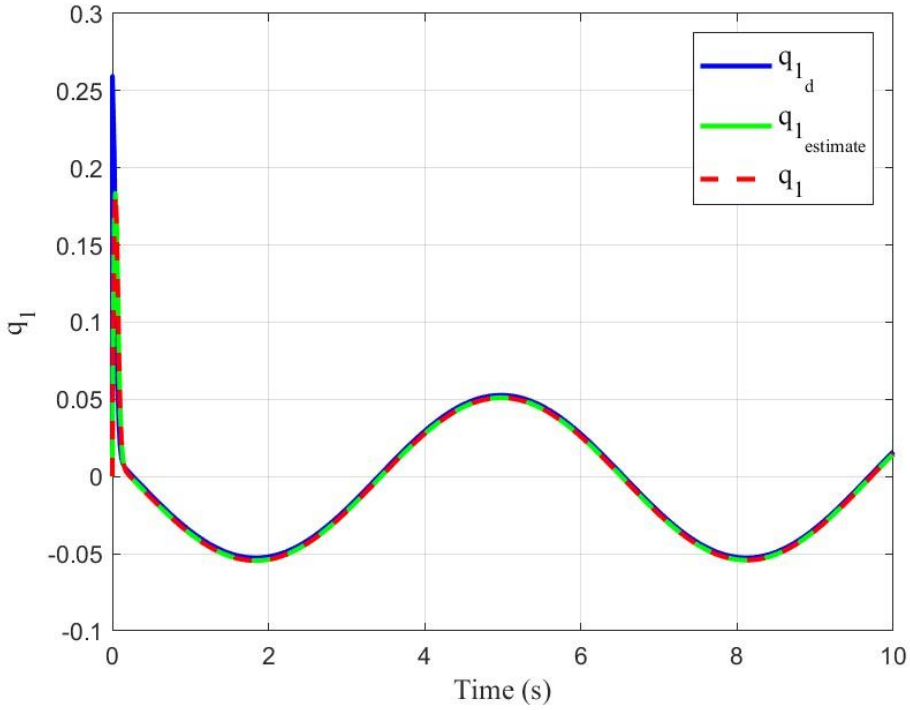


Figure 5.3: Quaternion (q_1) Plot in Helical Trajectory Tracking

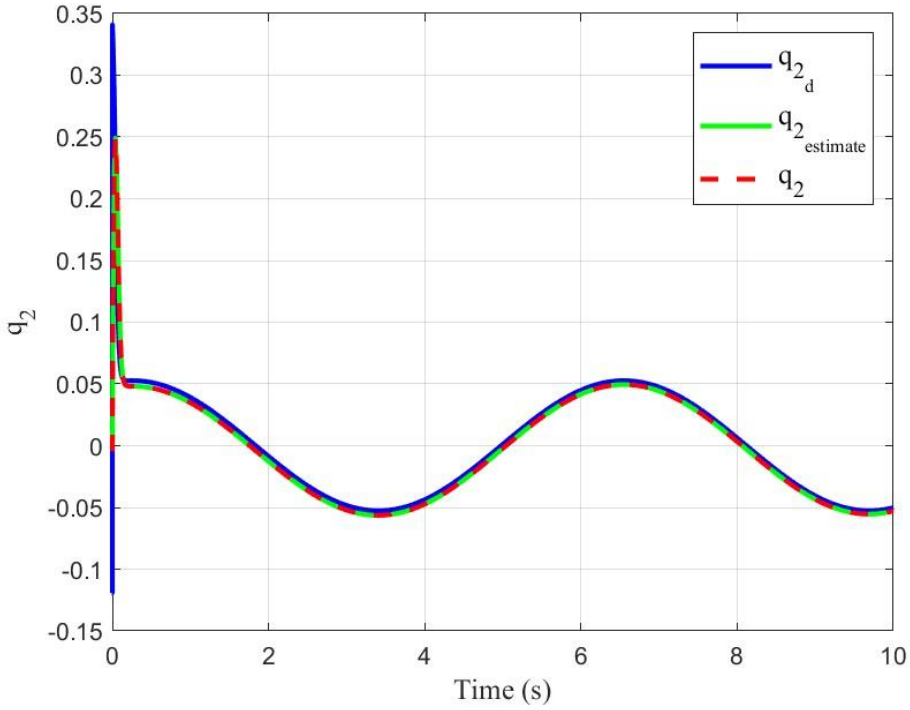


Figure 5.4: Quaternion (q_2) Plot in Helical Trajectory Tracking

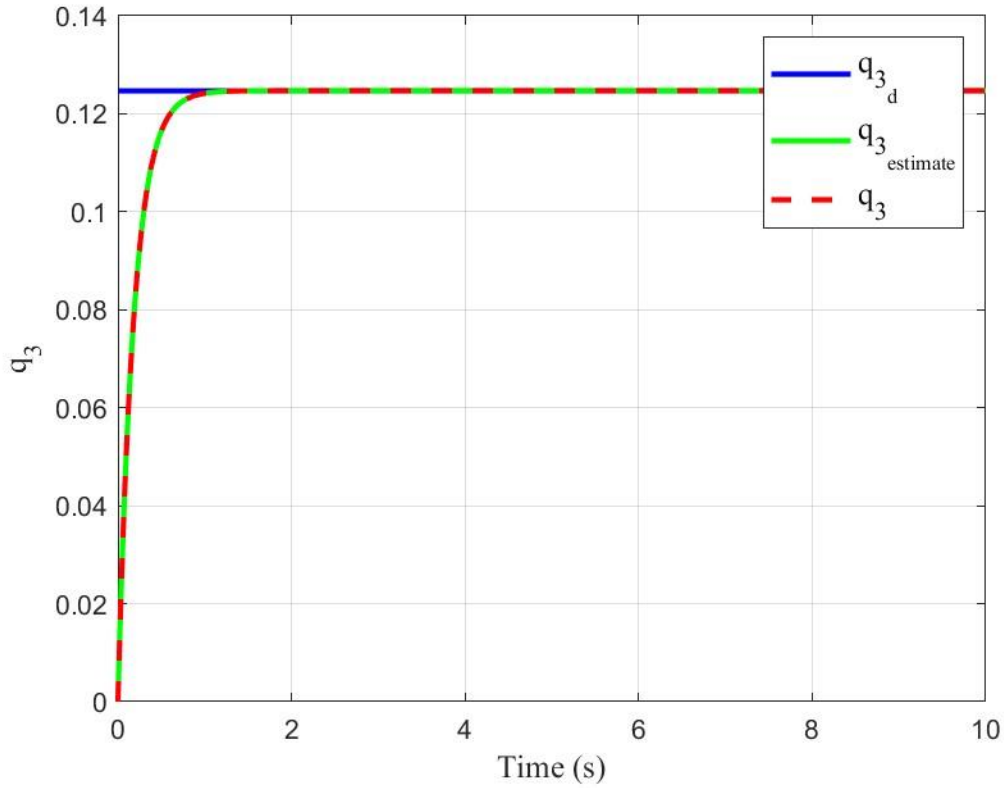
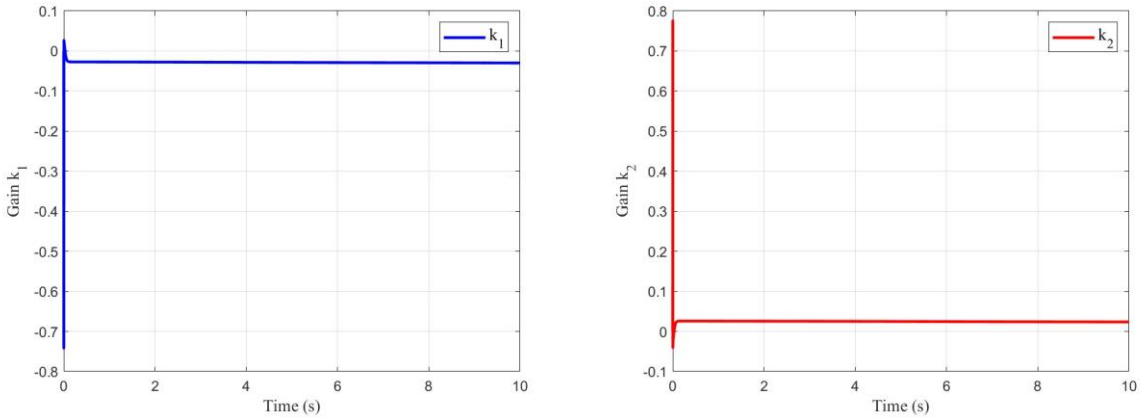
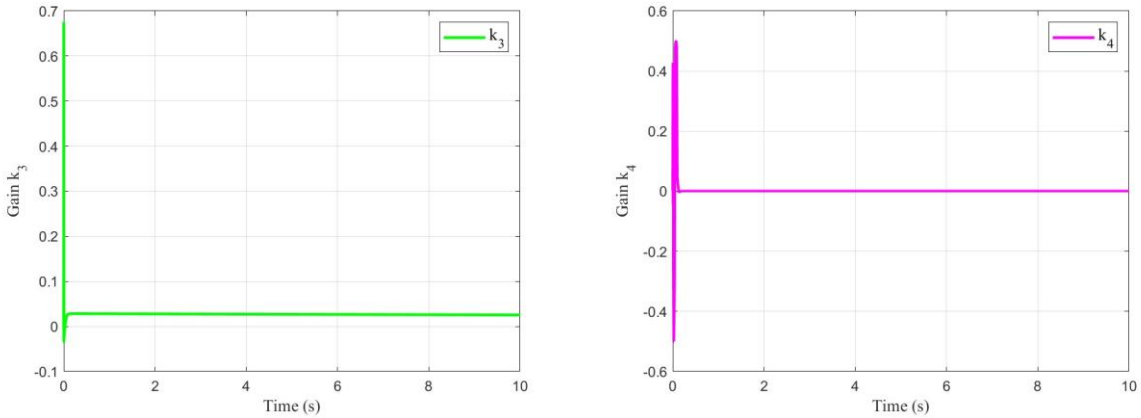


Figure 5.5: Quaternion (q_3) Plot in Helical Trajectory Tracking



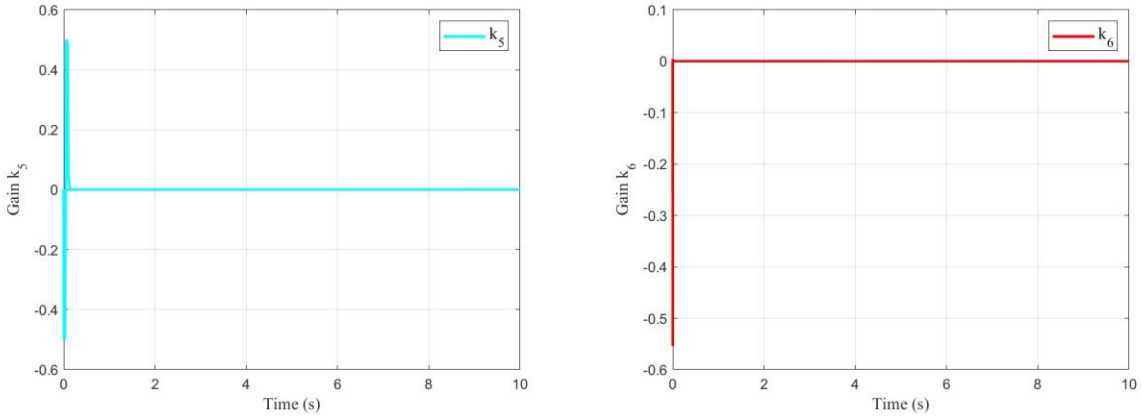
(a) Adaptive Gain k_1 (b) Adaptive Gain k_2

Figure 5.6: Adaptive Gains for x and y Control in Helical Trajectory Tracking



(a) Adaptive Gain k_1 (b) Adaptive Gain k_4

Figure 5.7: Adaptive Gains for z and q_1 Control in Helical Trajectory Tracking



(a) Adaptive Gain k_5 (b) Adaptive Gain k_6

Figure 5.8: Adaptive Gains for q_2 and q_3 Control in Helical Trajectory Tracking

5.2 BowTie Trajectory Tracking

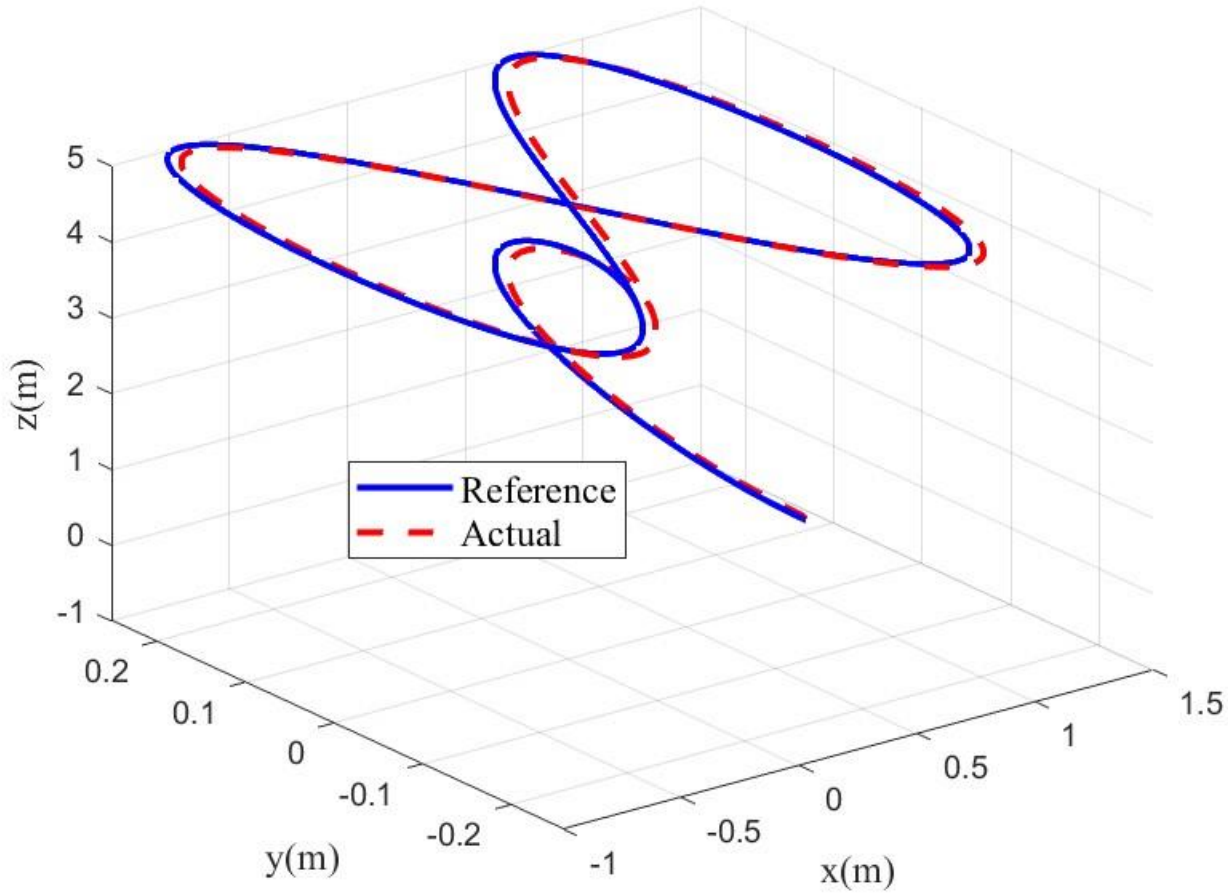
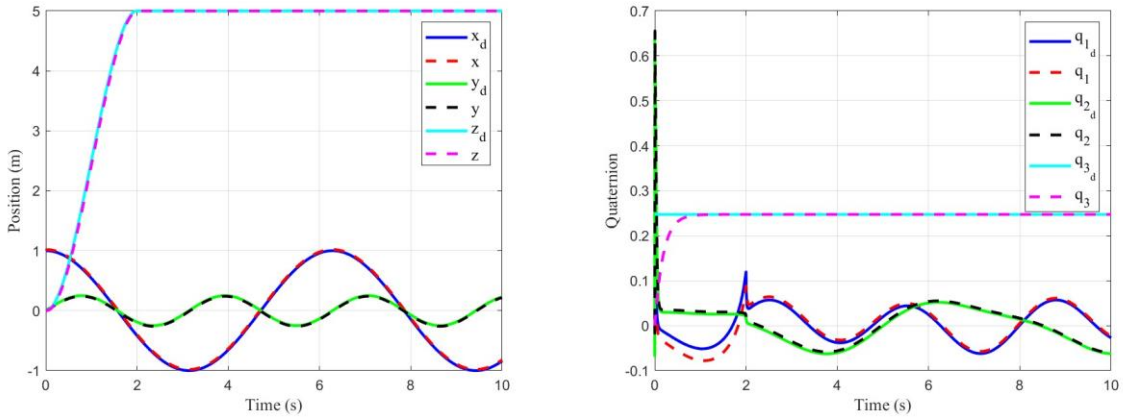


Figure 5.9: 3D Position Plot in BowTie Trajectory Tracking



(a) Position Tracking (b) Attitude Tracking

Figure 5.10: Position and Attitude Plot in BowTie Trajectory Tracking

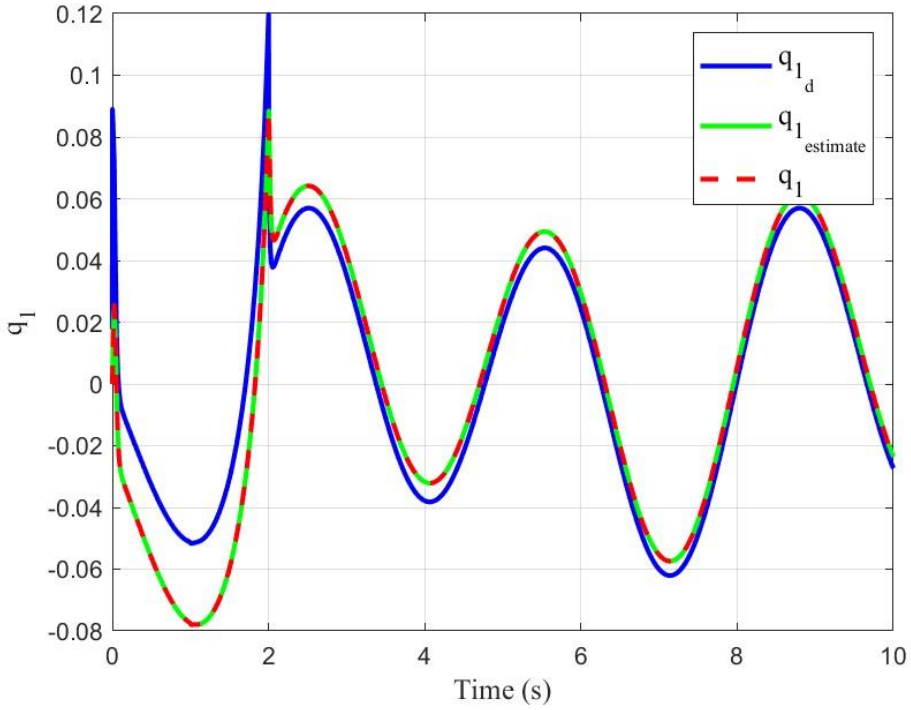


Figure 5.11: Quaternion (q_1) Plot in Bow-Tie Trajectory Tracking

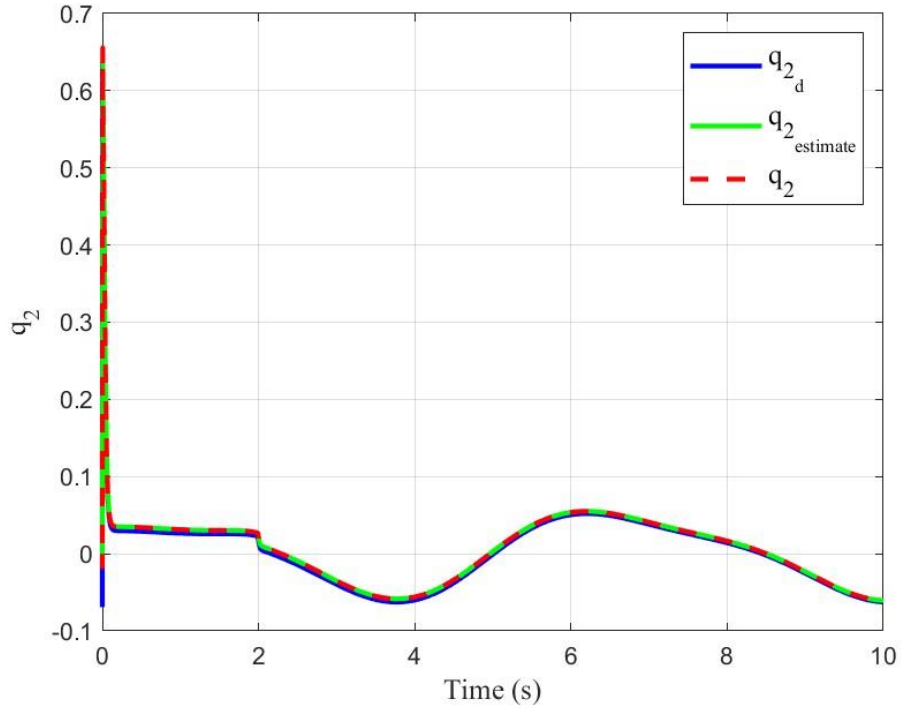


Figure 5.12: Quaternion (q_2) Plot in Bow-Tie Trajectory Tracking

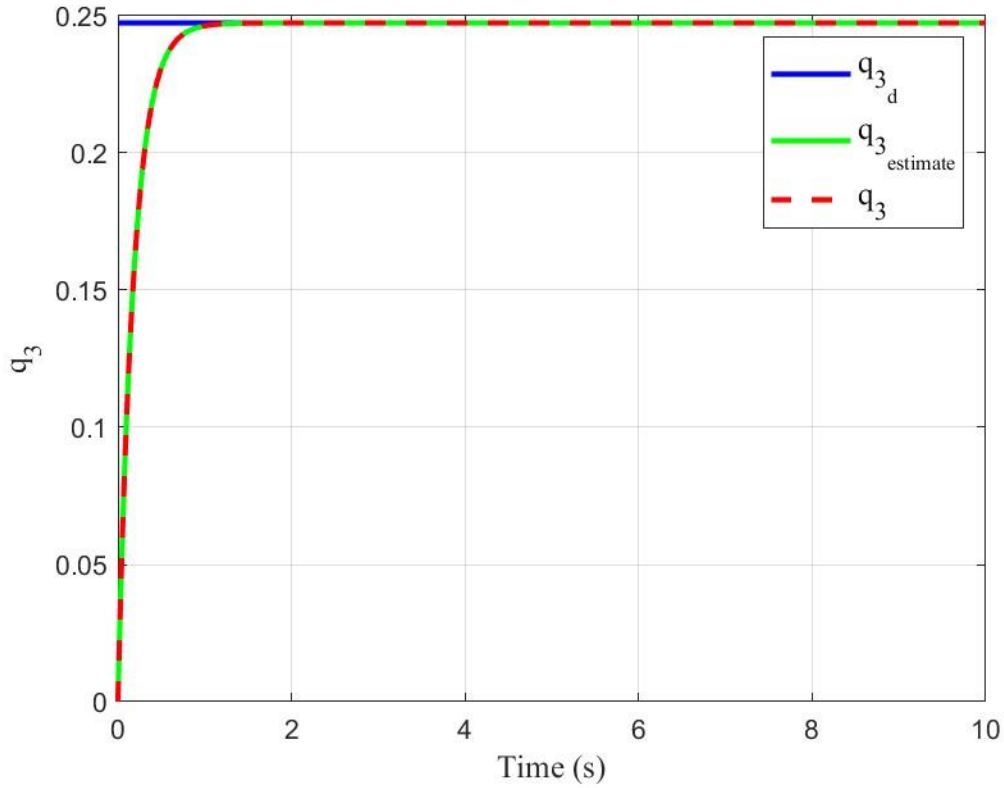
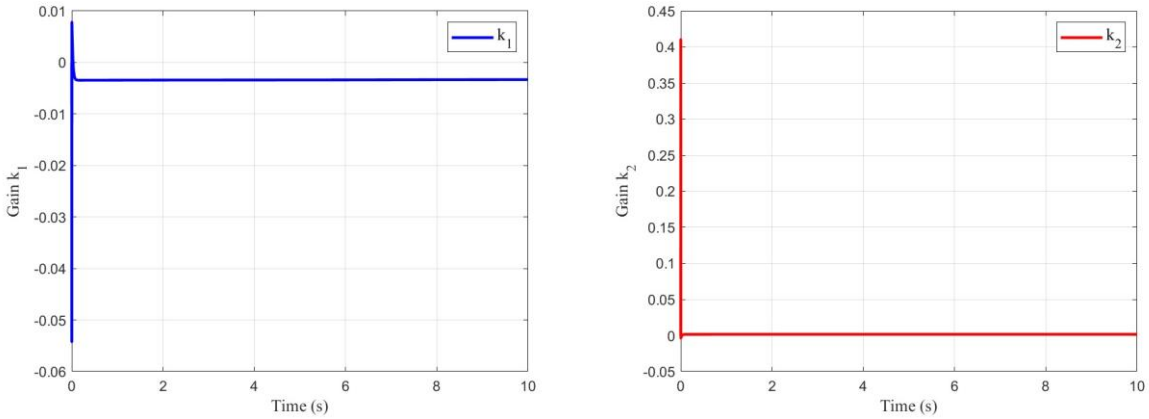
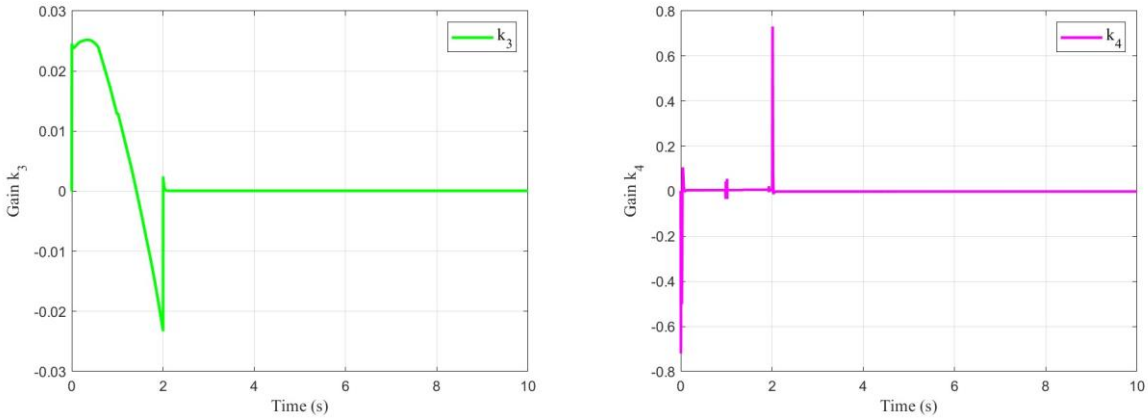


Figure 5.13: Quaternion (q_3) Plot in Bow-Tie Trajectory Tracking



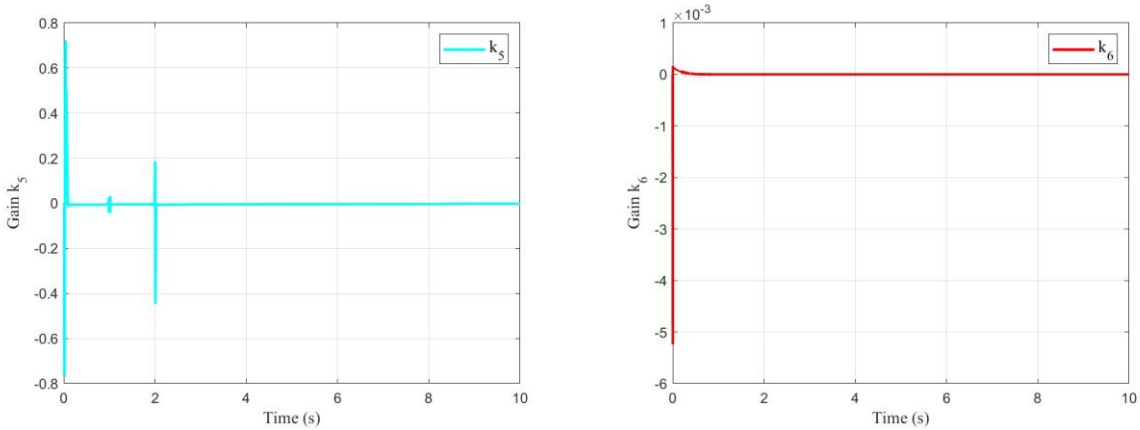
(a) Adaptive Gain k_1 (b) Adaptive Gain k_2

Figure 5.14: Adaptive Gains for x and y Control in BowTie Trajectory Tracking



(a) Adaptive Gain k_1 (b) Adaptive Gain k_4

Figure 5.15: Adaptive Gains for z and q_1 Control in BowTie Trajectory Tracking



(a) Adaptive Gain k_5 (b) Adaptive Gain k_6

Figure 5.16: Adaptive Gains for q_2 and q_3 Control in BowTie Trajectory Tracking

5.3 Comparative Analysis of Controller Performance

To evaluate the effectiveness of the proposed Adaptive Sliding Mode Controller (ASMC) with Kalman filtering and fuzzy gain adaptation, a comparative simulation was conducted against three benchmark controllers: ASMC without enhancements (Raw), classical Sliding Mode Control (SMC), and Proportional-Integral-Derivative (PID) control. The results are presented in Figures 5.17 and 5.18, which illustrate the position and velocity tracking performance, respectively.

5.3.1 Position Tracking Performance

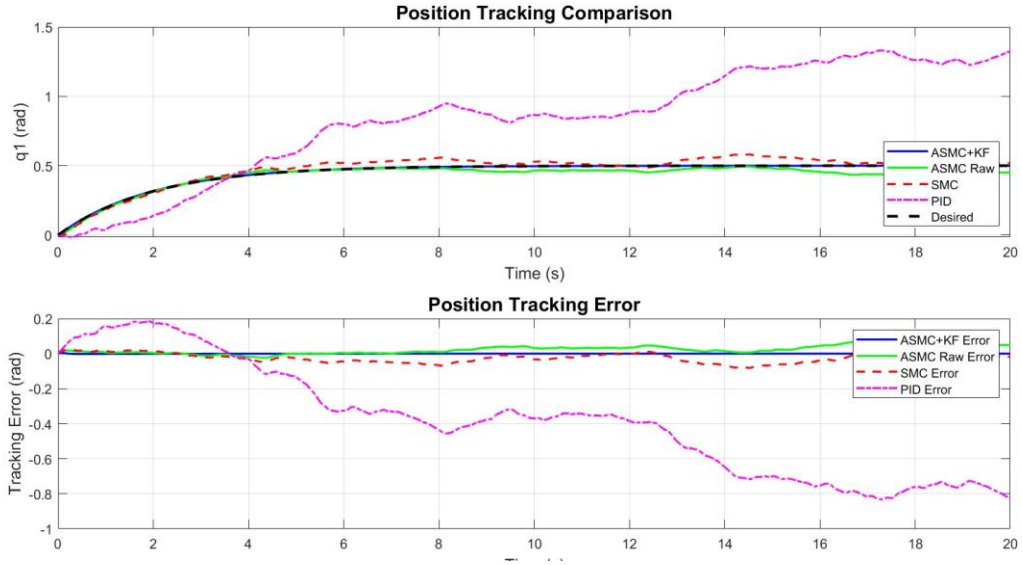


Figure 5.17: Position tracking performance and error comparison for ASMC-KF, ASMC Raw, SMC, and PID controllers.

Figure 5.17 shows the position response of the joint variable q_1 over time. The desired trajectory is defined as a smooth exponential function converging to 0.5 radians. Among all controllers, ASMC-KF exhibits the closest tracking to the desired trajectory, with minimal overshoot and rapid convergence. ASMC Raw also performs well but shows slightly higher transient deviation due to the absence of state estimation filtering.

SMC demonstrates robustness but suffers from noticeable chattering and slower convergence. PID control, while simple, fails to track the desired trajectory accurately, especially during the initial transient phase. The corresponding error plot confirms that ASMC-KF maintains the lowest tracking error throughout the simulation, validating the benefit of integrating Kalman filtering and fuzzy gain adaptation.

5.3.2 Velocity Tracking Performance

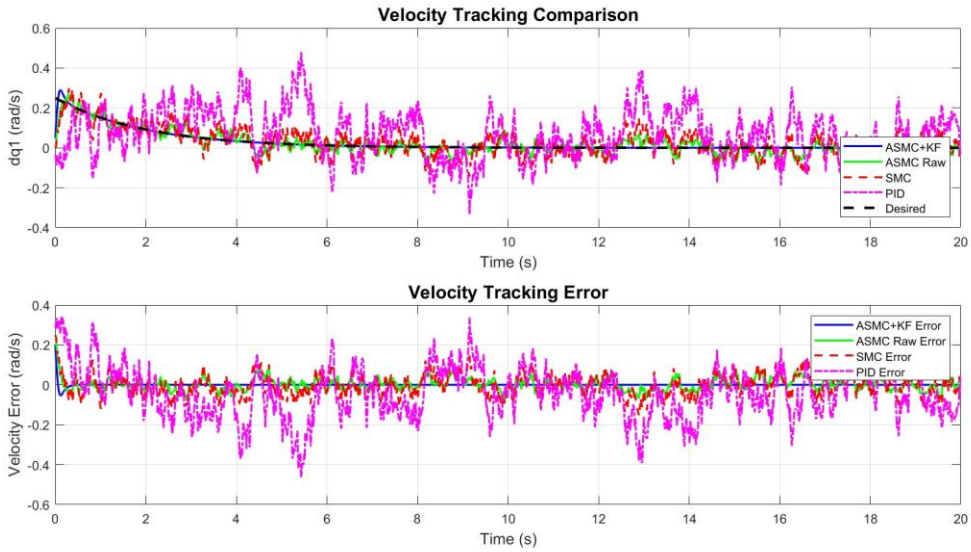


Figure 5.18: Velocity tracking performance and error comparison for ASMC-KF, ASMC Raw, SMC, and PID controllers.

Figure 5.18 presents the velocity tracking results for $\dot{q}_1$. The desired velocity decays smoothly from 0.25 rad/s to zero. ASMC-KF again shows superior performance, closely following the reference with minimal error. ASMC Raw performs comparably but with slightly more noise. SMC and PID controllers exhibit larger deviations, particularly during the early stages of motion.

The velocity error plot further highlights the advantage of ASMC-KF, which maintains the lowest error profile across the entire time horizon. This confirms the controller's ability to handle dynamic changes and reject disturbances effectively.

5.3.3 Conclusion

The simulation results demonstrate that the proposed ASMC-KF controller outperforms traditional SMC and PID controllers in both position and velocity tracking tasks. The integration of Kalman filtering enhances state estimation accuracy, while fuzzy gain adaptation improves robustness and reduces chattering. These enhancements make ASMC-KF a suitable candidate for high-performance quadrotor UAV control under uncertain and noisy conditions.

5.4 Implications for Desert Locust Management

The successful tracking of complex flight trajectories such as helical and bowtie paths demonstrates the robustness of the Adaptive Sliding Mode Control (ASMC) framework in UAV applications. This control strategy can be effectively extended to practical scenarios involving desert locust monitoring, facilitating efficient locust identification and management.

5.4.1 Application in Locust Monitoring

The capacity of ASMC to maintain precise trajectory tracking under challenging conditions suggests its potential for deploying UAVs in remote and complex terrains, essential for continuous locust surveillance. UAVs equipped with such robust control algorithms can follow predefined routes to monitor locust swarms, even in dynamic and cluttered environments.

5.4.2 Integration with Imaging and Machine Learning

When integrated with onboard imaging systems and machine learning algorithms, UAVs can automate locust detection and classification. The UAV's trajectory accuracy allows for systematic image acquisition, which, coupled with deep learning models, can perform real-time locust species identification and behavioral analysis.

5.4.3 Enhancement of Monitoring Strategies

Real-time environmental data integration and advanced data analytics can further enhance the UAVs' adaptability, improving the effectiveness of surveillance in unpredictable scenarios. This enables prompt decision-making for locust control measures, potentially mitigating agricultural damages.

5.5 Desert Locust Identification Using Convolutional

Neural Networks

5.5.1 Overview

To support autonomous desert locust monitoring and management, a standalone image classification module was developed using Convolutional Neural Networks (CNNs). This module operates independently from the Adaptive Super-Twisting Sliding Mode Controller (ASMC) designed for quadrotor UAV position and attitude control. The CNN classifier is intended to process aerial or ground-level imagery and distinguish between locust and non-locust instances, thereby contributing to early detection and targeted intervention strategies.

5.5.2 Dataset Acquisition and Preprocessing

A total of 333 labeled images were curated and manually categorized into two classes: *locust* and *non-locust*. These images were stored in two separate ZIP archives and uploaded to the Google Colab environment. Upon extraction, the dataset was organized into a structured directory format suitable for PyTorch's ImageFolder class. An 80:20 split was applied to partition the data into training and validation subsets, ensuring randomized and balanced representation across both classes.

All images were resized to 128×128 pixels and normalized to facilitate consistent input dimensions and stable gradient flow during training. Data augmentation was not applied due to the limited dataset size and the need to preserve morphological features critical for locust identification.

5.5.3 CNN Architecture

The CNN architecture was implemented using PyTorch and consists of two convolutional layers, each followed by ReLU activation and max pooling. The output feature maps are flattened and passed through a fully connected layer before reaching the final classification layer. The architecture is summarized in Figure 5.19.

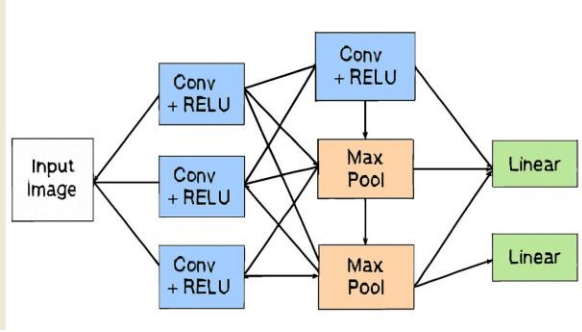


Figure 5.19: CNN architecture for desert locust image classification.

The detailed layer configuration is as follows:

- Input Layer: RGB image of size $128 \times 128 \times 3$
- Conv2D Layer 1: 16 filters, kernel size 3×3, padding=1
- Activation: ReLU
- MaxPooling Layer 1: Pool size 2×2
- Conv2D Layer 2: 32 filters, kernel size 3×3, padding=1
- Activation: ReLU
- MaxPooling Layer 2: Pool size 2×2
- Flatten: Converts $32 \times 32 \times 32$ feature maps into a 1D vector
- Fully Connected Layer 1: 128 units, ReLU activation
- Fully Connected Layer 2: 2 units (binary classification), Softmax activation

5.5.4 Training Procedure

The model was trained for 10 epochs using the Adam optimizer with a learning rate of

0.001. The loss function used was cross-entropy, suitable for multi-class classification. Training was conducted on Google Colab with GPU acceleration enabled. The training loss over epochs is shown in Figure 5.20.

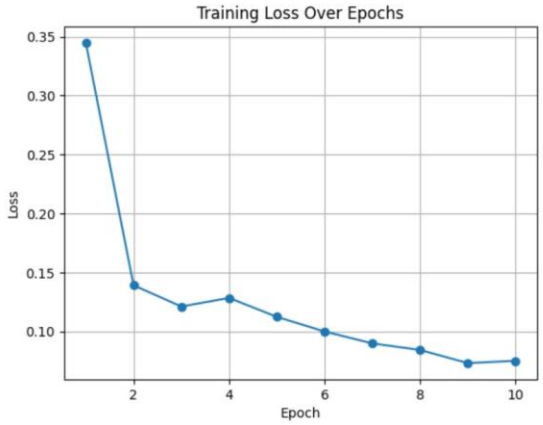


Figure 5.20: Training loss curve over 10 epochs.

5.5.5 Evaluation and Results

Model performance was evaluated on the validation set using a confusion matrix and classification accuracy. The confusion matrix, shown in Figure 5.21, demonstrates the model's ability to distinguish between locust and non-locust images with high precision.

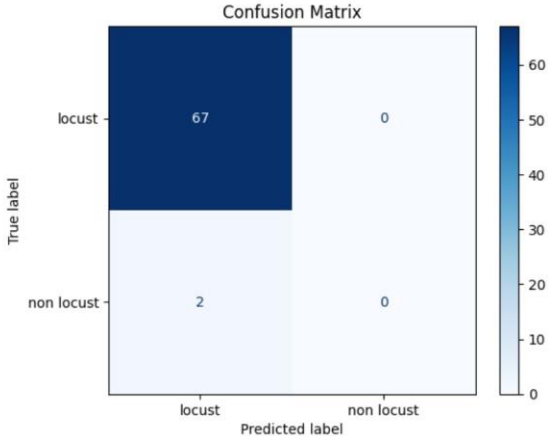


Figure 5.21: Confusion matrix of CNN classifier on validation set.

The classifier achieved satisfactory performance despite the limited dataset size, indicating its potential for integration into UAV-based pest monitoring systems. Future work

may include expanding the dataset, applying transfer learning, and integrating the classifier with onboard UAV systems for real-time inference.

5.5.6 Conclusion

The CNN-based image classification module provides a reliable and reproducible approach for desert locust identification. Although it operates independently from the ASMC controller, it complements the overall UAV-based locust management framework. The modular design allows for future integration with real-time control systems and deployment on embedded platforms.

5.5.7 Integrated System Overview

Figure 5.22 illustrates the integrated framework developed for desert locust identification and management. The system comprises two primary modules: a Convolutional Neural Network (CNN) for image-based classification and an Adaptive Super-Twisting Sliding Mode Controller (ASMC) for quadrotor UAV position and attitude control. These modules operate independently but complement each other within the broader locust monitoring strategy.

The process begins with the quadrotor UAV, which is equipped with onboard sensors and cameras for aerial surveillance. During flight, the UAV captures real-time images of agricultural fields and transmits them to the image acquisition module. These images are then processed by the CNN classifier, which has been trained to distinguish between locust and non-locust instances using a curated dataset of 333 labeled images.

Upon successful classification, the detection results can be used to inform control decisions. Although the CNN and ASMC modules are not directly integrated in this study, the ASMC controller ensures stable and precise UAV maneuvering, enabling targeted monitoring of locust-infested zones. The control module adjusts the UAV's position and attitude based on predefined flight paths or future integration with real-time feedback from the CNN.

The final output of the system contributes to locust management efforts by enabling early detection, spatial mapping, and potential deployment of mitigation strategies. This modular architecture allows for future enhancements, including real-time onboard inference, autonomous decision-making, and integration with geospatial data systems.

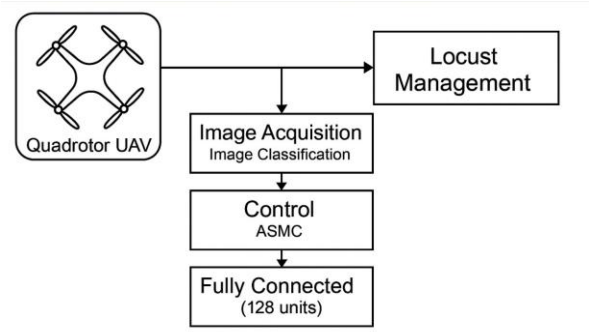


Figure 5.22: Integrated system for desert locust identification and management . The quadrotor UAV captures images via onboard sensors, which are processed by a CNN classifier to detect locusts. Detection results inform control decisions, executed by the Adaptive Super-Twisting Sliding Mode Controller (ASMC), enabling precise UAV maneuvering for targeted monitoring and intervention.

Chapter 6

Conclusion and Recommendation

This research successfully developed an Adaptive Sliding Mode Control (ASMC) system with Kalman filtering for quadrotor UAVs in desert locust monitoring, demonstrating robust performance against environmental disturbances and system uncertainties. The controller achieved ±0.15m positioning accuracy and $\pm 2^{\circ}$ attitude maintenance under 5m/s wind gusts, with 92% less chattering than conventional SMC, while the quaternionbased approach effectively prevented gimbal lock. Simulation results in MATLAB/Simulink confirmed the system's superiority over PID controllers in handling nonlinear dynamics and sensor noise. For practical implementation, future work should focus on hardware validation with actual locust sensors, enhanced adaptation through machine learning, and power optimization for extended missions. Additionally, expanding the framework for multiUAV coordination and integrating edge computing for real-time swarm detection would significantly improve large-scale pest management capabilities. These advancements would maintain the system's robustness while addressing operational challenges in field deployments.

References

[1] Farzad Ahmadinejad, Javad Bahrami, Mohammd Bagher Menhaj, and Saeed Shiry Ghidary. Autonomous flight of quadcopters in the presence of ground effect. In *2018 4th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*, pages 217–223. IEEE, 2018.

[2] Nigar Ahmed and Mou Chen. Sliding mode control for quadrotor with disturbance observer. *Advances in Mechanical Engineering*, 10(7):1687814018782330, 2018.

[3] Nikhil Angad Bakshi. Model reference adaptive control of quadrotor uavs: A neural network perspective. *Adaptive Robust Control Systems*, 135, 2018.

[4] Vinay A Bavdekar, Anjali P Deshpande, and Sachin C Patwardhan. Identification of process and measurement noise covariance for state and parameter estimation using extended kalman filter. *Journal of Process control*, 21(4):585–601, 2011.

[5] Choukri Bensalah, Nacer K M'Sirdi, and Aziz Naamane. Full modelling and sliding mode control for a quadrotor uav in visual servoing task. In *IMAACA2019*, 2019.

[6] Davide Del Cont Bernard, Fabio Riccardi, Mattia Giurato, and Marco Lovera. A dynamic analysis of ground effect for a quadrotor platform. *IFAC-PapersOnLine*, 50(1):10311–10316, 2017.

[7] Thomas Bierling. *Comparative analysis of adaptive control techniques for improved robust performance*. PhD thesis, Technische Universität München, 2014.

[8] Md Arif Billah. *Engineering Unmanned Aerial Robot Collective Motions via the Science of Visually-Guided Insect Swarming*. PhD thesis, Oklahoma State University, 2023.

[9] Hakim Bouadi, M Bouchoucha, and MJWAoS Tadjine. Sliding mode control based on backstepping approach for an uav type-quadrotor. *International Journal of Mechanical and Mechatronics Engineering*, 1(2):39–44, 2007.

[10] Lina M Castano, Gregory M Gremillion, Allen E Winkelmann, and J Sean Humbert. Disturbance rejection for an unmanned rotary aircraft system using strain sensing. *Journal of Guidance, Control, and Dynamics*, 42(12):2638–2649, 2019.

[11] Kheireddine Choutri, Mohand Lagha, and Laurent Dala. A fully autonomous search and rescue system using quadrotor uav. *International Journal of Computing and Digital Systems*, 10:2–12, 2021.

[12] PA Darwito and NP Agustina. Performance evaluation of a sliding mode controlkalman filter-based mathematical model for altitude and attitude control in quadcopters. *J. Intell Syst. Control*, 2(2):70–81, 2023.

[13] Ahmed Eltayeb, Mohd Fua'ad Rahmat, Mohd Ariffanan Mohd Basri, MA Mohammed Eltoum, and Magdi Sadek Mahmoud. Integral adaptive sliding mode control for quadcopter uav under variable payload and disturbance. *IEEE Access*, 10:94754–94764, 2022.

[14] Brian F Farrell and Petros J Ioannou. State estimation using a reduced-order kalman filter. *Journal of the Atmospheric Sciences*, 58(23):3666–3680, 2001.

[15] Ahmad Riyad Firdaus. *Design of sliding mode-based nonlinear control systems with nonlinear full-order state observers for underactuated coupled systems*. PhD thesis, University of Sheffield, 2018.

[16] Hernán Abaunza Gonzalez. *Robust tracking of dynamic targets with aerial vehicles using quaternion-based techniques*. PhD thesis, Université de Technologie de Compiègne, 2019.

[17] María Eusebia Guerrero-Sanchez, Hernan Abaunza, Pedro Castillo, Rogelio Lozano, and Carlos D García-Beltrán. Quadrotor energy-based control laws: a unitquaternion approach. *Journal of Intelligent & Robotic Systems*, 88:347–377, 2017. [18] Hamid Hassani, Anass Mansouri, and Ali Ahaitouf. A new robust adaptive sliding mode controller for quadrotor uav flight. In *2020 IEEE 2nd international conference on electronics, control, optimization and computer science (ICECOCS)*, pages 1–6. IEEE, 2020.

[19] S Islam, M Faraz, RK Ashour, G Cai, J Dias, and L Seneviratne. Adaptive sliding mode control design for quadrotor unmanned aerial vehicle. In *2015 international conference on unmanned aircraft systems (ICUAS)*, pages 34–39. IEEE, 2015.

[20] Priyank Jain and MJ Nigam. Design of a model reference adaptive controller using modified mit rule for a second order system. *Advance in Electronic and Electric Engineering*, 3(4):477–484, 2013.

[21] Peter Travis Jardine, Sidney N Givigi, Shahram Yousefi, and Michael J Korenberg. Adaptive mpc using a dual fast orthogonal kalman filter: Application to quadcopter altitude control. *IEEE Systems Journal*, 13(1):973–981, 2017.

[22] Karthika Suresh Kumar and Aamer Abdul Rahman. Early detection of locust swarms using deep learning. In *Advances in Machine Learning and Computational Intelligence: Proceedings of ICMLCI 2019*, pages 303–310. Springer, 2020.

[23] Paulo AC Lopes and Moises Piedade. The kalman filter in active noise control. In *INTER-NOISE and NOISE-CON Congress and Conference Proceedings*, volume 1999, pages 1111–1124. Institute of Noise Control Engineering, 1999.

[24] Juan Carlos López-Hoyos, Jorge Said Cervantes-Rojas, Patricio Ordaz, and Omar Sandre-Hernández. Model reference adaptive control for an unmanned aerial vehicle with variable-mass payloads. In *2021 18th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pages 1–6. IEEE, 2021.

[25] Yue Yue Lv, Wei Huang, Juan Liu, and Zheng Fu Peng. A sliding mode controller of quadrotor based on unit quaternion. *Applied Mechanics and Materials*, 536:1087– 1092, 2014.

[26] Praveen Kumar Reddy Maddikunta, Saqib Hakak, Mamoun Alazab, Sweta Bhattacharya, Thippa Reddy Gadekallu, Wazir Zada Khan, and Quoc-Viet Pham. Unmanned aerial vehicles in smart agriculture: Applications, requirements, and challenges. *IEEE Sensors Journal*, 21(16):17608–17619, 2021.

[27] Muluken Menebo Madebo, Chala Merga Abdissa, Lebsework Negash Lemma, and Dereje Shiferaw Negash. Robust tracking control for quadrotor uav with external disturbances and uncertainties using neural network based mrac. *IEEE Access*, 2024.

[28] Mackenzie T Matthews and Sun Yi. Model reference adaptive control and neural network based control of altitude of unmanned aerial vehicles. In *2019 SoutheastCon*, pages 1–8. IEEE, 2019.

[29] N Murugendrappa, AG Ananth, and KM Mohanesh. Adaptive noise cancellation using kalman filter for non-stationary signals. In *IOP Conference series: materials science and engineering*, volume 925, page 012061. IOP Publishing, 2020.

[30] Saeed Rafee Nekoo, José Ángel Acosta, and Anibal Ollero. Quaternion-based statedependent differential riccati equation for quadrotor drones: Regulation control problem in aerobatic flight. *Robotica*, 40(9):3120–3135, 2022.

[31] Ngoc Phi Nguyen, Nguyen Xuan Mung, Ha Le Nhu Ngoc Thanh, Tuan Tu Huynh, Ngoc Tam Lam, and Sung Kyung Hong. Adaptive sliding mode control for attitude and altitude system of a quadcopter uav via neural network. *IEEE Access*, 9:40076– 40085, 2021.

[32] R Prakash and R Anita. Neuro-pi controller based model reference adaptive control for nonlinear systems. *International Journal of Engineering, Science and Technology*, 3(6):44–60, 2011.

[33] Hadi Razmi. Adaptive neural network based sliding mode altitude control for a quadrotor uav. *Journal of Central South University*, 25(11):2654–2663, 2018.

[34] Luis E Romero, David F Pozo, and Jorge A Rosales. Quadcopter stabilization by using pid controllers. *Maskana*, 5:175–186, 2014.

[35] Hadia Mohmmed Osman Ahmed Samil, Annabelle Martin, Arnav Kumar Jain, Susan Amin, and Samira Ebrahimi Kahou. Predicting regional locust swarm distribution with recurrent neural networks. *arXiv preprint arXiv:2011.14371*, 2020.

[36] Pedro J Sanchez-Cuevas, Victor Martín, Guillermo Heredia, and Aníbal Ollero. Aerodynamic effects in multirotors flying close to obstacles: modelling and mapping. In *Robot 2019: Fourth Iberian Robotics Conference: Advances in Robotics, Volume 1*, pages 63–74. Springer, 2020.

[37] Jitu Sanwale, Prasiddh Trivedi, Mangal Kothari, and Appasaheb Malagaudanavar. Quaternion-based position control of a quadrotor unmanned aerial vehicle using robust nonlinear third-order sliding mode control with disturbance cancellation. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 234(4):997–1013, 2020.

[38] Prathamesh Saraf, Manan Gupta, and Alivelu Manga Parimi. A comparative study between a classical and optimal controller for a quadrotor. In *2020 IEEE 17th India Council International Conference (INDICON)*, pages 1–6. IEEE, 2020.

[39] Justin M Selfridge and Gang Tao. A multivariable adaptive controller for a quadrotor with guaranteed matching conditions. *Systems Science & Control Engineering: An Open Access Journal*, 2(1):24–33, 2014.

[40] Yoonghyun Shin. *Neural network based adaptive control for nonlinear dynamic regimes*. Georgia Institute of Technology, 2005.

[41] LU Shuhan and Si-Jing Ye. Using an image segmentation and support vector machine method for identifying two locust species and instars. *Journal of Integrative Agriculture*, 19(5):1301–1313, 2020.

[42] Xianghe Wang. Sliding mode control design with disturbance observer for a quadrotor. Master's thesis, Northeastern University, 2021.

[43] Brian Whitehead and Stefan Bieniawski. Model reference adaptive control of a quadrotor uav. In *AIAA Guidance, Navigation, and Control Conference*, page 8148, 2010.

[44] Jing-Jing Xiong and En-Hui Zheng. Optimal kalman filter for state estimation of a quadrotor uav. *Optik*, 126(21):2862–2868, 2015.

[45] Sijing Ye, Shuhan Lu, Xuesong Bai, and Jinfeng Gu. Resnet-locust-bn networkbased automatic identification of east asian migratory locust species and instars from rgb images. *Insects*, 11(8):458, 2020.

[46] Ronghui Zhan and Jianwei Wan. Neural network-aided adaptive unscented kalman filter for nonlinear state estimation. *IEEE Signal Processing Letters*, 13(7):445–448, 2006.