Adaptive Nonlinear Gain Approximation of a Robust Nonlinear Controller to Handle Uncertain Disturbances for UAV Systems Using Neural Networks

MS Thesis

by

Waqas Ahmad

CIIT/FA23-REE-003/LHR

COMSATS University Islamabad
Pakistan

Fall 2025

Adaptive Nonlinear Gain Approximation of a Robust Nonlinear Controller to Handle Uncertain Disturbances for UAV Systems using Neural Networks

A thesis submitted to
COMSATS University Islamabad, Lahore Campus

In partial fulfillment
of the requirement for the degree of

Master of Science
in
Electrical Engineering

by
Waqas Ahmad
CIIT/FA23-REE-003/LHR

Department of Electrical Engineering
Faculty of Engineering

COMSATS University Islamabad,
Lahore Campus, Pakistan

Fall 2025

Adaptive Nonlinear Gain Approximation of a Robust Nonlinear Controller to Handle Uncertain Disturbances for UAV Systems using Neural Networks

This thesis is submitted to the Department of Electrical Engineering in partial fulfillment of the requirements for the award of the degree of Master of Science in Electrical Engineering.

Name	Registration Number
Waqas Ahmad	CIIT/FA23-REE-003/LHR

Supervisor:

Dr. Mirza Tariq Hamayun

Associate Professor,
Head Department of Electrical Engineering
COMSATS University Islamabad, Lahore Campus

January, 2026

Certificate of Approval

This thesis titled

Adaptive Nonlinear Gain Approximation of a Robust Nonlinear Controller to Handle Uncertain Disturbances for UAV Systems using Neural Networks

By

Waqas Ahmad

CIIT/FA23-REE-003/LHR

Has been approved

For COMSATS University Islamabad, Lahore Campus.

External Examiner:_____

Dr. Yasir Saleem

University of Engineering and Technology (UET), Lahore

Supervisor:_____

Dr. Mirza Tariq Hamayun

Department of Electrical Engineering, CUI, Lahore Campus

Head of Department:_____

Dr. Mirza Tariq Hamayun

Department of Electrical Engineering, CUI, Lahore Campus

Author's Declaration

I Waqas Ahmad, Registration No. CIIT/FA23-REE-003/LHR, hereby declare that I have produced the work presented in this thesis during the scheduled period of study. I also declare that I have not taken any material from any source except as referred to, and therefore, the amount of plagiarism is within an acceptable range. If a violation of HEC rules on research has occurred in this thesis, I shall be liable to punishable action under the plagiarism rules of HEC.

Date:_____ _____

Waqas Ahmad

CIIT/FA23-REE-003/LHR

Certificate

It is certified that Mr. Waqas Ahmad, CIIT/FA23-REE-003/LHR, has carried out all the work related to this thesis under my supervision at the Department of Electrical Engineering, COMSATS University Islamabad, Lahore Campus, and the work fulfills the requirements for the award of MS degree.

Date:_____

Supervisor

_____

Dr. Mirza Tariq Hamayun
Associate Professor
COMSATS University Islamabad
Lahore Campus

Dedication

To my beloved parents, supportive and caring family, and lovely siblings.

Acknowledgments

All praises are due to Almighty ALLAH, the Most Gracious and the Most Merciful. First and foremost, I am deeply grateful to Almighty ALLAH for granting me the opportunity, determination, and strength to undertake and complete my research work. His continuous grace and mercy have been with me throughout my life, and especially during the course of my research. May the countless blessings and peace of Allah be upon the Holy Prophet Muhammad (SAW) and his noble family, who guided humanity toward the righteous path and a virtuous way of life.

It is a great pleasure to express my sincere appreciation to my supervisor, Dr. Mirza Tariq Hamayun, for his invaluable mentorship and outstanding guidance throughout this research. His continuous support enabled me to successfully complete this work. His exceptional knowledge and expertise in control systems, along with his insightful guidance and encouraging attitude, have been a constant source of inspiration and will be remembered. I am thankful to my supervisory committee members, Dr. Muhammad Farooq-i-Azam, Dr. Ayesha Ali, and Dr. Aamer Bilal Asghar, for their critical reviews and insightful comments despite their demanding schedules. Their valuable recommendations and thoughtful queries significantly contributed to improving this work. I am also thankful to Dr. Zainab Akhtar (UET, Lahore) for her guidance, and recommendations throughout this research.

My friends, Dr. Naveed Mazhar, Saeed Mazhar, and Maaz Tahir Malik, deserve special thanks. Without their help, support, and encouragement, this research wouldn't be possible.

I am sincerely thankful to my organization for the opportunity to enhance my education. I would also like to extend my sincere thanks to Dr. Zahid Hameed Qazi, Syed Muhammad Akram Hussain Shah, Mr. Muhammad Arif, and Mr. Asif Mahmood Bhatti for their consideration and support.

Waqas Ahmad

CIIT/FA23-REE-003/LHR

Abstract

The quadrotor unmanned aerial vehicle (UAV) has been a part of scientific studies owing to its applications in military, agriculture, infrastructure and construction, and environmental monitoring and conservation. However, its control design is challenging due to its nonlinear underactuated dynamics, external disturbances, parametric uncertainties, and the aerodynamic variations encountered during flight operations. Sliding mode control (SMC) is the most widely used control technique for handling nonlinear dynamics in the presence of external disturbances and modeling uncertainties. However, conventional SMC requires a priori knowledge of the upper bound on the overall disturbance at all times. Usually, these upper bounds on disturbances are derived from past data, environmental conditions, and empirical modeling; therefore, a conservative bound is used in SMC design, leading to high control effort and draining more energy from the system. To address this issue, adaptive sliding mode control (ASMC) strategies based on a neural network (NN) are presented in this dissertation.

Neural networks are recognized as the leading approach to intelligent computation and are highly effective in handling nonlinearity, fault tolerance, adaptation, and continuous online learning. In this research, NN-based ASMC strategies are presented for a quadrotor UAV to improve flight performance by online estimation of adaptive modulation gain in response to variations in disturbance magnitude. Different neural network schemes are employed, including feed-forward neural networks (FNNs) using backpropagation (BP) and Levenberg-Marquardt (LM) algorithms for optimization, and a radial basis function neural network (RBFNN) with a Gaussian activation function, to adaptively estimate the modulation gains. Furthermore, the stability analysis of the proposed controllers is proven using Lyapunov theory.

Numerical simulations in MATLAB are conducted to validate the tracking performance and disturbance-rejection capabilities of the presented controllers for a quadrotor UAV. Finally, a comprehensive comparative analysis between the NN-based schemes outlined in this thesis and classical SMC is presented, highlighting the effectiveness of the proposed control schemes in enhancing flight performance in the presence of perturbations while minimizing energy consumption.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ASMC Adaptive Sliding Mode Control

ABPSMC Adaptive Back Propagation Sliding Mode Control

ALMSMC Adaptive Levenberg-Marquardt Sliding Mode Control

ARBFSMC Adaptive Radial Basis Function Sliding Model Control

DO Disturbance Observer

DOF Degree of Freedom

EBP Error Backpropagation

EKF Extended Kalman Filter

ENN Evidential Neural Network

FCRNN Fully Connected Recurrent Neural Network

FNN Feedforward Neural Network

ISMC Integral Sliding Mode Control

IAE Integral Absolute Error

ITAE Integral Time Absolute Error

LMA Levenberg-Marquardt Algorithm

MLP Minimum Learning Parameter

MIMO Multiple Input Multiple Output

NNs Neural Networks

RBFNN Radial Basis Function Neural Network

RSME Root Mean Squared Error

SMC Sliding Mode Control

TSMC Twisting Sliding Mode Control

UAV Unmanned Aerial Vehicle

USV Unmanned Surface Vehicle

VTOL Vertical Take-Off and Landing

Chapter 1
Introduction

1.1 Background and Motivation

Research on UAVs has gained significant attention in recent years due to the advancements in nonlinear control techniques and their integration with machine learning algorithms. The need for a stable and reliable UAV is gaining importance due to its diverse applications, particularly in military and rescue operations, agriculture, surveillance, logistics, and industrial inspections. However, controlling quadrotor UAVs faces several challenges due to their underactuated structure, nonlinear dynamics, unmodeled uncertainties, and vulnerability to external disturbances, such as wind gusts. These factors significantly affect the reliability of quadrotor UAV flight operations; therefore, an adaptive and robust controller is required to overcome these challenges and ensure smooth flight.

Numerous control strategies for quadrotor UAVs have evolved significantly, spanning from classical linear controllers to modern robust and adaptive controllers. In nonlinear control schemes, SMC has emerged as a prominent controller due to its robustness to external disturbances. However, SMC faces several challenges, including the requirement of prior knowledge of disturbance bounds, high-frequency chattering, and sensitivity during the reaching phase. Therefore, to address these shortcomings, a viable solution is to utilize adaptive gain tuning and integrate SMC with intelligent control algorithms, such as neural networks.

Neural networks can learn and adapt the dynamics of nonlinear systems in real-world scenarios. By estimating uncertain disturbances and dynamically tuning control parameters, NNs enhance robustness and overcome the limitations of classical SMC. Motivated by these developments, this research proposes the design of an NN-based adaptive and intelligent control framework that can estimate and adjust the modulation gain online to ensure stability, accurate tracking, and robustness against

uncertain disturbances, while minimizing control effort and energy consumption.

1.2 Problem Statement

Conventional SMC offers robustness but relies on prior knowledge of disturbance bounds. These bounds are not known at all times or may be time-varying, which necessitates overly conservative gains. This results in high control effort, increased energy consumption, and high chattering in the actuator, thereby accelerating actuator wear and degradation. Therefore, an adaptive and intelligent control framework is required to estimate and adjust the modulation gain online, to ensure stability, accurate tracking, and robustness against uncertain disturbances, while minimizing control effort and energy consumption.

1.3 Research Objectives

The main objectives of this research work are:

- Design of a robust nonlinear controller for the UAV system.
- Approximation of an adaptive nonlinear gain to deal with uncertain disturbances using different NN-based algorithms and their comparison.
- Comparison of the proposed methodology with the classical or fixed-gain nonlinear controller.

1.4 Research Methodology

This research outlines a systematic process for determining the optimal adaptive gain by leveraging feedforward neural networks (FNNs) and radial basis function neural networks (RBFNNs) to significantly improve SMC performance. The proposed approach aims to enhance tracking performance and disturbance rejection for UAVs during flight operations while minimizing control effort. The NN-based adaptive weights effectively counteract the adverse effects of noise, wind gusts, and system

parameter variations, ensuring robust performance in dynamic flight conditions. The block diagram of the proposed scheme is shown in Figure 1.1.

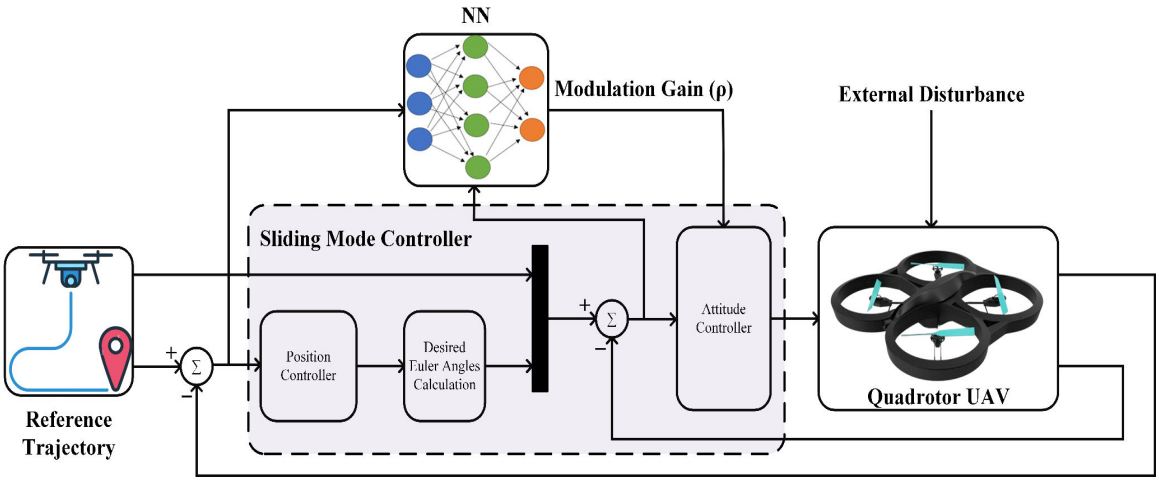

Figure 1.1: Block Diagram of NN-based SMC for Quadrotor UAV

Figure 1.1 illustrates the working of the proposed NN-based SMC for a quadrotor UAV. The reference trajectory provides the desired positions and orientations, which are compared with the UAV's actual states to compute the tracking error. This error is processed by the SMC, which includes a position controller and a block that computes the desired Euler angles, generating the necessary roll, pitch, yaw, and altitude commands. The state errors are also fed to the NN block, which adaptively estimates the modulation gain, enabling the controller to handle disturbances and modeling uncertainties without relying on conservative fixed bounds. The attitude controller converts the desired Euler angles into motor-control signals for the UAV, enabling it to execute the commanded motions despite external disturbances, such as wind gusts. The UAV's actual states are continuously fed back to the controller, thereby closing the loop and ensuring accurate trajectory tracking with improved robustness.

To assess the algorithm's reliability, performance metrics such as mean squared error (MSE) and root mean squared error (RMSE) are computed. Moreover, if errors exceed a predefined threshold, the NN can be fine-tuned iteratively, often by increasing the number of hidden layers or optimizing other hyperparameters. Finally, the improved algorithm is validated through simulations. The proposed approach demonstrates its

efficacy, offering robust control, accurate tracking, and effective disturbance rejection. Ultimately, it improves UAV performance in challenging terrains. Figure 1.2 illustrates the overall NN training framework and highlights the sequential phases of this approach.

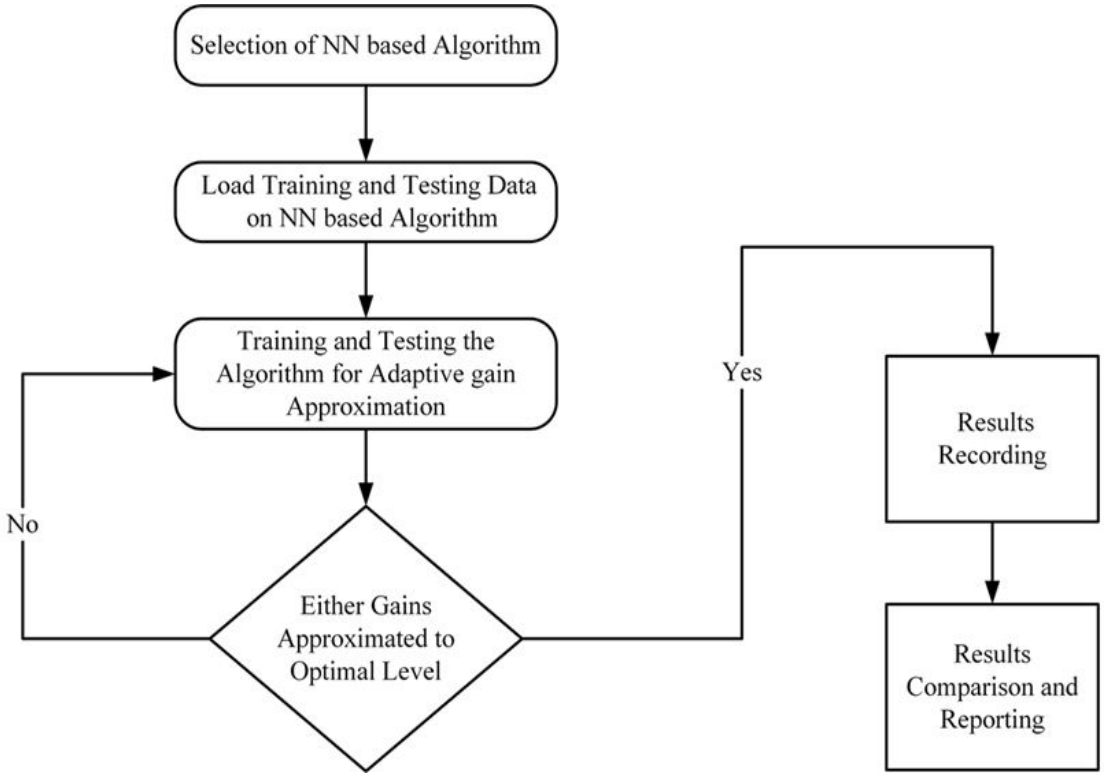

Figure 1.2: Flow chart of NN Training Framework

In Figure 1.2, the flow chart represents the procedure for training and evaluating a neural network (NN)-based algorithm for adaptive gain approximation. The process begins by selecting a suitable NN-based algorithm, then loading the training and test data into the model. The algorithm is then trained and tested to approximate the controller's required adaptive gain. Once training is complete, the gains are evaluated to determine whether they have reached the desired optimal level. If the gains are not optimal, the process loops back to retraining the algorithm until satisfactory performance is achieved. If the gains reach the optimal level, the results are recorded, and then performance metrics are compared and reported. This ensures that the NN-based approach provides an adaptive and accurate gain approximation suitable for the control framework.

1.5 Dissertation Outline

The subsequent chapters of this dissertation are organized as follows:

- The detailed literature review related to adaptive control of UAVs is provided in Chapter 2.
- Chapter 3 presents the mathematical modeling of the quadrotor UAV.
- Chapter 4 starts with the introduction of the conventional sliding surface and the SMC. The design of the sliding surface and the stability analysis of SMC are provided. Then, the adaptive sliding mode control (ASMC) and its Lyapunov analysis are presented for a class of nonlinear systems. Finally, the SMC and ASMC are designed for six DOF control of a quadrotor UAV. The simulation results are presented and discussed.
- Chapter 5 presents the improvements achieved in sliding mode control (SMC) through the integration of adaptive neural network–based techniques. A detailed theoretical background of neural networks is provided, covering artificial neural networks (ANNs), feedforward neural networks (FNNs), the backpropagation (BP) algorithm, the Levenberg–Marquardt (LM) algorithm, and radial basis function neural networks (RBFNNs). Throughout this thesis, SMC frameworks enhanced using FNNs and RBFNNs are referred to as follows:

  1. SMC improved using FNN with the BP algorithm for optimization, will be called adaptive back propagation sliding mode control (ABPSMC).
  2. SMC improved using FNN with the LM algorithm for optimization, will be called adaptive Levenberg-Marquardt sliding mode control (ALMSMC).
  3. SMC improved using RBFNN, will be called adaptive radial basis function sliding model control (ARBFSMC).

  Simulation results for the three proposed controllers, ABPSMC, ALMSMC, and ARBFSMC, are presented and discussed.

- Chapter 6 provides a comprehensive performance comparison between classical SMC and the NN-enhanced SMC schemes, namely ABPSMC, ALMSMC, and ARBFSMC.
- Chapter 7 concludes the thesis and also outlines potential directions for future research.

Summary

This chapter presents an overview of UAV control and highlights the challenges associated with their control due to under-actuation, nonlinear dynamics, strong coupling, external disturbances, and parameter uncertainties. The limitations of classical linear and nonlinear control strategies are discussed, which motivates the need for robust and adaptive control approaches. NN-based ASMC is proposed as a promising solution. The research objectives, contribution, and the overall structure of the thesis are also provided.

Chapter 2
Literature Review

In recent years, UAVs have been widely used in military operations of search and rescue during disastrous situations [2], intelligent surveillance [3], smart agriculture [4–7], aerial photography [8], aerial mapping [9], industry inspection [10], logistics [11, 12], and environmental inspection, particularly through remote monitoring of coastal areas to assess and manage the health and sustainability of coastal ecosystems [13]. An example of a UAV is the quadrotor, which exhibits characteristics such as vertical takeoff and landing (VTOL), stable hovering, convenient carrying, small size, and a payload capacity comparable to that of fixed-wing aircraft [14, 15]. However, quadrotor UAVs face several challenges during flight operations, including external disturbances, aerodynamic drag, and wind gusts. Therefore, to achieve smooth and reliable flight operations, the attitude and altitude control of quadrotor UAVs is becoming increasingly important [16]. The attitude controller of a quadrotor is usually designed to be an under-actuated subsystem owing to the highly nonlinear characteristics associated with the inherent unmodeled dynamics, parameter perturbation, and external disturbance [17, 18]. A quadrotor is a special example of an under-actuated system with 6 degrees of freedom (DOF) and four control inputs. This leads to the design of a control law that comprises two loops: an outer loop for position control and an inner loop for attitude control [13].

To date, various control techniques in the literature have been proposed to effectively manage the flight operations of quadrotor UAVs, including nonlinear, robust, and adaptive control [18]. However, SMC is one of the most promising techniques to handle quadrotors efficiently, owing to the attributes of fast response and strong robustness to deal with system uncertainties and external disturbances with known bounds [19]. A comparison of different control schemes with SMC is presented in [20] to demonstrate its robustness and accuracy.

Despite the strong above-mentioned abilities of a classical SMC, a few challenges

faced by the SMC that need to be resolved are: chattering of high-frequency control signals, non-convergence in finite time, application of large control signal to overcome the parametric uncertainties, the requirement of exact and accurate knowledge of the plant dynamics for the development of equivalent control, prior knowledge of the magnitude of the disturbance and uncertainty. To address the challenges associated with SMC, particularly gain tuning and chattering reduction, various advanced methods have been proposed in the literature to enhance the classical SMC framework. These methods aim to improve the robustness, adaptability, and precision of SMC by introducing modifications that address specific limitations [21, 22].

In view of the above discussion, ASMC is proposed to stabilize and mitigate parametric uncertainties in finite time. A second-order SMC is deployed in [23, 24] to handle the attitude and position control of a small-sized quadrotor. Likewise, in [17], a fast dynamic terminal SMC is proposed for finite time tracking of attitude and position control in the presence of external disturbances for a small quadrotor UAV. In [23], an adaptive fuzzy-gain scheduling SMC is proposed to control the quadrotor attitude in the presence of external disturbances like wind gusts and parametric uncertainties. Similarly, in [25], a novel robust terminal SMC is deployed in conjunction with SMC to handle actuated and under-actuated quadrotor UAVs, respectively. The authors of [26] proposed an adaptive integral SMC to avoid chattering and stabilize multiple-input multiple-Output (MIMO) systems, addressing both matched and unmatched uncertainties. The effectiveness of the proposed controller is demonstrated by its ability to stabilize the VTOL of the aircraft. In [27], second-order super twisting SMC, along with certain modifications, is employed in real-time to track the altitude in outdoor environments.

The main challenge associated with the SMC is tuning the gains of the sliding surfaces. Generally, the SMC is described in two phases: the reaching phase, in which trajectories can be initiated from any given initial point and move towards the fixed sliding surface. During the reaching phase, the system's response is sensitive to disturbances and parameter variations, making direct control difficult. The second phase is the sliding phase, where the trajectories are insensitive to disturbances and

parameter uncertainties. In other words, the reaching phase of SMC is not robust. Therefore, various methods have been proposed to shorten or eliminate the reaching phase. To accomplish the above task, a time-varying sliding surface is chosen as an efficient alternative to a constant sliding surface. However, these methods rely on knowledge of system models that are not readily available. One viable solution to address this effect is to use NN-based techniques to enhance the controller's robustness. The conventional SMC assumes that the disturbance bound is known for all time, a stringent condition for controller design that necessitates unnecessary maintenance of higher cumulative controller effort. This can also be handled using neural networks trained on the available data.

In this regard, the authors in [28] employ a global SMC combined with fuzzy logic for trajectory control in the presence of uncertainties and disturbances. In the said method, chattering, disturbances, and the reaching phase are eliminated using an adaptive law and fuzzy rules. In [29], the authors proposed an adaptive-fuzzy SMC based on a radial basis function (RBF) neural network. RBFNN is used to approximate actuator faults, uncertainties, and disturbances, whereas the chattering problem is controlled using fuzzy logic. Motivated by further improvement, the authors in [30] suggested a backstepping control in conjunction with neural networks to control the attitude and altitude of the quadrotor. Furthermore, in the field of aircraft control, a robust ASMC method enhanced by neural networks is proposed in [31]. To estimate model uncertainties and disturbances using neural networks, the authors in [32] employed a minimum learning parameter (MLP)-based neural network scheme for an underactuated unmanned surface vehicle (USV). Radial basis function neural networks (RBFNNs) are used within an adaptive disturbance observer to estimate the total disturbance, including unknown nonlinearities and external disturbances [33]. Authors in [34] used evidential neural networks (ENNs) to estimate disturbance/ uncertainty by modeling outputs as Dirichlet distributions.

In today's technologically advanced era, neural networks are recognized as among the most intelligent and powerful tools for learning unknown parameters, particularly due to their online learning capabilities. This enables them to adapt in real time,

making them well-suited for applications that require continuous updates based on changing inputs. Neural networks also have the unique ability to handle system nonlinearity and maintain fault tolerance, making them well-suited to complex, dynamic environments where conventional methods may struggle to adapt effectively [35]. Given these advantages, the present study aims to design and develop a NN-based ASMC controller tailored explicitly for quadrotor UAVs. By integrating this approach, the controller can handle external disturbances and model uncertainties, thereby enhancing the quadrotor's ability to perform stable, precise flight maneuvers. This neural network-enhanced SMC framework is expected to enable the UAV to achieve satisfactory maneuverability, even in challenging or unpredictable conditions. Table 2.1 given below represents the summary of work done in recent years by using different variants of SMC along with NN.

Table 2.1: Overview of the Recent Studies Relevant to Present Research

Ref.	**Year**	**Methodology**	**Purpose**	**Findings**
[21]	2018	ASMC	Position control of quadrotor with finite time convergence achieved	Dynamic control is used for disturbance bound estimation; Finite time stability is illustrated when an initial bound on disturbance is known
[22]	2019	ASMC, NN	Sliding surface gain is adopted using NN to achieve robustness in reaching phase; ASMC handled external disturbance	Improved transient behavior of closed loop system is shown using proposed method; The reaching phase robustness is shown using known upper bound of disturbances

Ref	Year	Methodology	Purpose	Findings
[36]	2020	Conventional SMC, NN, EKF	Position and attitude control of quadrotor; EKF is used to counter the sensor noise; NN is used to estimate unknown state-dependent uncertainties	Small cumulative tracking error and improved stability is achieved; The noise cancellation is achieved using EKF
[37]	2021	ISMC, FCRNN	Position and attitude control of quadrotor; Quadrotor parameters are assumed to be unknown, and FCRNN is used for estimation of parametric uncertainties	The superiority of FCRNN is shown by comparison with RNN and RBFNN; The robustness under bounded disturbances is illustrated
[38]	2021	RBFNN, Discrete time SMC	Discrete time control used for position and attitude control of Qball-X4; RBFNN is used to approximate the model uncertainty	Satisfactory tracking performance of the position and attitude is achieved; Decoupling problem to handle under-actuation is mitigated by using discrete control

Ref	Year	Methodology	Purpose	Findings
[39]	2022	STSMC, RBFNN, DO	DO is used to estimate the mismatched disturbance; RBFNN estimates the unmodeled dynamics and refines the STSMC performance	Effectively reduced chattering effect due to use of DO and RBFNN; Improved robustness to both mismatched and matched disturbances while bound for matched disturbance is known a priori
[40]	2023	FCRNN, TSMC	Unknown system parameters are estimated using FCRNN; The actuator fault compensation is also achieved	Comparison of FCRNN-based TSMC is provided with RBFNN-based piecewise TSMC; The computational efficiency of FCRNN-based controller is validated in Gazebo
[41]	2024	RBFNN, SMC	Linearized model is used for position tracking; RBFNN provides adaptation to SMC	It is shown that the increase in required control input due to RBFNN is negligible compared to its impact; Theoretically, it is shown that blade malfunction can also be catered using RBFNN along with SMC

From the literature perspective, very limited work exists on the use of different neural network techniques and their comparison for under-actuated UAVs, especially for estimating the modulation gain of robust ASMC. Existing studies have demonstrated the effectiveness of neural network algorithms for estimating controller gains in some cases. This study aims to bridge this gap by fine-tuning several neural network models and comparing them to achieve an optimized solution for designing a robust nonlinear UAV controller that outperforms classical fixed-gain nonlinear controllers.

Chapter 3
Mathematical Modeling of Quadrotor UAV

3.1 Nonlinear Mathematical Modeling of Quadrotor UAV [1]

The quadrotor UAV has four propellers for aerial maneuvering and control. Each rotor has a propeller and a motor. These propellers are mounted on each of the four arms so that two spin clockwise and the other two counterclockwise, balancing torque and providing upward thrust. The structure of a typical quadrotor UAV is shown in Figure 3.1.

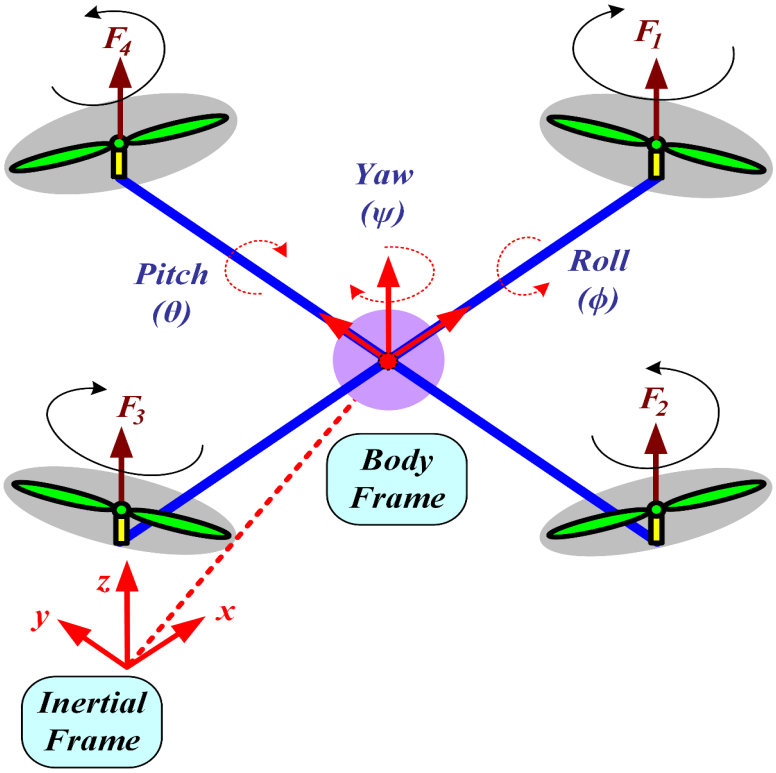

Figure 3.1: Quadrotor UAV Structure

In literature, the following are the generic assumptions being considered for mathematical modeling of quadrotor UAV [1];

- The quadrotor has a symmetric and rigid structure, with the gravitational center positioned at the origin of the structure.

- The structure of the rotors is also rigid.
- The upward-thrust/lift as well as drag force provided by rotors are quadratically proportional to the speed of the rotors.
- The diagonal inertial matrix is applied here.
- The *North-East-Down (NED)* convention is followed to represent the inertial frame of reference.
- The inductance and resistance of the motors are neglected, implying that the motor dynamics are considerably faster than the quadrotor's.

These stated assumptions substantially reduce the complexity of mathematical modeling for a quadrotor UAV. Nevertheless, the modeling and control of this system remain challenging because a quadrotor UAV is underactuated. It has four governing inputs to control six degrees of freedom (DOF). These DOF include three translational motions and three rotational motions as depicted in Figure 3.1. x, y, and z denote the translational motions along the x-axis, y-axis, and z-axis, respectively. The rotational motions of roll, pitch, and yaw are represented by ϕ, θ, and ψ, respectively. The associated angular velocities are denoted by p, q, and r. The rotational speed of the i-th rotor is denoted by Ω_i, (where $i = 1, 2, 3, 4$). The four control inputs that govern the quadrotor include upward-thrust (U_1), rolling torque (U_2), pitching torque (U_3), and yawing torque (U_4). Twelve state variables define the complete model of a quadrotor UAV:

- **Translational states:** Three translational positions x, y, z defined in the Earth-fixed (inertial) frame along with their associated linear velocities $\dot{x}$, $\dot{y}$, $\dot{z}$.
- **Rotational states:** Three orientation angles ϕ, θ, ψ defined in inertial frame along with their associated angular rates $\dot{\phi}$, $\dot{\theta}$, $\dot{\psi}$.

The orientation of the quadrotor UAV in two different frames of reference is related through the rotation matrix $\mathbf{R}$, derived from Euler angles: roll, pitch, and yaw. The

overall rotation matrix is obtained by successively multiplying different matrices and is defined as follows:

$$\mathbf{R} = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi + \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi - \sin\phi\sin\psi \\ -\cos\theta\sin\psi & -\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & -\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \tag{3.1}$$

Once this matrix is known, it becomes straightforward to transform the velocities (u, v, and w) defined to be the velocities along the x-axis, y-axis, and z-axis, respectively in body-frame into the inertial-frame velocities $\dot{x}, \dot{y}$ and $\dot{z}$. The relation is as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \mathbf{R} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{3.2}$$

The attitude vector in the inertial frame is represented as $\boldsymbol{A}^E = \begin{bmatrix} \phi, \theta, \psi \end{bmatrix}^T$, while the angular velocity in the body frame is given by $\boldsymbol{\Omega}^B = \begin{bmatrix} p, q, r \end{bmatrix}^T$. To relate these two, transfer matrix $\mathbf{T}$ is used as: $\dot{\boldsymbol{A}}^E = \mathbf{T}\boldsymbol{\Omega}^B$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \mathbf{T} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.3}$$

where,

$$\mathbf{T} = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)\sec(\theta) & \cos(\phi)\sec(\theta) \end{bmatrix} \tag{3.4}$$

External Forces and Moments:

Resultant Force: The motion of a quadrotor is affected by the forces acting on it [42, 43]. According to Newton's second law, the total force applied can be written

as:

$$\sum \mathbf{F} = \mathbf{F}_{\text{th}} + \mathbf{F}_{\text{grav}} = m\vec{a} \tag{3.5}$$

In the body-fixed frame, translational motion is mainly affected by two forces:

Thrust Force: This is the upward force generated by all rotors together to uplift the quadrotor. The thrust from each rotor depends on the square of its angular speed: $\vec{F_i} = b\vec{\Omega_i^2}$

$$\mathbf{F}_{\text{th}}^{\text{B}} = \mathbf{F_1} + \mathbf{F_2} + \mathbf{F_3} + \mathbf{F_4} = b\vec{\Omega_1^2} + b\vec{\Omega_2^2} + b\vec{\Omega_3^2} + b\vec{\Omega_4^2} \tag{3.6}$$

Gravitational Force: The gravitational force affects the UAV along the z-axis. Euler angles can be used to express this force in the body-fixed frame; using the Earth's inertial frame provides a more accurate expression [43].

$$\mathbf{F}_{\text{grav}}^{B} = \left({}_{B}^{E}\mathbf{R}\right)^{-1} \mathbf{F}_{\text{grav}}^{E}, \quad \text{where:} \quad \left({}_{B}^{E}\mathbf{R}\right)^{-1} = \left({}_{B}^{E}\mathbf{R}\right)^{T}, \quad \text{and} \tag{3.7}$$

$$\mathbf{F}_{\text{grav}}^{E} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \tag{3.8}$$

The translational forces are described in the Earth's inertial reference frame, and the rotation matrix can be applied to express the thrust force.

$$\mathbf{F}_{\text{grav}}^{B} = m\mathbf{g} \cdot \begin{bmatrix} \sin(\theta) \\ -\cos(\theta)\sin(\phi) \\ -\cos(\theta)\cos(\phi) \end{bmatrix} \tag{3.9}$$

Applying the Newton-Euler formalism for force balance in the body frame:

$$\sum \mathbf{F}^{B} = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = m \cdot \begin{bmatrix} \dot{u} + (qw - vr) \\ \dot{v} + (ru - pw) \\ \dot{w} + (pv - uq) \end{bmatrix} \tag{3.10}$$

The total force $\sum \mathbf{F}^{B}$ in the body frame is the combination of thrust and gravitational

forces:

$$\mathbf{F}^B_{\text{total}} = \mathbf{F}^B_{\text{th}} + \mathbf{F}^B_{\text{grav}} = m \cdot \mathbf{a} \tag{3.11}$$

The thrust vector in the body frame typically acts along the body Z-axis (assuming a multi-rotor UAV):

$$\mathbf{F}^B_{\text{th}} = \begin{bmatrix} 0 \\ 0 \\ F_1 + F_2 + F_3 + F_4 \end{bmatrix} \tag{3.12}$$

Combining the thrust and gravity contributions, the equation of motion in the body frame becomes:

$$\begin{bmatrix} 0 \\ 0 \\ F_1 + F_2 + F_3 + F_4 \end{bmatrix} + mg \cdot \begin{bmatrix} \sin(\theta) \\ -\cos(\theta)\sin(\phi) \\ -\cos(\theta)\cos(\phi) \end{bmatrix} = m \cdot \begin{bmatrix} \dot{u} + (qw - vr) \\ \dot{v} + (ru - pw) \\ \dot{w} + (pv - uq) \end{bmatrix} \tag{3.13}$$

From this, we isolate the linear accelerations:

$$\begin{cases} \dot{u} = g\sin(\theta) - (qw - vr) \\ \dot{v} = -g\sin(\phi)\cos(\theta) - (ru - pw) \\ \dot{w} = -g\cos(\phi)\cos(\theta) - (pv - uq) + \dfrac{F^B_{\text{th}}}{m} \end{cases} \tag{3.14}$$

To interpret the body-frame thrust in the Earth-fixed (inertial) frame, a rotation matrix is used:

$$\mathbf{F}^E_{\text{th}} = \left({}^E_B R\right) \cdot \mathbf{F}^B_{\text{th}} \tag{3.15}$$

This transformation results in the following components in the Earth frame:

$$\mathbf{F}_{\text{th}}^{E} = F_{\text{th}}^{B} \cdot \begin{bmatrix} \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi) \\ \cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi) \\ \cos(\phi)\cos(\theta) \end{bmatrix} \tag{3.16}$$

The gravitational force in the Earth-fixed inertial reference frame is directed vertically downward and is represented as:

$$\mathbf{F}_{\text{grav}}^{E} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \tag{3.17}$$

In this frame, the net force is the sum of the thrust and gravitational forces acting on the object. According to Newton's second law, this total force is equal to the mass times the acceleration vector:

$$\mathbf{F}_{\text{net}}^{E} = \mathbf{F}_{\text{th}}^{E} + \mathbf{F}_{\text{grav}}^{E} = m \cdot \mathbf{a} \tag{3.18}$$

To describe the thrust force generated by the vehicle's rotors, we express it as the total force generated from all four rotors:

$$U_1 = F_{\text{th}}^{B} = b \cdot \left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right) \tag{3.19}$$

Substituting this thrust into the equation for total force, the translational acceleration components along the global X, Y, and Z directions are derived as follows:

$$\begin{cases} \ddot{x} = \dfrac{U_1}{m}\left(\cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi)\right) \\ \ddot{y} = \dfrac{U_1}{m}\left(\cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi)\right) \\ \ddot{z} = \dfrac{U_1}{m}\left(\cos(\phi)\cos(\theta)\right) - g \end{cases} \tag{3.20}$$

Similar to linear dynamics, the rotational dynamics can be expressed by considering net torque and reaction forces:

$$\sum \tau = \mathbf{M}_{\text{thrust}} + \mathbf{F}_{\text{gyro}} \tag{3.21}$$

The quadrotor's rotational dynamics can be derived using both reference frames. In the body-fixed frame, the moments from thrust forces ($\mathbf{M}_{\text{thrust}}$) can be expressed as:

$$\mathbf{M}_{\text{thrust}} = \sum(\mathbf{r}_i \times \mathbf{F}_i) \tag{3.22}$$

Considering clockwise propeller rotation as negative, while counter-clockwise as positive, we get:

$$\begin{cases} \tau_x^B = \dfrac{l}{\sqrt{2}}\left(-F_1 + F_2 + F_3 - F_4\right) \\ \tau_y^B = \dfrac{l}{\sqrt{2}}\left(F_1 + F_2 - F_3 - F_4\right) \\ \tau_z^B = -M_1 + M_2 - M_3 + M_4 \end{cases} \tag{3.23}$$

From Equation (3.23), let:

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

The equilibrium is achieved when the combined speeds of all rotors produce no net yaw torque. This condition is met when the torques produced by the rotors cancel each other to nullify the net yaw torque. If this balance is disrupted, it can lead to instability, potentially introducing gyroscopic effects that impact the roll and pitch dynamics. The following equation can express the collective speed of the propellers:

$$\mathbf{F}_{\text{gyro}}^B = -\sum_{k=1}^{4} Jr \left(\begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right) (-1)^k \Omega_k \tag{3.24}$$

$$\mathbf{F}_{\text{gyro}}^B = Jr \left(\begin{bmatrix} q(\Omega_r) \\ p(-\Omega_r) \\ 0 \end{bmatrix} \right) \tag{3.25}$$

The time rate of change of angular momentum (L) is equal to the sum of all external torques applied to the quadrotor. This principle is known as Euler's second law, also

called the method of torque balancing in this context.

$$\left(\frac{dL}{dt}\right)_E = \frac{d}{dt}\left(I_B\Omega\right) \tag{3.26}$$

where I_B is the inertia and Ω is angular velocity. Owing to geometric symmetry, the quadrotor's rigid-body inertia is given by:

$$J = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{3.27}$$

Angular momentum

$$(L_E) = I\Omega = \begin{bmatrix} I_{xx}\Omega_1 \\ I_{yy}\Omega_2 \\ I_{zz}\Omega_3 \end{bmatrix}_E \tag{3.28}$$

$$L_E = \begin{bmatrix} I_{xx}p \\ I_{yy}q \\ I_{zz}r \end{bmatrix}_E \tag{3.29}$$

For Body frame,

$$\left(\frac{dL}{dt}\right)_{\text{B-frame}} = \left(\frac{dL}{dt}\right)_{\text{E-frame}} + \Omega \times L \tag{3.30}$$

$$\tau^B = \dot{L} + \Omega \times L \tag{3.31}$$

This is referred to as the Euler Equation.

$$\tau^B = \begin{bmatrix} I_{xx}\dot{p} \\ I_{yy}\dot{q} \\ I_{zz}\dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_{xx}p \\ I_{yy}q \\ I_{zz}r \end{bmatrix} \tag{3.32}$$

$$\begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} I_{xx}\dot{p} \\ I_{yy}\dot{q} \\ I_{zz}\dot{r} \end{bmatrix} + \begin{bmatrix} rqI_{zz} - rqI_{yy} \\ rpI_{xx} - rpI_{zz} \\ pqI_{yy} - pqI_{xx} \end{bmatrix} \tag{3.33}$$

The motion generated with respect to the Body Frame, along with the total torque τ exerted in the Body Frame of Reference, is described by Equation 3.34

$$\sum \vec{M}^B = \vec{M}^B_{\text{th}} + \vec{F}^B_{\text{gyro}} = \dot{\vec{L}} + \vec{\Omega} \times \vec{L} \tag{3.34}$$

$$\underbrace{\begin{bmatrix} I_{xx}\dot{p} + rq(I_{xx} - I_{yy}) \\ I_{yy}\dot{q} + rp(I_{zz} - I_{xx}) \\ I_{zz}\dot{r} + pq(I_{yy} - I_{xx}) \end{bmatrix}}_{\dot{\vec{L}} + \vec{\Omega} \times \vec{L}} = \underbrace{\begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix}}_{\vec{M}^B_{\text{Thrust}}} + J_r \left(\underbrace{\begin{bmatrix} q\Omega_r \\ -p\Omega_r \\ 0 \end{bmatrix}}_{\vec{F}^B_{\text{gyro}}} \right) \tag{3.35}$$

Angular acceleration expressed in Body Frame of Reference:

$$\begin{cases} \dot{p} = \frac{U_2}{I_{xx}} - \frac{qr(I_{xx} - I_{yy})}{I_{xx}} + \frac{J_r \Omega_r q}{I_{xx}} \\ \dot{q} = \frac{U_3}{I_{yy}} - \frac{rp(I_{zz} - I_{xx})}{I_{yy}} - \frac{J_r \Omega_r p}{I_{yy}} \\ \dot{r} = \frac{U_4}{I_{zz}} - \frac{pq(I_{yy} - I_{xx})}{I_{zz}} \end{cases} \tag{3.36}$$

Equating equations (3.20),(3.35), (3.36) and considering the small-angle approximations for roll (ϕ), pitch (θ), and yaw (ψ) angles provides the nonlinear

dynamical equations of the quadrotor as also derived by author in [1]:

$$\begin{cases} \ddot{x} = (\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi)\dfrac{U_1}{m} \\ \ddot{y} = (\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi)\dfrac{U_1}{m} \\ \ddot{z} = -g + (\cos\phi \cos\theta)\dfrac{U_1}{m} \\ \ddot{\phi} = \dot{\theta}\dot{\psi}\left(\dfrac{I_y - I_z}{I_x}\right) - \dfrac{I_r}{I_x}\dot{\theta}\Omega + \dfrac{U_2}{I_x} \\ \ddot{\theta} = \dot{\phi}\dot{\psi}\left(\dfrac{I_z - I_x}{I_y}\right) - \dfrac{I_r}{I_y}\dot{\phi}\Omega + \dfrac{U_3}{I_y} \\ \ddot{\psi} = \dot{\phi}\dot{\theta}\left(\dfrac{I_x - I_y}{I_z}\right) + \dfrac{U_4}{I_z} \end{cases} \tag{3.37}$$

where U_1, U_2, U_3, and U_4 are the control inputs, which will be defined later in the controller design chapter. The quadrotor UAV has the following states:

$$\left[x\ \dot{x}\ y\ \dot{y}\ z\ \dot{z}\ \phi\ \dot{\phi}\ \theta\ \dot{\theta}\ \psi\ \dot{\psi}\right] = [x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7\ x_8\ x_9\ x_{10}\ x_{11}\ x_{12}] \tag{3.38}$$

The state-space representation of the UAV's dynamic model, as given in Equation

3.38, is shown below:

$$
\begin{cases}
\dot{x}_1 = \dot{x} = x_2 \\
\dot{x}_2 = \ddot{x} = \left(\cos x_7 \sin x_9 \cos x_{11} + \sin x_7 \sin x_{11}\right) \dfrac{U_1}{m} \\
\dot{x}_3 = \dot{y} = x_4 \\
\dot{x}_4 = \ddot{y} = \left(\cos x_7 \sin x_9 \sin x_{11} - \sin x_7 \cos x_{11}\right) \dfrac{U_1}{m} \\
\dot{x}_5 = \dot{z} = x_6 \\
\dot{x}_6 = \ddot{z} = -g + \left(\cos x_7 \cos x_9\right) \dfrac{U_1}{m} \\
\dot{x}_7 = \dot{\phi} = x_8 \\
\dot{x}_8 = \ddot{\phi} = x_{10} x_{12} \left(\dfrac{I_y - I_z}{I_x}\right) - \dfrac{I_r}{I_x} x_{10} \Omega + \dfrac{U_2}{I_x} \\
\dot{x}_9 = \dot{\theta} = x_{10} \\
\dot{x}_{10} = \ddot{\theta} = x_8 x_{12} \left(\dfrac{I_z - I_x}{I_y}\right) - \dfrac{I_r}{I_y} x_8 \Omega + \dfrac{U_3}{I_y} \\
\dot{x}_{11} = \dot{\psi} = x_{12} \\
\dot{x}_{12} = \ddot{\psi} = x_8 x_{10} \left(\dfrac{I_x - I_y}{I_z}\right) + \dfrac{U_4}{I_z}
\end{cases}
\tag{3.39}
$$

3.2 External Disturbances and Uncertainties

In real-world scenarios, quadrotor UAVs are subject to external disturbances and model uncertainties. These effects are associated with atmospheric conditions, aerodynamic interactions, actuator imperfections, and unmodeled dynamics. These disturbances significantly affect translational and rotational motion, particularly in low-speed flight and hover [44, 45]. To evaluate the performance of the proposed control strategy, the external disturbances are directly applied to the quadrotor dynamic model.

Instead of explicitly modeling these disturbances and uncertainties, all external perturbations can be represented as a lumped disturbance. This approach aggregates multiple disturbance sources into a unified set of disturbance terms, thereby simplifying the system representation while preserving the dominant effects governing its behavior.

Lumped disturbance modeling is widely used in the nonlinear and robust control literature and is particularly suitable for sliding-mode control frameworks [46, 47]. The lumped disturbance considered in this work includes, but is not limited to:

- Wind gusts and atmospheric turbulence.
- Unmodeled aerodynamic drag and lift effects.
- Rotor–wake interactions and airflow asymmetries.
- Parametric uncertainties and unmodeled nonlinearities.

The translational dynamics of the quadrotor, as described by the equation (3.5) in the quadrotor modeling section, will now take the following form in the presence of external disturbances and model uncertainties.

$$\sum \mathbf{F} = \mathbf{F}_{\text{th}} + \mathbf{F}_{\text{grav}} + \mathbf{F}_{\text{d}} = m\vec{a} \tag{3.40}$$

where, $\mathbf{F}_d$ denotes forced applied to the quadrotor by the lumped disturbance [45]. Similar to translational dynamics, rotational dynamics can be described by the following equation in the presence of external disturbances and model uncertainties.

$$\sum \tau = \mathbf{M}_{\text{thrust}} + \mathbf{F}_{\text{gyro}} + \tau_{\text{d}} \tag{3.41}$$

where, $\boldsymbol{\tau}_d$ denotes the lumped disturbance moment vector. The combined effects of the lumped disturbance acting on both translational and rotational dynamics are denoted by Δ for controller design purposes, hereafter throughout this thesis. In accordance with standard assumptions in SMC theory, the external disturbance force and moment vectors are assumed to be unknown but bounded, such that

$$\|\mathbf{F}_d(t)\| \leq \bar{F}_d, \quad \|\boldsymbol{\tau}_d(t)\| \leq \bar{\tau}_d, \tag{3.42}$$

where $\bar{F}_d$ and $\bar{\tau}_d$ are positive constants representing the maximum disturbance bounds. For controller design, the combined effects of external disturbances, unmodeled

dynamics, and parametric uncertainties acting on both the translational and rotational dynamics are lumped into a single equivalent disturbance term, denoted by Δ. This lumped disturbance is assumed to be unknown but bounded, satisfying

$$|\Delta| \leq \gamma, \tag{3.43}$$

where γ is a positive constant representing the upper bound of the lumped disturbance. This assumption is physically justified by finite aerodynamic loading, actuator limitations, and structural constraints inherent to small-scale quadrotor UAVs [46, 47]. Notably, the exact values of these bounds are not required for controller implementation, provided that the control gains are selected to dominate the worst-case disturbances [48]. The external disturbances considered in this work exhibit the following characteristics:

- Time varying behavior due to changing atmospheric conditions.
- Continuous and slowly varying profiles representative of wind-induced effects.
- Matched disturbances entering the system through the same channels as the control inputs.

These characteristics are consistent with practical quadrotor operating environments and satisfy the theoretical requirements for SMC design [47].

Summary

This chapter presents the nonlinear mathematical model of the quadrotor UAV, derived using the Newton–Euler formulation. The translational and rotational dynamics corresponding to six degrees of freedom and twelve state variables are developed. The relationship between the rotor thrusts and the control input has been established. This model forms the basis for controller design and simulation analysis. Details of external disturbances and model uncertainties are also discussed.

Chapter 4
Adaptive Sliding Mode Control

4.1 Sliding Mode Control (SMC)

SMC is derived from the framework of variable-structure systems (VSS), in which multiple control laws are defined over distinct regions of the system's state space. Instead of employing a fixed control strategy, the system transitions between these laws based on predetermined switching conditions, thereby ensuring robust performance in a closed-loop configuration.

SMC is a nonlinear strategy that alters system dynamics via a discontinuous control input, compelling the system state to move along a designated manifold, known as the sliding surface. The core concept of SMC is that regulating the lower-order (first-order) dynamics, even in the presence of nonlinearity or uncertainty, is more tractable than directly handling higher-order terms. SMC is developed using a two-phase method. Initially, a specific sliding surface is formulated to define the desired system behavior. Subsequently, a discontinuous control law is constructed to ensure that the system trajectory reaches and remains on this surface. Once the system's states are constrained to this surface, the resulting behavior is called sliding motion. For elaborating the mathematical essence of SMC, let us consider a nonlinear system, defined by the equation 4.1 below:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= f(x_1, x_2) + g(x_1, x_2)u + \Delta \\ y &= x_1 \end{aligned} \tag{4.1}$$

where Δ is a matched disturbance, which includes modeling uncertainties, external disturbances, and parametric variations. Moreover, $x \equiv [x_1, x_2]^T \in \mathbb{R}^2$ represents the state vector, $u \in \mathbb{R}$ is the input, $y \in \mathbb{R}$ represents the plant output. It is considered that $f(x) \equiv f(x_1, x_2)$ and $g(x) \equiv g(x_1, x_2)$ are known and smooth.

4.1.1 Sliding Surface Design

The sliding surface is a stable manifold constructed from the desired system output. Appropriate control actions are required to keep the system state on this surface. When the condition is maintained, and state trajectories reach the sliding manifold, the control objective is achieved. Furthermore, once the system state reaches and remains on the sliding surface, its behavior is governed not by the original system dynamics, but by the properties of the sliding surface itself. [49]

There is no single, fixed method for designing the sliding surface. Its structure can vary depending on the control objectives and specific system requirements. Several commonly used forms of sliding surfaces exist in the literature. One general approach to defining a sliding surface is as follows:

$$s = s(x, t),$$

i.e., the surface is a function of system states and time. To establish sliding mode, the condition $s(x, t) = 0$ must be satisfied.

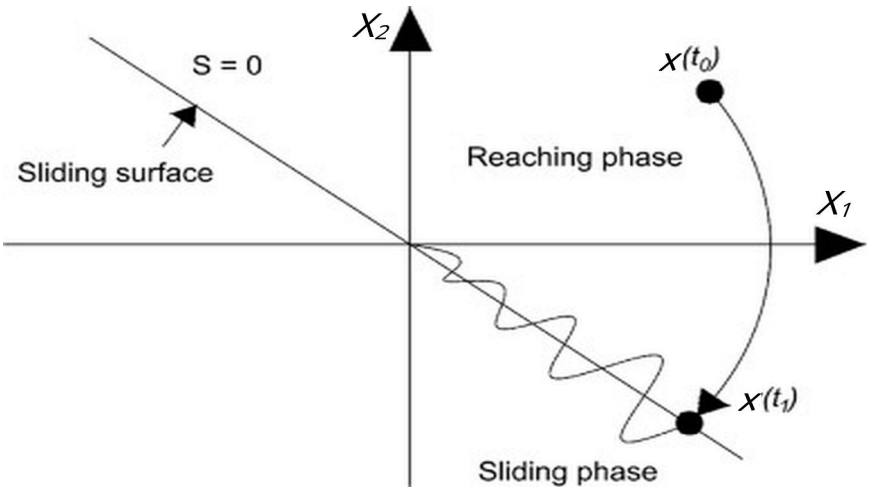

Figure 4.1: Sliding Surface

Conventionally, a linear sliding surface is used for addressing the tracking problem, and its design is given as:

$$s = \dot{e} + \beta e \tag{4.2}$$

where $e = x_1 - r$ is the error while tracking a reference r and β is a positive real constant. The phase portrait of a typical closed-loop second-order system with sliding mode is provided in Figure 4.1.

4.1.2 Stability of a Sliding Mode

Before addressing the stability proof, it is essential to recall that the primary goal of SMC is to ensure that the system's state trajectories converge to a pre-defined sliding surface. Once the trajectories reach this surface, they must stay on it for all future time. To demonstrate the stability of the sliding surface, a Lyapunov candidate function V must be selected. To show the stability of the system, V should satisfy the following conditions:

- V must be a *positive definite* and radially unbounded function.
- Its time derivative should be *negative definite*, indicating that the function consistently decreases over time, which confirms system stability.

There is no universally defined method for selecting a Lyapunov candidate function. However, for single-input single-output (SISO) systems, a commonly used and effective choice is:

$$V = \frac{1}{2}s^2 \tag{4.3}$$

This function is clearly *positive definite*, as it produces positive values for all non-zero values of the sliding variable s, and is zero only when $s = 0$. If the time derivative $\dot{V}$ satisfies the condition:

$$\dot{V} = s\dot{s} < 0 \tag{4.4}$$

Then, the system's state trajectories will converge to and remain on the sliding surface. The illustration of SMC for different initial conditions can be found in Figure 4.2.

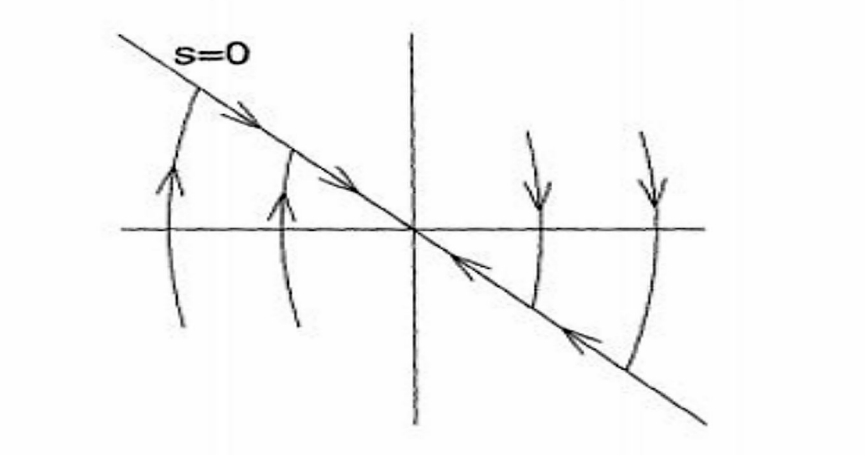

Figure 4.2: Phase Portrait under SMC

The most vibrant feature of SMC is its ability to suppress the effect of matched external disturbances. Typically, in system 4.1, the impact of Δ can be compensated by using SMC, if it is bounded by a positive real constant γ, i.e., $|\Delta| \leq \gamma$. The constant γ should be known a priori.

4.1.3 SMC Control Law Design

Consider the non-linear system defined by Equation 4.1 and the sliding surface defined in Equation 4.2. Taking the time derivative of the Lyapunov function 4.3,

$$\dot{V} = s\dot{s} \tag{4.5}$$

Substituting the expressions for s and $\dot{s}$, we obtain:

$$\dot{V} = s(\ddot{e} + \beta\dot{e}) \tag{4.6}$$

Since $e = x_1 - r$, we get:

$$\dot{V} = s(\ddot{x}_1 - \ddot{r} + \beta\dot{x}_1 - \beta\dot{r}) \tag{4.7}$$

Using the system dynamics 4.1

$$\dot{V} = s\Big(f(x_1, x_2) + g(x_1, x_2)u + \Delta - \ddot{r} + \beta x_2 - \beta\dot{r}\Big) \tag{4.8}$$

In SMC, the control input is given by

$$u = u_l + u_n \tag{4.9}$$

where u_l is the equivalent or nominal control defined by:

$$u_l = -\frac{1}{g(x_1, x_2)} \left(f(x_1, x_2) - \ddot{r} + \beta x_2 - \beta \dot{r} + \mu_1 s \right) \tag{4.10}$$

While u_n is the disturbance compensation control, which is given as:

$$u_n = -\frac{1}{g(x_1, x_2)} \left(k \frac{s}{\|s\|} \right) \tag{4.11}$$

where $k > \gamma$.

By substituting the value of u into the equation 4.8, we get:

$$\begin{aligned} \dot{V} = s \Bigg(& f(x_1, x_2) + g(x_1, x_2) \left[\frac{-1}{g(x_1, x_2)} \left(f(x_1, x_2) - \ddot{r} + \beta x_2 - \beta \dot{r} + \mu_1 s - k \frac{s}{\|s\|} \right) \right] \\ & + \Delta - \ddot{r} + \beta x_2 - \beta \dot{r} \Bigg) \end{aligned} \tag{4.12}$$

The rearrangement of terms leads to

$$\dot{V} = s \left((f(x_1, x_2) - f(x_1, x_2)) + (\ddot{r} - \ddot{r}) + (\beta x_2 - \beta x_2) + \Delta - \mu_1 s - k \frac{s}{\|s\|} \right) \tag{4.13}$$

Since, it is known that $|\Delta| \leq \gamma$

$$\dot{V} \leq s \left(\gamma - \mu_1 s - k \frac{s}{\|s\|} \right) \tag{4.14}$$

By employing the condition that $k > \gamma$

$$\dot{V} < -\mu_1 s^2 \tag{4.15}$$

Finally,

$$\dot{V} < 0 \tag{4.16}$$

Thus, the stability of the system 4.1 is guaranteed by using control input 4.9.

4.1.4 SMC Shortcomings

- In the reaching phase, the system remains sensitive to matched uncertainties, which may lead to undesirable behavior and potentially disturb the desired trajectory.
- SMC assumes prior knowledge of the bounds on system uncertainties, allowing the controller gains to be set slightly above these limits. However, if these bounds are unknown or cannot be estimated, it becomes challenging to ensure the existence of sliding motion.

4.2 Sliding Mode Control for Quadrotor UAV

A quadrotor UAV is an underactuated system with six degrees of freedom (DOF) and four control inputs. This under-actuation requires a control law with two loops: an outer loop for position control and an inner loop for attitude control [13]. The six degrees of freedom are as follows:

The longitudinal dynamics of quadrotor UAV are represented by the following state-space model (4.17):

$$\begin{cases} \dot{x} = x_2 \\ \ddot{x} = (\cos x_7 \sin x_9 \cos x_{11} + \sin x_7 \sin x_{11}) \dfrac{U_1}{m} + \Delta_x \end{cases} \tag{4.17}$$

The lateral position of the quadrotor UAV is represented through the following dynamical equations (4.18):

$$\begin{cases} \dot{y} = x_4 \\ \ddot{y} = \left(\cos x_7 \sin x_9 \sin x_{11} - \sin x_7 \cos x_{11}\right) \dfrac{U_1}{m} + \Delta_y \end{cases} \tag{4.18}$$

The dynamics of height of quadrotor UAV are given by (4.19):

$$\begin{cases} \dot{z} = x_6 \\ \ddot{z} = -g + \left(\cos x_7 \cos x_9\right) \dfrac{U_1}{m} + \Delta_z \end{cases} \tag{4.19}$$

The roll dynamics are given as by (4.20):

$$\begin{cases} \dot{\phi} = x_8 \\ \ddot{\phi} = x_{10} x_{12} \left(\dfrac{I_y - I_z}{I_x}\right) - \dfrac{I_r}{I_x} x_{10} \Omega + \dfrac{U_2}{I_x} + \Delta_\phi \end{cases} \tag{4.20}$$

The pitch orientation of quadrotor UAV are represented by (4.21):

$$\begin{cases} \dot{\theta} = x_{10} \\ \ddot{\theta} = x_8 x_{12} \left(\dfrac{I_z - I_x}{I_y}\right) - \dfrac{I_r}{I_y} x_8 \Omega + \dfrac{U_3}{I_y} + \Delta_\theta \end{cases} \tag{4.21}$$

Finally, the state-space of the yaw dynamics is represented by (4.22):

$$\begin{cases} \dot{\psi} = x_{12} \\ \ddot{\psi} = x_8 x_{10} \left(\dfrac{I_x - I_y}{I_z}\right) + \dfrac{U_4}{I_z} + \Delta_\psi \end{cases} \tag{4.22}$$

Each equation from (4.17) to (4.22), representing a system dynamic, resembles the nonlinear system, already defined by equation 4.1 for which we had already implemented SMC.

The states x and y cannot be controlled directly. In order to achieve desired $x - y$ translation, the control variables τ_{xx} and τ_{yy} can be considered as virtual commands, where:

$$\tau_{xx} = \cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi \tag{4.23}$$

$$\tau_{yy} = \cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi \tag{4.24}$$

The desired expressions for ϕ_d and θ_d can be obtained from these virtual commands specifically:

$$\phi_d = \sin^{-1}(\sin\psi\,\tau_{xx} - \cos\psi\,\tau_{yy}) \tag{4.25}$$

$$\theta_d = \sin^{-1}\left(\frac{\cos\psi\,\tau_{xx} + \sin\psi\,\tau_{yy}}{\cos\phi_d}\right) \tag{4.26}$$

The block diagram of SMC is shown in the following diagram;

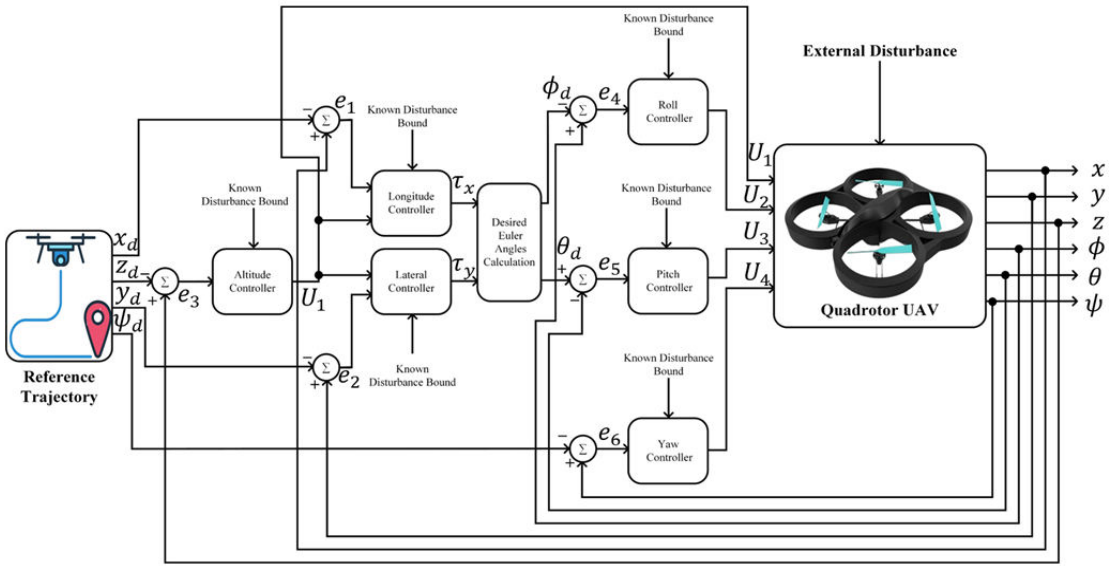

Figure 4.3: SMC Block Diagram

Figure 4.3 illustrates the working of the SMC for a quadrotor UAV. The reference trajectory provides the desired positions and orientations, which are compared with the UAV's actual states to generate tracking errors. The respective sliding mode controller blocks process these errors. Controllers generate the motor control signals required by the UAV, enabling it to execute commanded motions despite external disturbances, such as wind gusts. The disturbance bound is known in this case, so fixed gains are embedded in the controllers. Values of these gains are shown in table 4.2. The UAV's actual states are continuously fed back to the controller, thereby closing the loop and ensuring accurate trajectory tracking with improved robustness.

4.2.1 Longitude Control

The following differential equation can describe the x-position of the quadrotor UAV: The modified x-axis dynamics equation with the new control input relationship:

$$\begin{cases} \dot{x} = x_2 \\ \ddot{x} = \tau_x \frac{U_4}{m} + \Delta_x \end{cases} \tag{4.27}$$

The tracking error e_1 between the state x and desired state x_d is defined as:

$$e_1 = x - x_d \tag{4.28}$$

The sliding surface s_1 is defined as:

$$s_1 = \dot{e_1} + \beta_1 e_1, \qquad \beta_1 > 0 \tag{4.29}$$

Since,Equation (4.27) resembles the nonlinear system defined by Equation (4.1), thus the corresponding control input is

$$\tau_x = \tau_{x,l} + \tau_{x,n} \tag{4.30}$$

where $\tau_{x,l}$ is the equivalent control:

$$\tau_{x,l} = \frac{m}{U_1} \left(\ddot{x}_d - \beta_1 \dot{e}_1 - \mu_1 s_1 \right) \tag{4.31}$$

and $\tau_{x,n}$ is the switching control:

$$\tau_{x,n} = -\frac{m}{U_1} \left(k_1 \frac{s_1}{\|s_1\|} \right) \tag{4.32}$$

with $k_1 > \|\Delta_x\|$, where Δ_x represents the combined effects of system uncertainties and external disturbances in x-dynamics.

4.2.2 Lateral Control

The following differential equation can describe the y-position of the quadrotor UAV: The modified y-axis dynamics equation with the new control input relationship:

$$\begin{cases} \dot{y} = x_4 \\ \ddot{y} = \tau_y \frac{U_1}{m} + \Delta_y \end{cases} \tag{4.33}$$

The tracking error e_2 between the state y and desired state y_d is defined as:

$$e_2 = y - y_d \tag{4.34}$$

The sliding surface s_2 is defined as:

$$s_2 = \dot{e_2} + \beta_2 e_2, \qquad \beta_2 > 0 \tag{4.35}$$

Since, Equation (4.33) resembles the nonlinear system defined by Equation (4.1), thus the corresponding control input is

$$\tau_y = \tau_{y,l} + \tau_{y,n} \tag{4.36}$$

where $\tau_{y,l}$ is the equivalent control:

$$\tau_{y,l} = \frac{m}{U_1} \left(\ddot{y}_d - \beta_2 \dot{e}_2 - \mu_2 s_2 \right) \tag{4.37}$$

and $\tau_{y,n}$ is the switching control:

$$\tau_{y,n} = -\frac{m}{U_1} \left(k_2 \frac{s_2}{\|s_2\|} \right) \tag{4.38}$$

with $k_2 > \|\Delta_y\|$, where Δ_y represents the combined effects of system uncertainties and external disturbances in y-dynamics.

4.2.3 Altitude Control

The z-dynamics of quadrotor from Equation (4.19) are:

$$\begin{cases} \dot{x}_5 = x_6 \\ \ddot{x}_5 = -g + (\cos x_7 \cos x_9) \frac{U_1}{m} + \Delta_z \end{cases} \tag{4.39}$$

The altitude error e_3 between the state z and desired state z_d is defined as:

$$e_3 = z - z_d \tag{4.40}$$

Which is equal to:

$$e_3 = x_5 - z_d \tag{4.41}$$

The sliding surface s_3 is defined as:

$$s_3 = \dot{e}_3 + \beta_3 e_3, \quad \beta_3 > 0 \tag{4.42}$$

Since Equation (4.39) resembles the nonlinear system defined by Equation (4.1). Thus, the corresponding control law for total thrust (U_1) is as follows:

$$U_1 = U_{1l} + U_{1n} \tag{4.43}$$

Where U_{1l} is the equivalent or nominal control of z-dynamics, which is defined as:

$$U_{1l} = \frac{m}{\cos\phi \cos\theta} \left(g + \ddot{z}_d - \beta_3 \dot{e}_3\right) \tag{4.44}$$

While U_{1n} is the disturbance compensation control for altitude dynamics, which is given by the following expression:

$$U_{1n} = \frac{m}{\cos\phi \cos\theta} \left(-k_3 \text{sat}\left(\frac{s_3}{\epsilon}\right)\right) \tag{4.45}$$

Combining both nominal and disturbance compensation law, we get the complete Control law for total thrust (U_1), which is as follows:

$$U_1 = \frac{m}{\cos x_7 \cos x_9} \left(g + \ddot{z}_d - \beta_3 \dot{e}_3 - k_3 \text{sat} \left(\frac{s_3}{\epsilon} \right) \right) \tag{4.46}$$

with $k_3 > \|\Delta_z\|$, where Δ_z represents the combined effects of system uncertainties and external disturbances in z-dynamics.

4.2.4 Roll Control

From Equation (4.20) the roll dynamics are as following:

$$\begin{cases} \dot{x}_7 = x_8 \\ \ddot{x}_7 = x_{10} x_{12} \left(\frac{I_y - I_z}{I_x} \right) - \frac{I_r}{I_x} x_{10} \Omega + \frac{U_2}{I_x} + \Delta_p hi \end{cases} \tag{4.47}$$

The roll error e_4 between the state ϕ and desired state ϕ_d is defined as:

$$e_4 = \phi - \phi_d \tag{4.48}$$

Which is equal to

$$e_4 = x_7 - \phi_d \tag{4.49}$$

The sliding surface s_4 is defined as:

$$s_4 = \dot{e}_4 + \beta_4 e_4, \quad \beta_4 > 0 \tag{4.50}$$

Since, Equation (4.47) resembles the nonlinear system defined by Equation (4.1), thus the corresponding Control Law for roll control (U_2) is as following:

$$U_2 = U_{2l} + U_{2n} \tag{4.51}$$

Where U_{2l} is the equivalent or nominal control of roll-dynamics, which is defined as:

$$U_{2l} = I_x \left(\ddot{\phi}_d - x_{10}x_{12} \left(\frac{I_y - I_z}{I_x} \right) + \frac{I_r}{I_x} x_{10}\Omega - \beta_4 (x_8 - \dot{\phi}_d) \right) \tag{4.52}$$

While U_{2n} is the disturbance compensation control for roll dynamics, which is given by the following expression:

$$U_{2n} = -I_x \left(k_4 \text{sat} \left(\frac{s_4}{\epsilon} \right) \right) \tag{4.53}$$

Combining both nominal and disturbance compensation law, we get the complete control law for roll dynamics (U_2), which is as follows:

$$U_2 = I_x \left(\ddot{\phi}_d - x_{10}x_{12} \left(\frac{I_y - I_z}{I_x} \right) + \frac{I_r}{I_x} x_{10}\Omega - \beta_4 (x_8 - \dot{\phi}_d) - k_4 \text{sat} \left(\frac{s_4}{\epsilon} \right) \right) \tag{4.54}$$

with $k_4 > \|\Delta_\phi\|$, where Δ_ϕ represents the combined effects of system uncertainties and external disturbances in roll dynamics.

4.2.5 Pitch Control

From Equation (4.21) the pitch dynamics are as following:

$$\begin{cases} \dot{x}_9 = x_{10} \\ \ddot{x}_9 = x_8 x_{12} \left(\frac{I_z - I_x}{I_y} \right) - \frac{I_r}{I_y} x_8 \Omega + \frac{U_3}{I_y} + \Delta_\theta \end{cases} \tag{4.55}$$

The pitch error e_5 between the state θ and desired state θ_d is defined as:

$$e_5 = \theta - \theta_d \tag{4.56}$$

Which is equal to

$$e_5 = x_9 - \theta_d \tag{4.57}$$

The sliding surface s_5 is defined as:

$$s_5 = \dot{e}_5 + \beta_5 e_5, \quad \beta_5 > 0 \tag{4.58}$$

Since, Equation (4.55) resembles the nonlinear system defined by Equation (4.1), thus the corresponding control law for pitch control (U_3) is as following:

$$U_3 = U_{3l} + U_{3n} \tag{4.59}$$

Where U_{3l} is the equivalent or nominal control of pitch dynamics, which is defined as:

$$U_{3l} = I_y \left(\ddot{\theta}_d - x_8 x_{12} \left(\frac{I_z - I_x}{I_y} \right) + \frac{I_r}{I_y} x_8 \Omega - \beta_5 (x_{10} - \dot{\theta}_d) \right) \tag{4.60}$$

While U_{3n} is the disturbance compensation control for pitch dynamics, which is given by the following expression:

$$U_{3n} = -I_y k_5 \text{sat} \left(\frac{s_5}{\epsilon} \right) \tag{4.61}$$

Combining both nominal and disturbance compensation law, we get the complete Control Law for pitch dynamics (U_3), which is as follows:

$$U_3 = I_y \left(\ddot{\theta}_d - x_8 x_{12} \left(\frac{I_z - I_x}{I_y} \right) + \frac{I_r}{I_y} x_8 \Omega - \beta_5 (x_{10} - \dot{\theta}_d) - k_5 \text{sat} \left(\frac{s_5}{\epsilon} \right) \right) \tag{4.62}$$

with $k_5 > \|\Delta_\theta\|$, where Δ_θ represents the combined effects of system uncertainties and external disturbances in pitch dynamics.

4.2.6 Yaw Control

From Equation (4.22) the yaw dynamics are as following:

$$\begin{cases} \dot{x}_{11} = x_{12} \\ \ddot{x}_{11} = x_8 x_{10} \left(\frac{I_x - I_y}{I_z} \right) + \frac{U_4}{I_z} + \Delta_\psi \end{cases} \tag{4.63}$$

The yaw error e_6 between the state ψ and desired state ψ_d is defined as:

$$e_6 = \psi - \psi_d \tag{4.64}$$

Which is equal to:

$$e_6 = x_{11} - \psi_d \tag{4.65}$$

The sliding surface s_6 is defined as:

$$s_6 = \dot{e}_6 + \beta_6 e_6, \quad \beta_6 > 0 \tag{4.66}$$

Since, Equation (4.63) resembles the nonlinear system defined by Equation (4.1), thus the corresponding control law for yaw control (U_4) is as following:

$$U_4 = U_{4l} + U_{4n} \tag{4.67}$$

Where U_{4l} is the equivalent or nominal control of yaw-dynamics, which is defined as:

$$U_{4l} = I_z \left(\ddot{\psi}_d - x_8 x_{10} \left(\frac{I_x - I_y}{I_z} \right) - \beta_6 (x_{12} - \dot{\psi}_d) \right) \tag{4.68}$$

While U_{4n} is the disturbance compensation control for yaw dynamics, which is given by the following expression:

$$U_{4n} = -I_z \left(k_6 \text{sat} \left(\frac{s_6}{\epsilon} \right) \right) \tag{4.69}$$

Combining both nominal and disturbance compensation law, we get the complete control law for yaw dynamics (U_4), which is as follows:

$$U_4 = I_z \left(\ddot{\psi}_d - x_8 x_{10} \left(\frac{I_x - I_y}{I_z} \right) - \beta_6 (x_{12} - \dot{\psi}_d) - k_6 \text{sat} \left(\frac{s_6}{\epsilon} \right) \right) \tag{4.70}$$

with $k_6 > \|\Delta_\psi\|$, where Δ_ψ represents the combined effects of system uncertainties and external disturbances in yaw dynamics.

4.2.7 Numerical Simulations

Once the sliding mode control is designed, the closed-loop system is simulated. Simulations are performed in Simulink/MATLAB. The ODE4 solver with a fixed step time of 0.0001s is used. The quadrotor UAV is assumed to be at rest at the origin in the NED frame at $t = 0$. In addition, the orientation of the quadrotor UAV is assumed to be $\phi = \theta = \psi = 0$ in the body frame. The system parameters used for the simulation are summarized in Table 4.1, which are consistent with [1].

Table 4.1: Parameters of the quadrotor

Parameters	**Values**	**Explanation**
$I_x\,(kg \cdot m^2)$	7.5×10^{-3}	Inertia of X-axis
$I_y\,(kg \cdot m^2)$	7.5×10^{-3}	Inertia of Y-axis
$I_z\,(kg \cdot m^2)$	1.3×10^{-2}	Inertia of Z-axis
$I_r\,(kg \cdot m^2)$	6×10^{-5}	Inertia of Rotor
$b\,(N \cdot s^2)$	3.13×10^{-5}	Coefficient of Thrust
$d\,(m \cdot s^2)$	7.15×10^{-7}	Drag Coefficient
$m\,(kg)$	1.1	Mass
$l\,(m)$	0.26	Arm length

The design parameters for SMC are presented in Table 4.2

Table 4.2: Design Parameters for SMC

Parameter	**Value**	**Parameter**	**Value**
β_1	2.5	k_1	2
β_2	2.5	k_2	2
β_3	6.5	k_3	10
β_4	5.5	k_4	363
β_5	5.5	k_5	363
β_6	4.5	k_6	212
ε_0	0.003		

4.2.8 Simulation Results of SMC

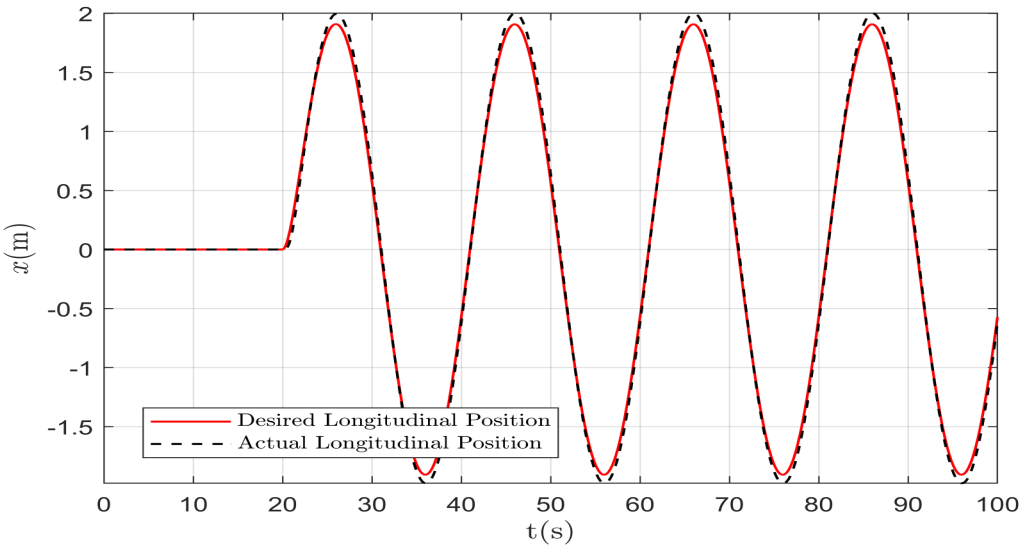

Figure 4.4: Longitudinal Position Control Using SMC

The graph in Figure 4.4 shows the performance of SMC for longitudinal position tracking for a quadrotor UAV over a 100-second time frame window. The solid red line represents the desired longitudinal position, while the dotted black line represents the actual position achieved by the quadrotor UAV. The UAV follows the desired longitudinal trajectory with high accuracy, indicating that the SMC implemented here can track arbitrary paths. However, the actual trajectory exhibits a slight overshoot at the positive and negative peaks of the sinusoidal input, with a slightly larger amplitude than desired and a consistent phase lag throughout. This overshoot represents the aggressive control action. It is due to large value of k_1 as the requirement is $k_1 > \|\Delta_x\|$, to compensate the disturbances. Overall, the system remains stable, with no divergence, oscillation buildup, or drift observed.

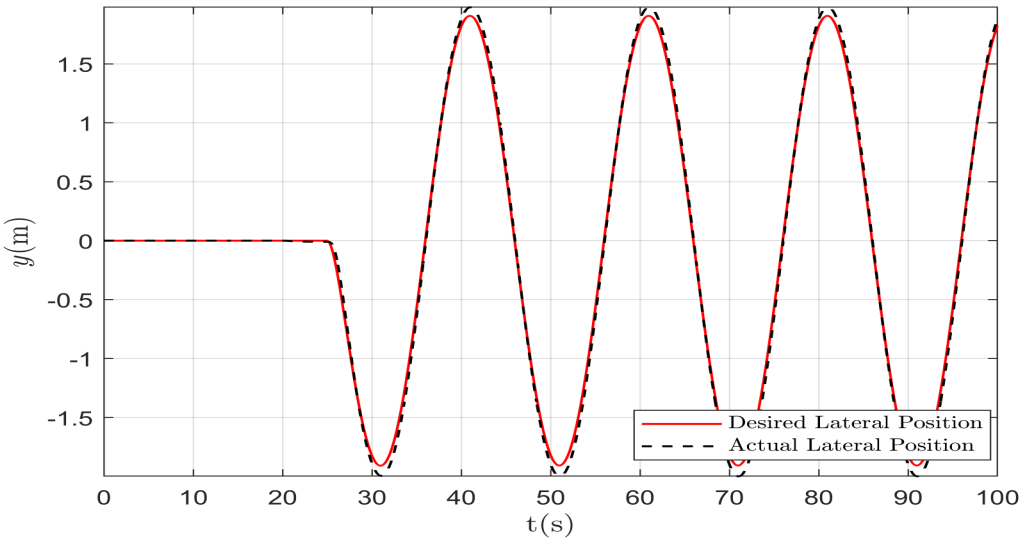

Figure 4.5: Lateral Position Control Using SMC

The graph in Figure 4.5 shows the performance of SMC for lateral position tracking of a quadrotor UAV over a 100-second time frame window. The solid red line represents the desired lateral position, while the dotted black line represents the actual position achieved by the quadrotor. The UAV accurately follows the desired lateral trajectory, indicating that the SMC implemented here can track arbitrary paths. However, the actual trajectory exhibits a slight overshoot at the positive and negative peaks of the sinusoidal input, with a slightly larger amplitude than desired and a consistent phase lag throughout. This represents the aggressive control action. It is due to large value of k_2 as the requirement is $k_2 > \|\Delta_y\|$, to compensate the disturbances. Overall, the system remains stable, with no divergence, oscillation buildup, or drift observed.

The graph in Figure 4.6 shows the performance of SMC for altitude tracking of a quadrotor UAV over a 100-second time frame window. The solid red line represents the desired altitude trajectory, while the dotted black line represents the actual altitude attained by the quadrotor. The actual altitude trajectory perfectly overlaps with the desired trajectory throughout the simulation. No delay, overshoot, oscillation buildup, or drift is observed at any point in the simulation.

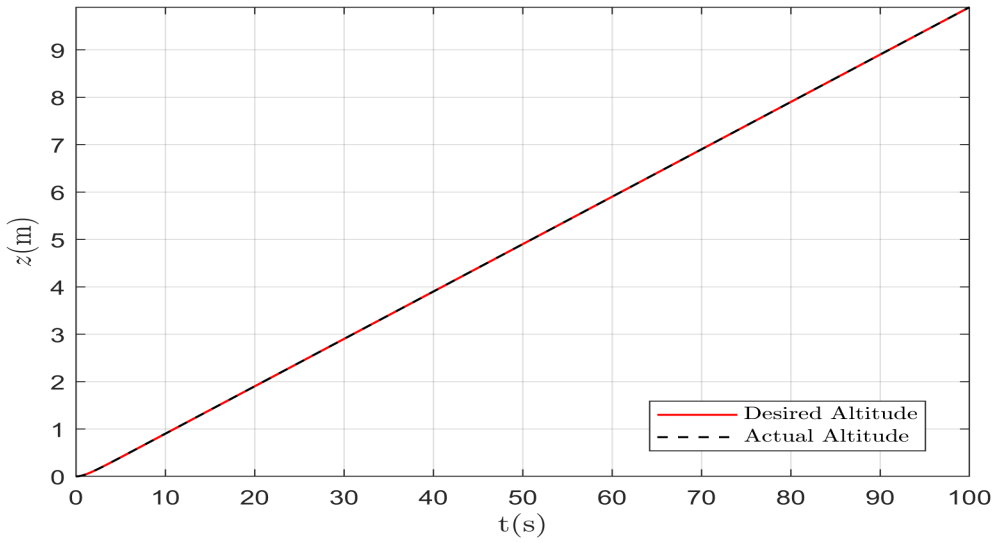

Figure 4.6: Altitude Control Using SMC

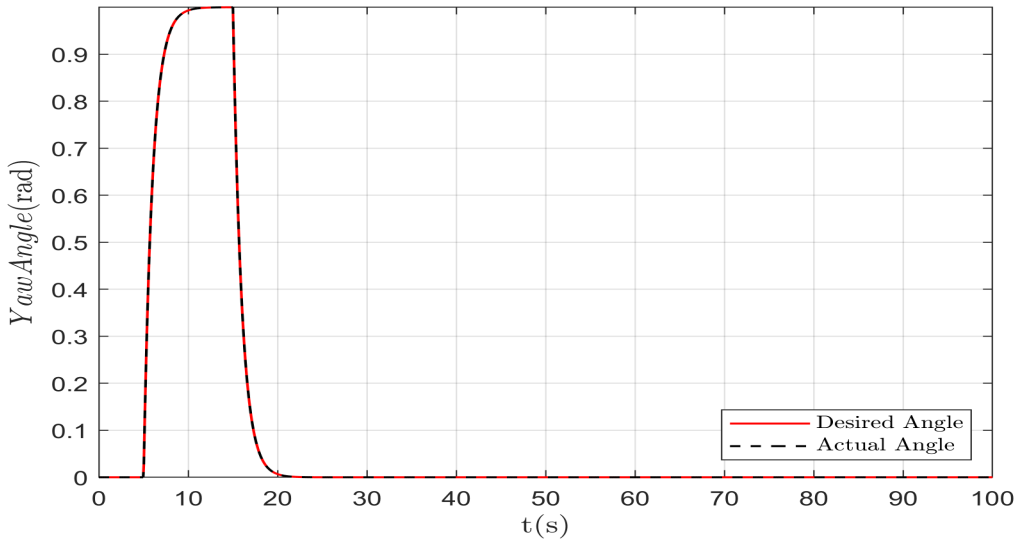

Figure 4.7: Yaw Angle Control Using SMC

The plot in Figure 4.7 represents the performance of yaw angle reference tracking for a quadrotor UAV. The solid red line represents the desired yaw angle, while the dotted black line represents the actual yaw angle attained by the quadrotor. The yaw angle is tracked perfectly by SMC. No delay, overshoot, oscillation buildup, or drift is observed at any point in the simulation.

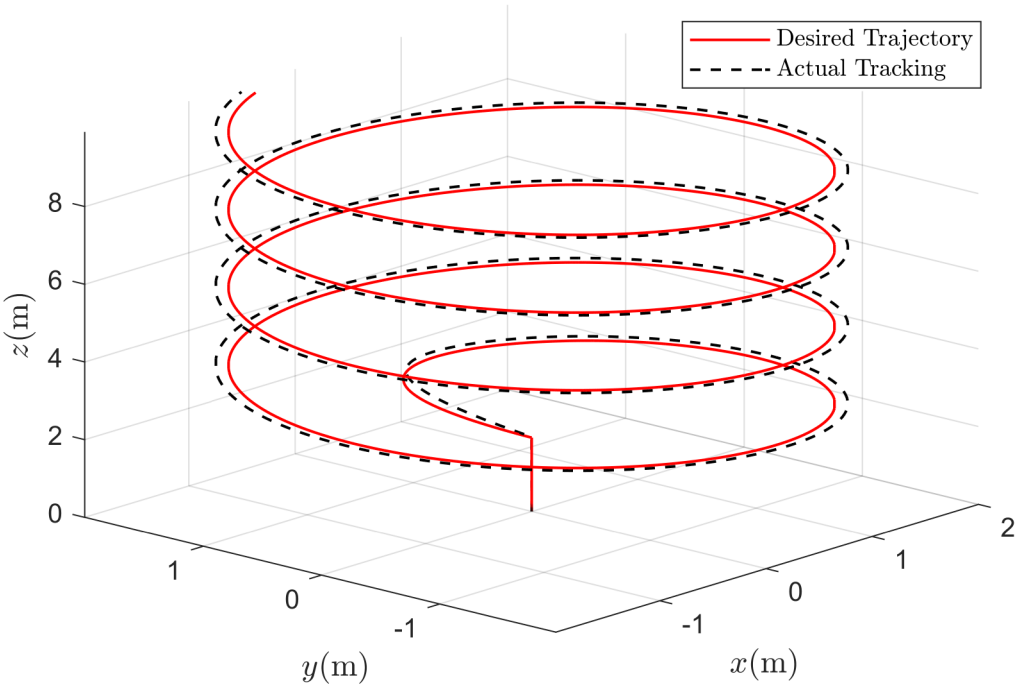

Figure 4.8: 3D Maneuvering of Quadrotor UAV using SMC

The 3D trajectory plot in Figure 4.8 depicts the performance of quadrotor UAV trajectory tracking in helical/spiral path in the x,y,z (3D) space. The solid red line shows the desired trajectory, while the dotted black line illustrates the actual trajectory followed by the UAV. From the plot in Figure 4.8, it is evident that the quadrotor has followed the desired 3D trajectory with minor and bounded error. Moreover, there exists a minor lag and overshoot. This overshoot is due to the controller's aggressive behavior, driven by high fixed gains. These high fixed gains are unavoidable because the controller must overcome external disturbances and uncertainties.

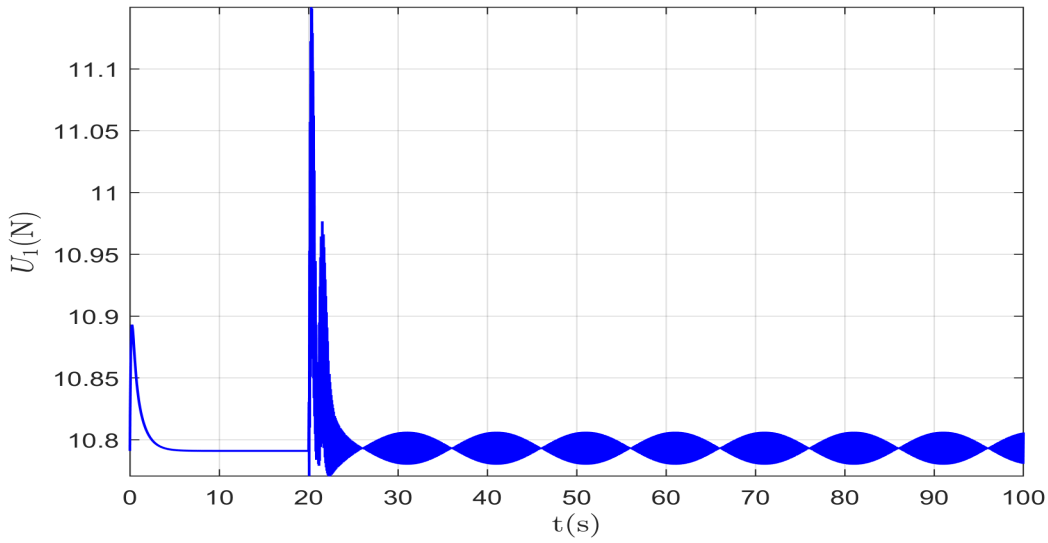

Figure 4.9: SMC Control Input: U_1

Plot in Figure 4.9 represents the control input U_1, which is the total trust generated by the rotors to counteract gravity and control vertical motion (altitude). As shown in Figure, initial spike of approximately 10.9N is required to counteract gravity ($m = 1.1\,\text{kg}$ x $g = 9.81\,\text{m/s}^2 = 10.79\text{N}$) and initiate vertical motion. In the current scenario, the quadrotor is not only ascending vertically but also moving along a circular x-y path, resulting in an overall 3D helical trajectory. Furthermore, a quadrotor is a coupled system, exhibiting a strong coupling between the x, y, roll, and pitch states. At time $t = 20\,\text{sec}$ the desired longitudinal position $x_d(t)$ raises from 0 to a positive value as depicted in Figure 4.4, while, lateral position $y_d(t)$ drops from 0 to a negative value at the same time as can be seen in Figure 4.5. To generate the motions in x & y directions, the quadrotor tilts. As a result of this tilt, the thrust vector is no longer purely vertical. Therefore, more thrust is required to maintain upward acceleration. Once this phase is over, the control input U_1 settles into a periodic oscillation pattern around the nominal thrust level (approximately $10.8\,\text{Nm}$).

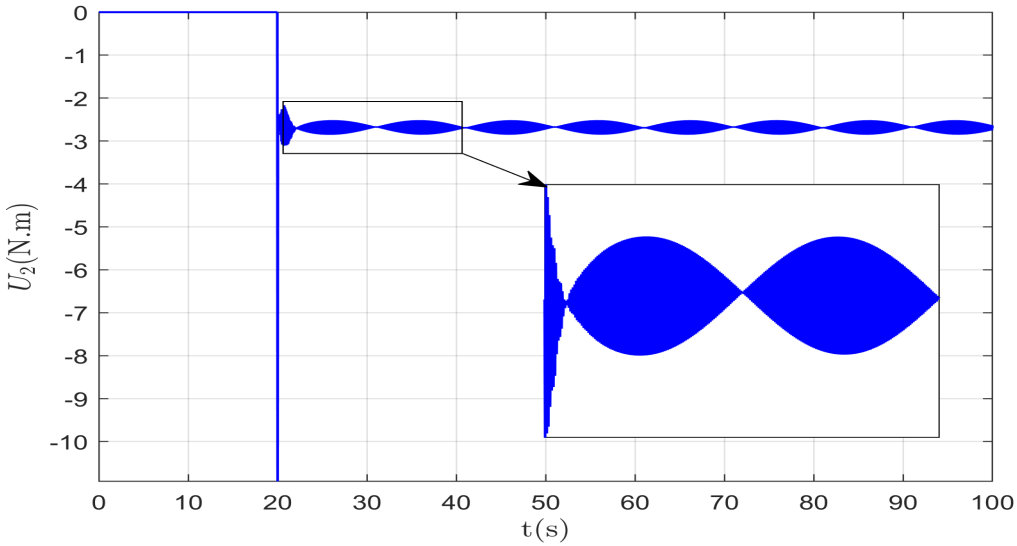

Figure 4.10: SMC Control Input: U_2

Figure 4.10 shows the evolution of control effort U_2 (N.m) with respect to time t (in seconds) for a quadrotor UAV. In a quadrotor UAV U_2, typically represents rolling torque, which adjusts its orientation about its longitudinal axis. Plot depicts that from $t = 0$ to $t \approx 20$ seconds, the control input U_2 remains 0. It implies that no corrective roll torque is required around the longitudinal axis, as cross verifies from Figure 4.4. At $t = 20$ seconds, there is a sharp negative spike in U_2. This indicates the need for immediate torque correction, as the desired longitudinal position rapidly changes at that instant. From $t > 20$ to $t = 100$ seconds, damped oscillations can be observed, settling into a more periodic manner. These oscillations indicate the tracking of a sinusoidal trajectory. The rapid damping indicates effective control and stabilization. The zoomed view shows periodic oscillations. The smooth, regular waveform indicates a stable controller response. The waveform's constant amplitude over time in this region suggests that the system is neither drifting nor diverging.

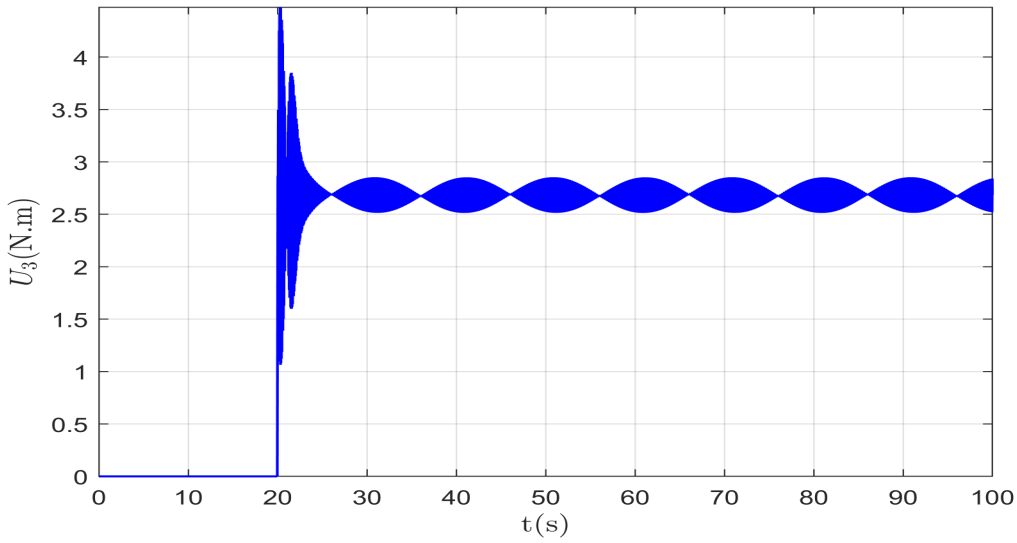

Figure 4.11: SMC Control Input: U_3

Figure 4.11 shows the evolution of control effort U_3 (N.m) with respect to time t (in seconds) for a quadrotor UAV. In a quadrotor UAV U_3, typically represents pitch torque, which adjusts its orientation about its lateral axis. Plot depicts that from $t = 0$ to $t \approx 20$ seconds, the control input U_2 remains 0. It implies that no corrective pitch torque is required around the lateral axis, as cross-verification from Figure 4.5. At $t = 20$ seconds, there is a sharp positive spike in U_3. This indicates the need for immediate torque correction, as the desired lateral position rapidly changes at that instant. From $t > 20$ to $t = 100$ seconds, damped oscillations can be observed, settling into a more periodic manner. These oscillations indicate the tracking of a sinusoidal trajectory. The rapid damping indicates effective control and stabilization.

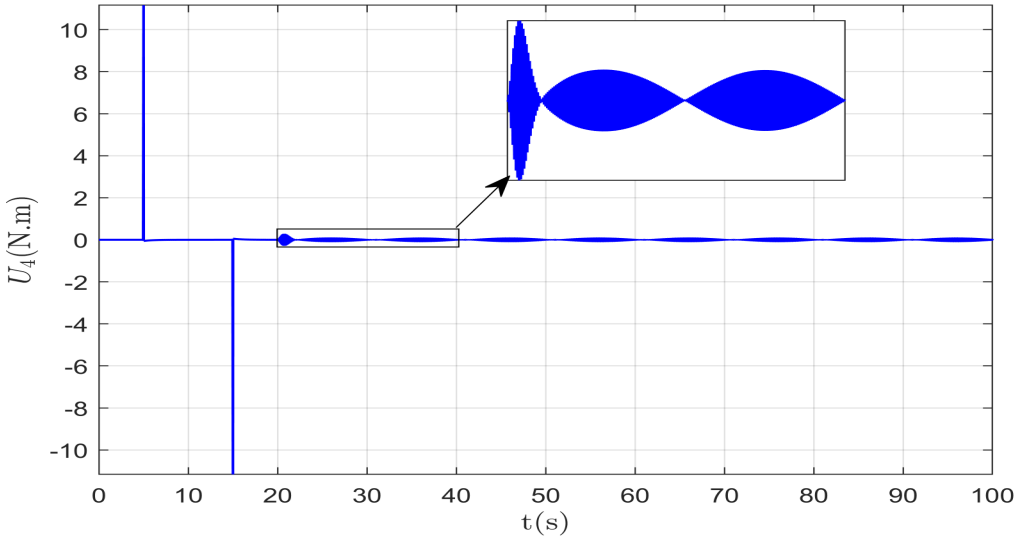

Figure 4.12: SMC Control Input: U_4

Figure 4.12 shows the evolution of control effort U_4 (N.m) with respect to time t (in seconds) for a quadrotor UAV. In the quadrotor UAV U_4, responsible for yaw control, which adjusts its orientation about the vertical z axis. This affects the UAV's heading. As shown in the Figure, the initial two positive and negative spikes, each approximately 11 N $\cdot$ m in magnitude, within the first 20 seconds, represent a sudden change in the yaw setpoint. This rapid change in yaw angle control can be counterchecked from Figure 4.7. It can be seen that the desired yaw angle abruptly changes from 0 to 1 radian and back to 0 radian over the first 20 seconds. To meet the high yaw-angle target, the control input U_4 responds aggressively. Time $t \approx 20$ to $t = 40$ seconds, the yaw torque U_4 exhibits decaying sinusoidal oscillations for stabilizing. From time $t = 40\,\text{s}$ onward, the control input U_4 remains stable with small periodic oscillations centered around $0\,\text{N}\cdot\text{m}$. These small yaw corrections indicate that the UAV maintains a stable heading while exhibiting only minor periodic motions, possibly due to coupled roll or pitch motions.

4.3 Adaptive SMC (ASMC)

The preceding discussion focused on SMC, a well-known method for controlling systems in the presence of disturbances. However, a key challenge in designing an SMC controller is its dependence on prior knowledge of an upper bound γ on the lumped disturbance Δ. In real-world applications, accurately determining these upper bounds can be challenging. Especially, the flight conditions of a quadrotor UAV may vary significantly between indoor and outdoor environments, ranging from negligible to very high wind gusts. In such circumstances, if these bounds are estimated a priori, then their values are often too conservative, resulting in unnecessarily high control effort due to the high values of gain k, in equation 4.11.

To address this limitation, a more robust control strategy is required that does not rely on prior knowledge of uncertainty/disturbance bounds. For this purpose, we propose an ASMC approach that offers strong robustness by dynamically adjusting the controller gain in response to the system's behavior. This eliminates the need for predefined knowledge of uncertainty or disturbance. Instead, knowledge of the initial disturbance at $(t = 0)$ bound will be sufficient to establish the robustness of the controller for all time. Thus, the objective is to design an adaptive mechanism capable of counteracting system uncertainties and external disturbances in real time with knowledge of a bound on Δ at time $t = 0$, i.e., $\eta > \Delta(0)$, where η is a positive real constant.

Similar to SMC, the ASMC structure is also composed of two components: $u = u_l$ (equivalent or nominal control) + u_n (adaptive disturbance compensator). The first governs the nominal system behavior, while the second compensates for parameter variations and external perturbations. A block diagram of a typical ASMC scheme is shown in Figure 4.13 below:

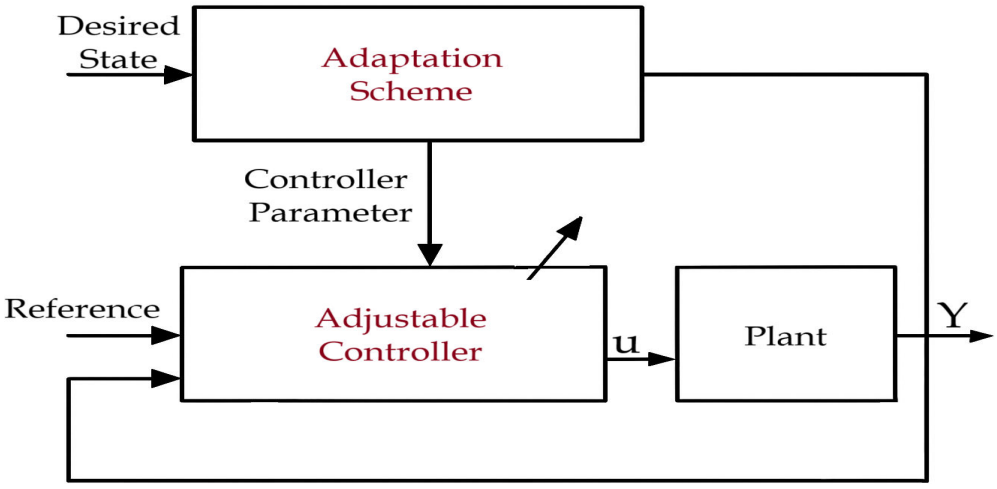

Figure 4.13: ASMC Scheme

4.3.1 ASMC Control Law Design

Consider the nonlinear system defined by Equation 4.1 with a less stringent condition of disturbance bound $\Delta(0) < \eta$. Moreover, let us define a self evolving parameter $\bar{k}$ such that,

$$\dot{\bar{k}}(t,x) = -\zeta_1 \bar{k}(t,x) + \zeta_2 \epsilon_0 \|u_l\| \|s\| \tag{4.71}$$

To design ASMC, we choose the same sliding surface 4.2. The Lyapunov candidate function considered here is

$$V = \frac{1}{2}s^2 + \frac{1}{2}\bar{k}^2 \tag{4.72}$$

which is, by construction, a positive definite function. Taking the derivative

$$\dot{V} = s\dot{s} + \bar{k}\dot{\bar{k}} \tag{4.73}$$

Substituting the expressions for $\dot{\bar{k}}$ and $\dot{s}$, we obtain:

$$\dot{V} = s(\ddot{e} + \beta\dot{e}) + \bar{k}_1 \left(-\zeta_1 \bar{k} + \zeta_2 \epsilon_0 \|u_l\| \|s\|\right) \tag{4.74}$$

By substituting the error, $e = x_1 - r$ into above equation we get:

$$\dot{V} = s(\ddot{x}_1 - \ddot{r} + \beta\dot{x}_1 - \beta\dot{r}) - \zeta_1\bar{k}_1^2 + \zeta_2\epsilon_0\bar{k}\|u_l\|\|s\| \tag{4.75}$$

Using the system dynamics 4.1

$$\dot{V} = s\left(f(x_1, x_2) + g(x_1, x_2)u + \Delta - \ddot{r} + \beta x_2 - \beta\dot{r}\right) - \zeta_1\bar{k}^2 + \zeta_2\epsilon_0\bar{k}\|u_l\|\|s\| \tag{4.76}$$

In ASMC, the control input is given by $u = u_l + u_n$, where u_l is the equivalent or nominal control input and u_n is the adaptive disturbance compensator.

$$u_l = -\frac{1}{g(x_1, x_2)}\left(f(x_1, x_2) - \ddot{r} + \beta x_2 - \beta\dot{r} + \mu_1 s\right) \tag{4.77}$$

$$u_n = -\frac{1}{g(x_1, x_2)}\left(k(t, x)\frac{s}{\|s\|}\right) \tag{4.78}$$

where, k_1 is adaptive gain that relates to η and $\bar{k}$ by following expression:

$$k(t, x) = \|u_l\|\bar{k}(t, x) + \eta \tag{4.79}$$

By substituting the u into the above equation, we get:

$$\begin{aligned}\dot{V} = s\Bigg(f(x_1, x_2) + g(x_1, x_2)\left[\frac{-1}{g(x_1, x_2)}\left(f(x_1, x_2) - \ddot{r} + \beta x_2 - \beta\dot{r} + \mu_1 s + k(t, x)\frac{s}{\|s\|}\right)\right] \\ + \Delta - \ddot{r} + \beta x_2 - \beta\dot{r}\Bigg) - \zeta_1\bar{k}^2 + \zeta_2\epsilon_0\bar{k}\|u_l\|\|s\|\end{aligned} \tag{4.80}$$

By simplifying the above expression, we get:

$$\dot{V} = s\left(\Delta - \mu_1 s - k(t, x)\frac{s}{\|s\|}\right) - \zeta_1\bar{k}^2 + \zeta_2\epsilon_0\bar{k}\|u_l\|\|s\| \tag{4.81}$$

$$\dot{V} = \Delta s - \mu_1 s^2 - k(t, x)\|s\| - \zeta_1\bar{k}^2 + \zeta_2\epsilon_0\bar{k}\|u_l\|\|s\| \tag{4.82}$$

This yields:

$$\dot{V} = \Delta s - \mu_1 s^2 - \left(\|u_l\|\bar{k}(t,x) + \eta\right)\|s\| - \zeta_1\bar{k}^2 + \zeta_2\epsilon_0\bar{k}\|u_l\|\|s\| \tag{4.83}$$

$$\dot{V} = \Delta s - \mu_1 s^2 - \bar{k}(t,x)\|u_l\|\|s\| - \eta\|s\| - \zeta_1\bar{k}^2 + \zeta_2\epsilon_0\bar{k}\|u_l\|\|s\| \tag{4.84}$$

By reducing the above expression, we get:

$$\dot{V} = -\mu_1 s^2 - \zeta_1\bar{k}^2 - (1-\zeta_2\epsilon_0)\bar{k}\|u_l\|\|s\| - \eta\|s\| + \Delta s \tag{4.85}$$

Taking $\|s\|$ common, the above expression take the following form:

$$\dot{V} = -\mu_1 s^2 - \zeta_1\bar{k}^2 - \left((1-\zeta_2\epsilon_0)\,\bar{k}\|u_l\| + \eta\right)\|s\| + \Delta s \tag{4.86}$$

Negative definiteness of $\dot{V}$ is ensured only if

$$\left((1-\zeta_2\epsilon_0)\bar{k}\|u_l\| + \eta\right) > \Delta,$$

and, it is known that $\eta > \Delta_x(0)$ and

$$\dot{\bar{k}}(t,x) = -\zeta_1\bar{k}(t,x) + \zeta_2\epsilon_0\|u_l\|\|s\|.$$

The block diagram of ASMC is shown in Figure 4.14. The reference trajectory specifies the desired positions and orientations, which are compared with the UAV's actual states to compute tracking errors. These errors are processed by the six ASMCs, comprising altitude, longitudinal, lateral, roll, pitch, and yaw controllers. The desired Euler angle calculation block produces the required roll and pitch commands. The adaptive law estimates the modulation gain, enabling the controller to handle disturbances and modeling uncertainties without relying on conservative fixed bounds. These controllers convert the desired angles into motor-control signals for the UAV, allowing it to execute the commanded motions despite external disturbances, such as wind gusts.

The UAV's actual states are continuously fed back to the controller, thereby closing the loop and ensuring accurate trajectory tracking with improved robustness.

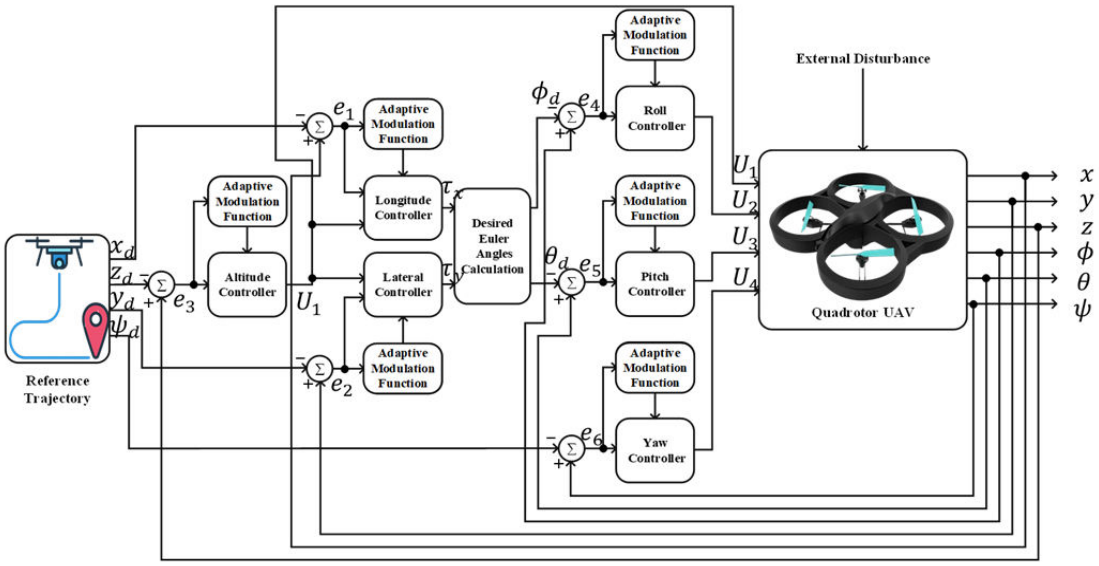

Figure 4.14: ASMC Block Diagram

4.4 Adaptive SMC for Quadrotor UAV

4.4.1 Longitude Control

Considering the same sliding surface s_1, as already defined by Equation 4.29, the longitudinal dynamics of the quadrotor UAV represented by Equation 4.27 resembles the nonlinear system model described by Equation 4.1. Hence, the control law for x-position control is the same as defined in equation 4.30, that is:

$$\tau_x = \tau_{x,l} + \tau_{x,n} \tag{4.87}$$

where $\tau_{x,l}$ is the equivalent control:

$$\tau_{x,l} = \frac{m}{U_1}\left(\ddot{x}_d - \beta_1 \dot{e}_1 - \mu_1 s_1\right) \tag{4.88}$$

and $\tau_{x,n}$ is the switching control:

$$\tau_{x,n} = -\frac{m}{U_1}\left(k_1 \frac{s_1}{\|s_1\|}\right) \tag{4.89}$$

Here, k_1 is the adaptive gain. k_1 adapts itself to compensate for the disturbance faced by the quadrotor, rather than a fixed value, as it is in SMC. The adaptive gain k_1 is defined by the following expression:

$$k_1(t,x) = \|u_l\|\bar{k}_1(t,x) + \eta \tag{4.90}$$

Such that,

$$\dot{\bar{k}}_1(t,x) = -\zeta_1 \bar{k}_1(t,x) + \zeta_2 \epsilon_0 \|u_l\|\|s_1\| \tag{4.91}$$

4.4.2 Lateral Control

Considering the same sliding surface s_2, as already defined by Equation 4.35, the longitudinal dynamics of the quadrotor UAV represented by Equation 4.33 resemble the nonlinear system model described by Equation 4.1. Hence, the control law for y-position control is the same as defined by Equation 4.36, that is:

$$\tau_y = \tau_{y,l} + \tau_{y,n} \tag{4.92}$$

where $\tau_{y,l}$ is the equivalent control:

$$\tau_{y,l} = \frac{m}{U_1}\left(\ddot{y}_d - \beta_2 \dot{e}_2 - \mu_2 s_2\right) \tag{4.93}$$

and $\tau_{y,n}$ is the switching control:

$$\tau_{y,n} = -\frac{m}{U_1}\left(k_2 \frac{s_2}{\|s_2\|}\right) \tag{4.94}$$

The adaptive gain k_2 is defined by the following expression:

$$k_2(t,x) = \|u_l\|\bar{k}_2(t,x) + \eta \tag{4.95}$$

Such that,

$$\dot{\bar{k}}_2(t,x) = -\zeta_1 \bar{k}_2(t,x) + \zeta_2 \epsilon_0 \|u_l\| \|s_2\| \tag{4.96}$$

4.4.3 Altitude Control

Considering the same sliding surface s_3, as already defined by Equation 4.42, altitude dynamics of quadrotor UAV represented by Equation 4.19 resembles the nonlinear system model represented by Equation 4.1. Thus, the corresponding control law for upward thrust (U_1) is as follows:

$$U_1 = U_{1l} + U_{1n} \tag{4.97}$$

Where U_{1l} is the equivalent or nominal control of z-dynamics, which is defined as:

$$U_{1l} = \frac{m}{\cos\phi \cos\theta} \left(g + \ddot{z}_d - \beta_3 \dot{e}_3\right) \tag{4.98}$$

While U_{1n} is the disturbance compensation control for altitude dynamics, which is given by the following expression:

$$U_{1n} = \frac{m}{\cos\phi \cos\theta} \left(-k_3 \text{sat}\left(\frac{s_3}{\epsilon}\right)\right) \tag{4.99}$$

Combining both nominal and disturbance compensation law, we get the complete control law for total thrust (U_1), which is as follows:

$$U_1 = \frac{m}{\cos x_7 \cos x_9} \left(g + \ddot{z}_d - \beta_3 \dot{e}_3 - k_3 \text{sat}\left(\frac{s_3}{\epsilon}\right)\right) \tag{4.100}$$

Where, the adaptive gain k_3 is defined by following expression:

$$k_3(t,x) = \|u_{1l}\| \bar{k}_3(t,x) + \eta \tag{4.101}$$

Such that,

$$\dot{\bar{k}}_3(t,x) = -\zeta_1 \bar{k}_3(t,x) + \zeta_2 \epsilon_0 \|u_{1l}\| \|s_3\| \tag{4.102}$$

4.4.4 Roll Control

Considering the same sliding surface s_4, as already defined by Equation 4.50, the roll dynamics of the quadrotor UAV represented by Equation 4.47 resembles the nonlinear system model described by Equation 4.1. Thus, the corresponding control law for roll dynamics (U_2) is as follows:

$$U_2 = U_{2l} + U_{2n} \tag{4.103}$$

Where U_{2l} is the equivalent or nominal control of roll-dynamics, which is defined as:

$$U_{2l} = I_x \left(\ddot{\phi}_d - x_{10} x_{12} \left(\frac{I_y - I_z}{I_x} \right) + \frac{I_r}{I_x} x_{10} \Omega - \beta_4 (x_8 - \dot{\phi}_d) \right) \tag{4.104}$$

While U_{2n} is the disturbance compensation control for roll dynamics, which is given by the following expression:

$$U_{2n} = -I_x \left(k_4 \text{sat} \left(\frac{s_4}{\epsilon} \right) \right) \tag{4.105}$$

Combining both nominal and disturbance compensation law, we get the complete control law for roll dynamics (U_2), which is as follows:

$$U_2 = I_x \left(\ddot{\phi}_d - x_{10} x_{12} \left(\frac{I_y - I_z}{I_x} \right) + \frac{I_r}{I_x} x_{10} \Omega - \beta_4 (x_8 - \dot{\phi}_d) - k_4 \text{sat} \left(\frac{s_4}{\epsilon} \right) \right) \tag{4.106}$$

Where, the adaptive gain k_4 is defined by following expression:

$$k_4(t, x) = \|u_{2l}\| \bar{k}_4(t, x) + \eta \tag{4.107}$$

Such that,

$$\dot{\bar{k}}_4(t, x) = -\zeta_1 \bar{k}_4(t, x) + \zeta_2 \epsilon_0 \|u_{2l}\| \|s_4\| \tag{4.108}$$

4.4.5 Pitch Control

Considering the same sliding surface s_5, as already defined by Equation 4.58, the pitch dynamics of the quadrotor UAV represented by Equation 4.55 resemble the nonlinear

system model described by Equation 4.1. Thus, the corresponding control law for pitch dynamics (U_3) is as follows:

$$U_3 = U_{3l} + U_{3n} \tag{4.109}$$

Where U_{3l} is the equivalent or nominal control of pitch-dynamics, which is defined as:

$$U_{3l} = I_y \left(\ddot{\theta}_d - x_8 x_{12} \left(\frac{I_z - I_x}{I_y} \right) + \frac{I_r}{I_y} x_8 \Omega - \beta_5 (x_{10} - \dot{\theta}_d) \right) \tag{4.110}$$

While U_{3n} is the disturbance compensation control for pitch dynamics, which is given by the following expression:

$$U_{3n} = -I_y k_5 \text{sat} \left(\frac{s_5}{\epsilon} \right) \tag{4.111}$$

Combining both nominal and disturbance compensation law, we get the complete control law for pitch dynamics (U_3), which is as follows:

$$U_3 = I_y \left(\ddot{\theta}_d - x_8 x_{12} \left(\frac{I_z - I_x}{I_y} \right) + \frac{I_r}{I_y} x_8 \Omega - \beta_5 (x_{10} - \dot{\theta}_d) - k_5 \text{sat} \left(\frac{s_5}{\epsilon} \right) \right) \tag{4.112}$$

Where, the adaptive gain k_5 is defined by following expression:

$$k_5(t, x) = \|u_{3l}\| \bar{k}_5(t, x) + \eta \tag{4.113}$$

Such that,

$$\dot{\bar{k}}_5(t, x) = -\zeta_1 \bar{k}_5(t, x) + \zeta_2 \epsilon_0 \|u_{3l}\| \|s_5\| \tag{4.114}$$

4.4.6 Yaw Control

Considering the same sliding surface s_6, as already defined by Equation 4.66, the yaw dynamics of the quadrotor UAV represented by Equation 4.63 resemble the nonlinear system model represented by Equation 4.1. Thus the corresponding control law for

yaw dynamics (U_4) is as following:

$$U_4 = U_{4l} + U_{4n} \tag{4.115}$$

Where U_{4l} is the equivalent or nominal control of yaw-dynamics, which is defined as:

$$U_{4l} = I_z \left(\ddot{\psi}_d - x_8 x_{10} \left(\frac{I_x - I_y}{I_z} \right) - \beta_6 (x_{12} - \dot{\psi}_d) \right) \tag{4.116}$$

While U_{4n} is the disturbance compensation control for yaw dynamics, which is given by the following expression:

$$U_{4n} = -I_z \left(k_6 \text{sat} \left(\frac{s_6}{\epsilon} \right) \right) \tag{4.117}$$

Combining both nominal and disturbance compensation law, we get the complete control law for yaw dynamics (U_4), which is as follows:

$$U_4 = I_z \left(\ddot{\psi}_d - x_8 x_{10} \left(\frac{I_x - I_y}{I_z} \right) - \beta_6 (x_{12} - \dot{\psi}_d) - k_6 \text{sat} \left(\frac{s_6}{\epsilon} \right) \right) \tag{4.118}$$

Where, the adaptive gain k_6 is defined by following expression:

$$k_6(t, x) = \|u_{4l}\| \bar{k}_6(t, x) + \eta \tag{4.119}$$

Such that,

$$\dot{\bar{k}}_6(t, x) = -\zeta_1 \bar{k}_6(t, x) + \zeta_2 \epsilon_0 \|u_{4l}\| |s_6| \tag{4.120}$$

4.4.7 Numerical Simulations

Similar to SMC, the ASMC performance is also evaluated in Simulink/Matlab. The same system parameters, initial conditions, and simulation settings are used. The design parameters for ASMC are presented in Table 4.3

The graph in Figure 4.15 shows the performance of ASMC for longitudinal position control of quadrotor UAV over a 100-second time frame window. The solid red line represents the desired longitudinal position, while the dotted black line represents

Table 4.3: Design Parameters for ASMC

Parameter	Value	Parameter	Value
β_1	2.5	μ_3	0
β_2	2.5	μ_4	0.0025
β_3	6.5	μ_5	0.0025
β_4	5.5	μ_6	0.00025
β_5	5.5	ζ_1	200
β_6	4.5	ζ_2	100
μ_1	8	η	0.9
μ_2	0	ε_0	0.003

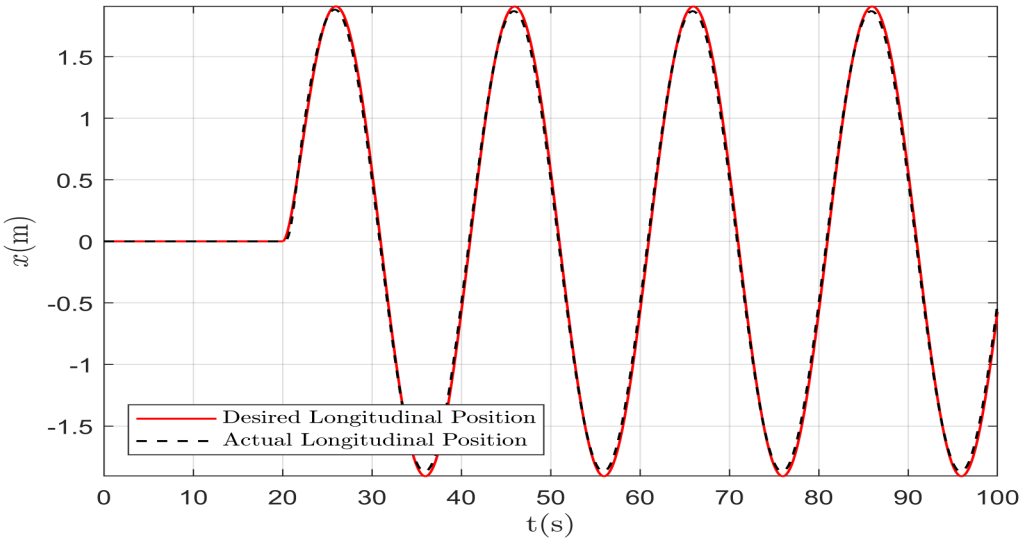

Figure 4.15: Longitudinal Position Control Using ASMC

the actual longitudinal position achieved by the quadrotor. As depicted by the plot, the error between the actual and desired trajectories is minimal. The actual position trajectory is almost identical to the desired position throughout the entire time frame, indicating the controller's high accuracy and robustness. Furthermore, no phase lag is found, and overshoot is nearly zero. Overall, the system remains stable, no divergence, oscillation buildup, or drift is observed.

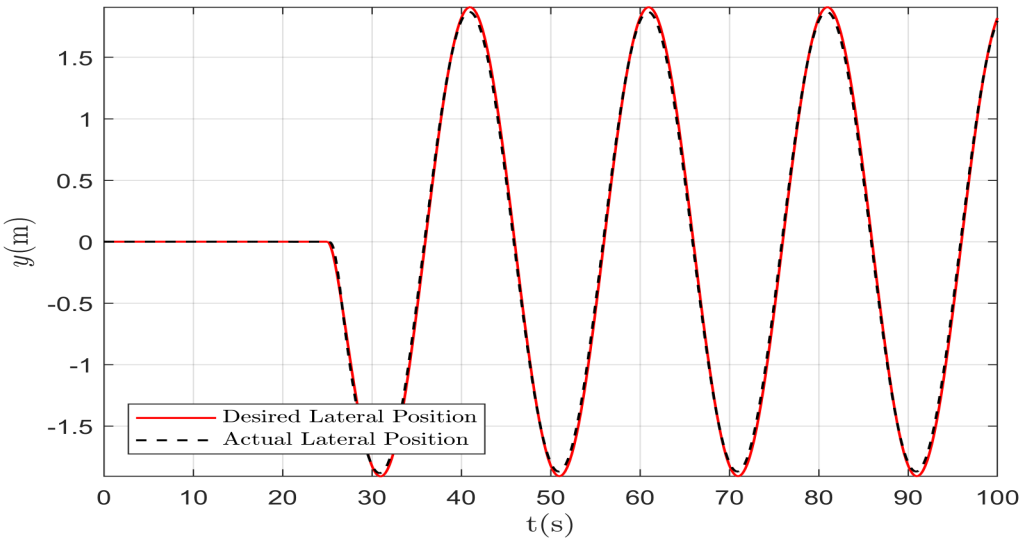

Figure 4.16: Lateral Position Control Using ASMC

The graph in Figure 4.16 shows the performance of ASMC for lateral position control of quadrotor UAV over a 100-second time frame window. The solid red line represents the desired lateral position, while the dotted black line represents the actual lateral position achieved by the quadrotor. As depicted by the plot, the error between the actual and desired trajectories is minimal. The actual position trajectory is almost identical to the desired position throughout the entire time frame, indicating the controller's high accuracy and robustness. Furthermore, no phase lag is found, and overshoot is nearly zero. Overall, the system remains stable, no divergence, oscillation buildup, or drift is observed.

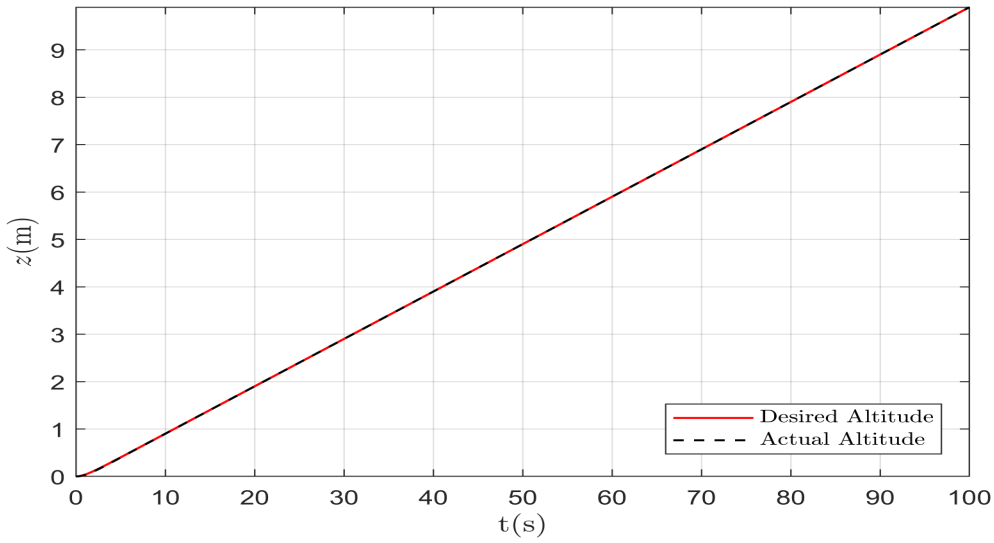

Figure 4.17: Altitude Control Using ASMC

The plot in Figure 4.17 represents the performance of altitude control for a quadrotor UAV. The solid red line represents the desired altitude, while the dotted black line represents the actual altitude attained by the quadrotor. ASMC achieves the desired altitude perfectly. No delay, overshoot, oscillation buildup, or drift is observed at any point in the simulation.

The plot in Figure 4.18 represents the performance of yaw angle control for a quadrotor UAV. The solid red line represents the desired yaw angle, while the dotted black line represents the actual yaw angle attained by the quadrotor. The yaw angle is traced perfectly by ASMC. No delay, overshoot, oscillation buildup, or drift is observed at any point in the simulation.

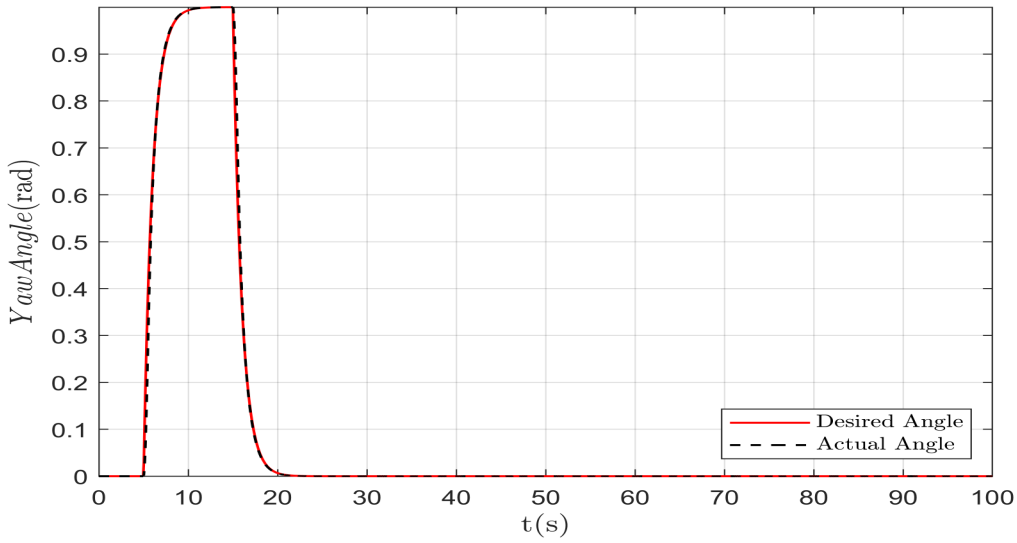

Figure 4.18: Yaw Angle Control Using ASMC

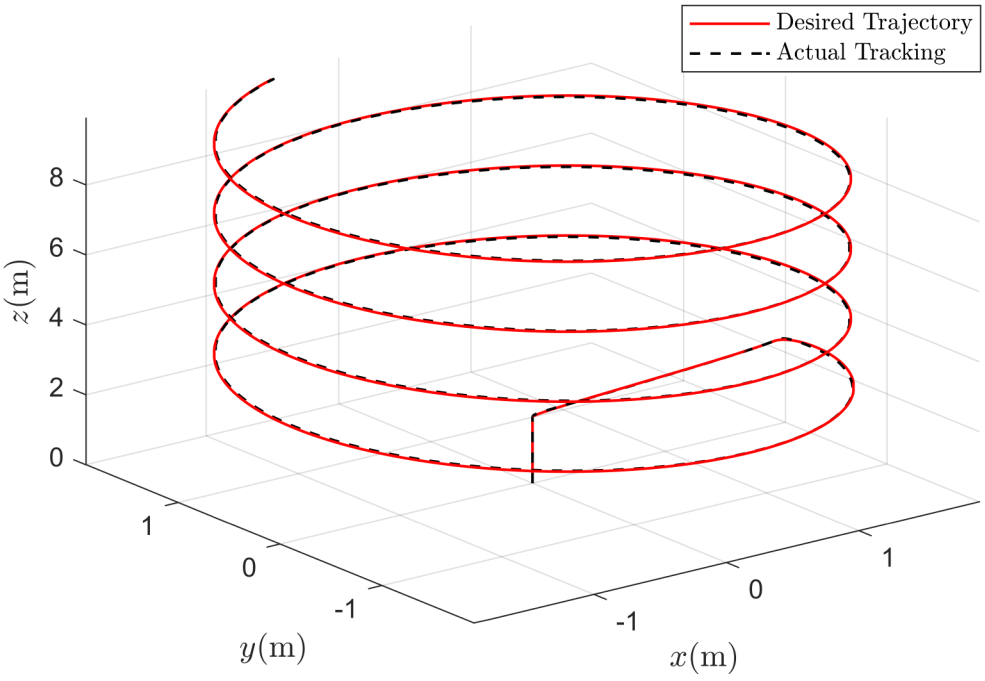

Figure 4.19: 3D Maneuvering of Quadrotor UAV using ASMC

The 3D trajectory plot in Figure 4.19 depicts the performance of quadrotor UAV tracking in a helical/spiral path in the x,y,z (3D) space. The solid red line shows the

desired trajectory, while the dotted black line illustrates the actual trajectory followed by the UAV. From the plot in Figure 4.19, it is evident that the quadrotor has followed the desired 3D trajectory accurately. No delay, overshoot, oscillation buildup, or drift is observed at any point in the simulation.

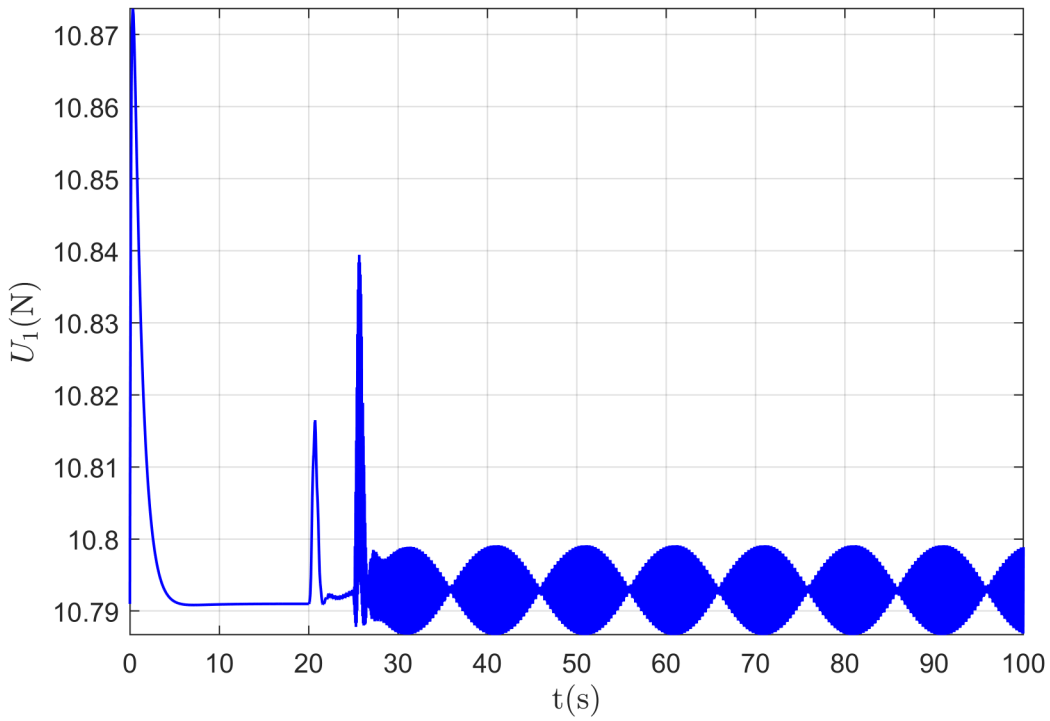

Figure 4.20: ASMC Control Input: U_1

Plot in Figure 4.20 represents the control input U_1, which is the total trust generated by the rotors to counteract gravity and control vertical motion (altitude). As shown in Figure, initial spike of approximately 10.88N is required to counteract gravity ($m = 1.1\,\text{kg}$ x $g = 9.81\,\text{m/s}^2 = 10.79\text{N}$) and initiate vertical motion. In the current scenario, the quadrotor is not only ascending vertically but also moving along a circular x-y path, resulting in an overall 3D helical trajectory. Furthermore, a quadrotor is a coupled system, exhibiting a strong coupling between the x, y, roll, and pitch states. At time $t = 20\,\text{sec}$ the desired longitudinal position $x_d(t)$ raises from 0 to a positive value as depicted in Figure 4.15, while, lateral position $y_d(t)$ drops from 0 to a negative value at the same time as can be seen in Figure 4.16. To generate the motions in x &

y directions, the quadrotor tilts. As a result of this tilt, the thrust vector is no longer purely vertical. Therefore, more thrust is required to maintain upward acceleration. Once this phase is over, the control input U_1 settles into a periodic oscillation pattern around the nominal thrust level (approximately 10.79 Nm).

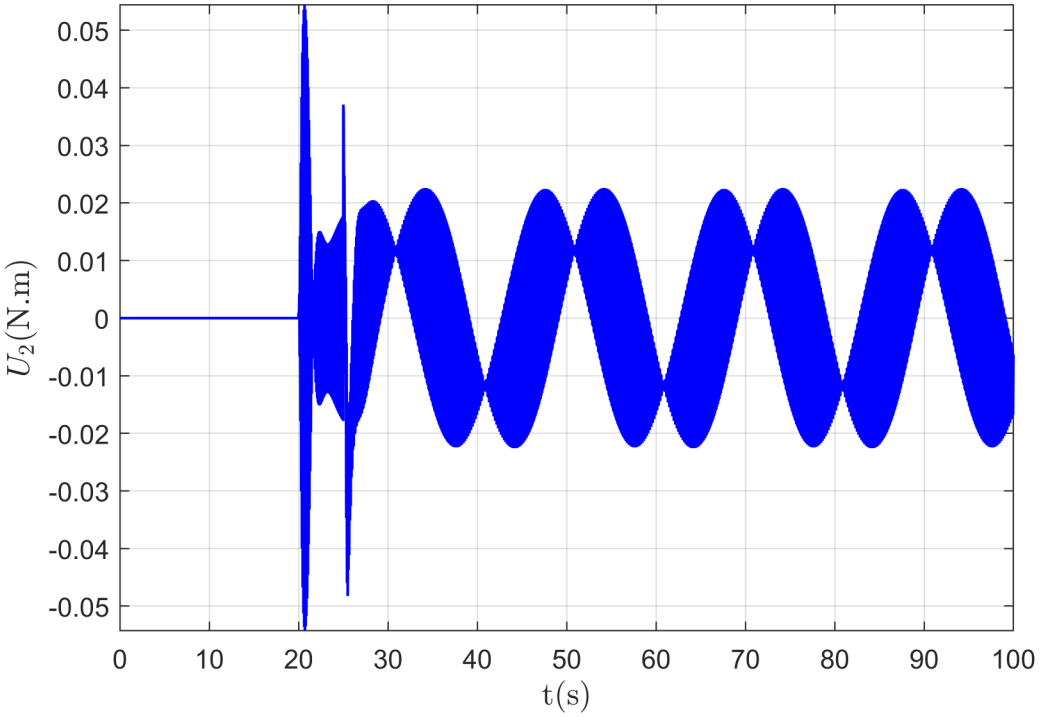

Figure 4.21: ASMC Control Input: U_2

Figure 4.21 shows the evolution of control effort U_2 (N.m) with respect to time t (in seconds) for a quadrotor UAV. In quadrotor UAV U_2, typically represents roll torque, which adjusts its orientation about its longitudinal axis. Plot depicts that from $t = 0$ to $t \approx 20$ seconds, the control input U_2 remains 0. It implies that no corrective roll torque is required around the longitudinal axis, as cross verifies from Figure 4.15. At $t = 20$ seconds, there is a sharp negative spike in U_2. This indicates the need for immediate torque correction, as the desired longitudinal position rapidly changes at that instant. From $t > 25$ to $t = 100$ seconds, oscillations can be observed which settle into a more periodic manner. These oscillations indicate the tracking of a sinusoidal trajectory. The periodic oscillations exhibit a smooth and regular waveform,

indicating a stable controller response. The waveform's constant amplitude over time in this region suggests that the system is neither drifting nor diverging.

Figure 4.22 shows the evolution of control effort U_3 (N.m) with respect to time t (in seconds) for a quadrotor UAV. In a quadrotor UAV U_3, typically represents pitch torque, which adjusts its orientation about its lateral axis. Plot depicts that from $t = 0$ to $t \approx 20$ seconds, the control input U_3 remains 0. It implies that no corrective pitch torque is required around the lateral axis, as cross-verification from Figure 4.16. At $t = 20$ seconds, there is a sharp spike in U_3. This indicates the need for immediate torque correction, as the desired lateral position rapidly changes at that instant. From $t > 25$ seconds onward, oscillations can be observed which settle into a more periodic manner. These oscillations indicate the tracking of a sinusoidal trajectory. The periodic oscillations exhibit a smooth and regular waveform, indicating a stable controller response. The constant amplitude of the waveform over time in this region suggests that the system is neither drifting nor diverging.

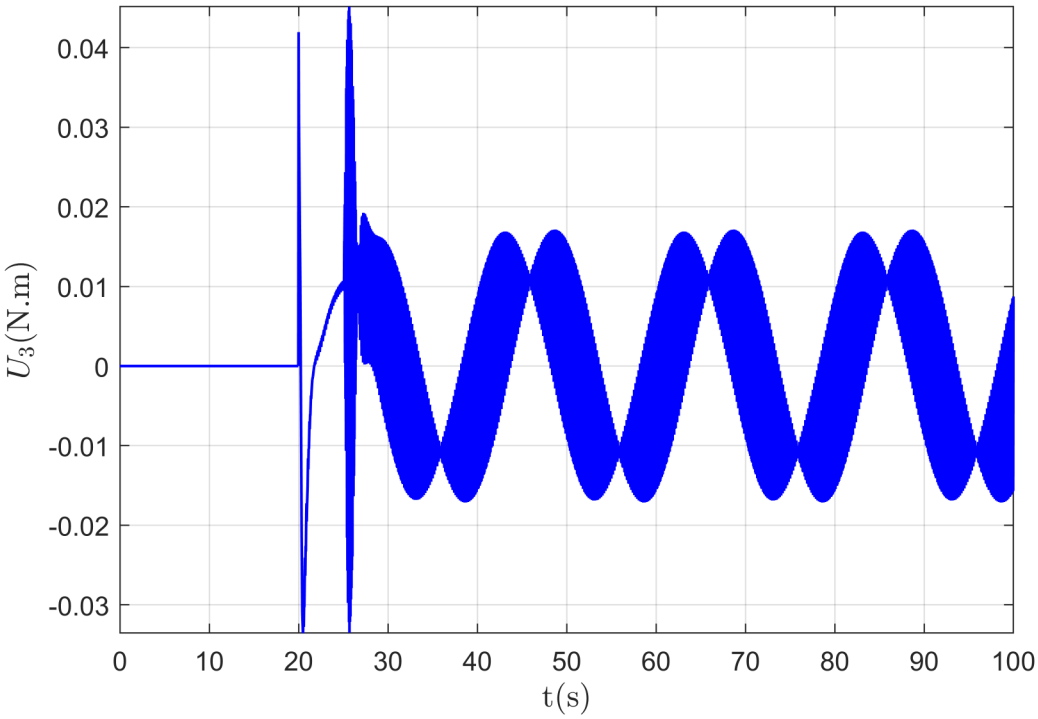

Figure 4.22: ASMC Control Input: U_3

Figure 4.23 shows the evolution of control effort U_4 (N.m) with respect to time t (in seconds) for a quadrotor UAV. In the quadrotor UAV U_4, responsible for yaw control, which adjusts its orientation about the vertical z axis. This affects the UAV's heading. As shown in the Figure, the Initial two positive and negative spikes, each approximately 0.25 N $\cdot$ m in magnitude, within the first 20 seconds, represent a sudden change in the yaw setpoint. This rapid change in yaw angle control can be counterchecked from Figure 4.18. It can be seen that the desired yaw angle abruptly changes from 0 to 1 radian and back to 0 radian over the first 20 seconds. To meet the yaw angle target, the control input U_4 is applied. Time from $t \approx 20$ to onward, the control input U_4 remains stable with small periodic oscillations centered around $0\,\mathrm{N}\cdot\mathrm{m}$. These small yaw corrections suggest that the UAV maintains a stable heading while exhibiting only minor periodic motions, possibly due to coupled roll or pitch motions.

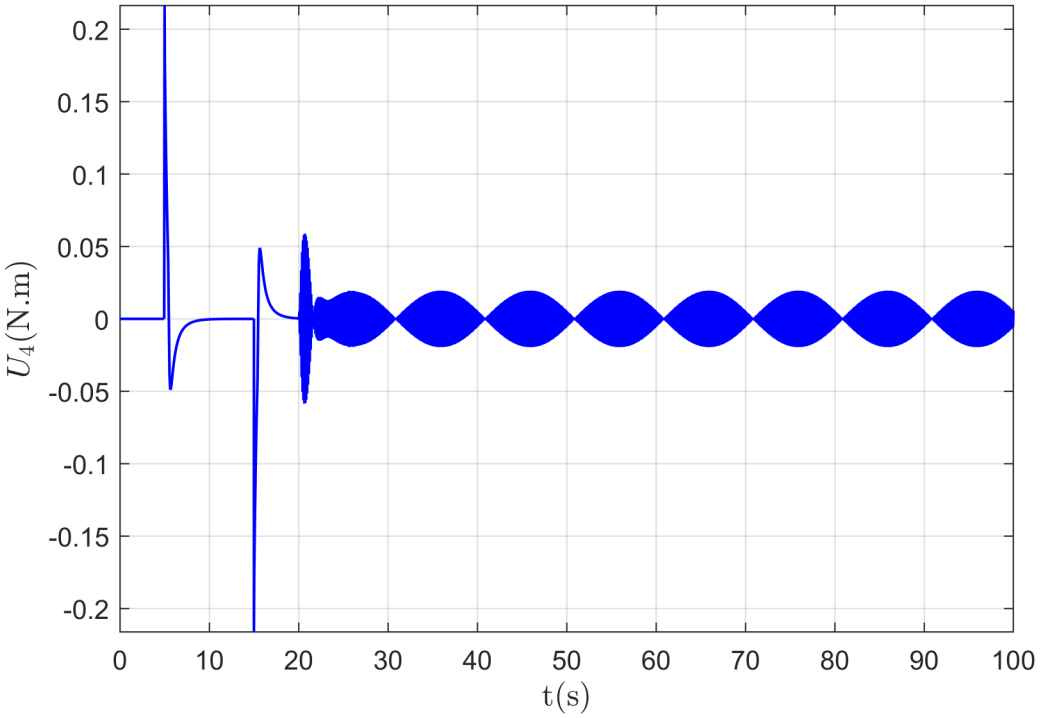

Figure 4.23: ASMC Control Input: U_4

Summary

The chapter presents the SMC and ASMC frameworks for quadrotor UAVs. Closed-loop system stability and finite-time convergence of the system states are rigorously proven using Lyapunov-based analysis under nonlinear dynamics with bounded disturbances. Furthermore, an adaptive law is introduced to eliminate the requirement for prior knowledge of disturbance bounds, thereby enhancing robustness. The proposed SMC and ASMC schemes are designed for a six-degree-of-freedom quadrotor UAV. Simulation results confirm accurate tracking performance, robustness against uncertain disturbances, and stable control behavior. Therefore, providing a solid foundation for the neural network–enhanced controllers presented in subsequent chapters.

Chapter 5
Adaptive NN Techniques for Nonlinear Gain Estimation

A neural network is a machine learning model designed to emulate the human brain's decision-making process. It operates using a network of interconnected nodes, analogous to biological neurons, to identify patterns, analyze data, and make decisions based on prior experience. While inspired by biology, neural networks are implemented using mathematical models.

Figure 5.1 illustrates the commonly used mathematical representation of a neuron in an ANN. Each neuron receives input signals denoted by $x_1, x_2, x_3, \ldots, x_n$, collectively represented as the vector $\mathbf{x}$. These inputs typically originate from other neurons in the network. Each input is associated with a weight w_i, reflecting the strength or effectiveness of the corresponding synapse. The weighted inputs are then passed to a summation function, representing the cell body, where they are combined algebraically to determine the excitability of the neuron:

$$y_{in} = \sum_{i=1}^{n} x_i w_i \tag{5.1}$$

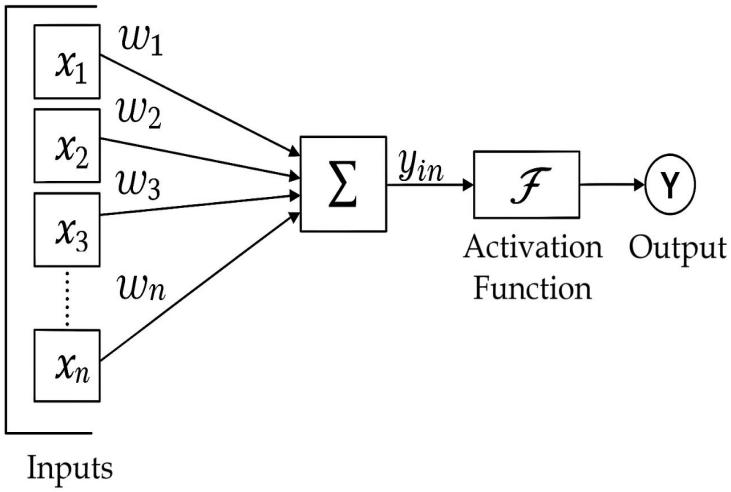

Figure 5.1: Mathematical Model of Neuron

By passing the excitement level through the function $\mathcal{F}(y_{in})$, also known as the *activation function* as in Equation 5.2, one may determine the output signal of a neuron.

$$Y = \mathcal{F}(y_{in}) \tag{5.2}$$

5.1 Architecture of Artificial Neural Network

Artificial neurons are connected to form neural networks that process multiple inputs to generate a single output. An Artificial Neural Network's primary purpose is to map input to a desirable form in production [50]. The neural network has many layers. Each layer served its specific function, and the deeper layers make the system more complex. A neural network is also known as the multilayer perceptron. To build neural networks, three types of layers are used: an input layer that accepts data from external sources, a hidden layer that performs computations according to the specified function, and an output layer that produces results based on the input. The input layer passes data to the hidden layer. Each input is connected to the next layer via links that assign random weights. These weights are multiplied individually by the corresponding inputs, and a bias term is added afterwards. An activation function

then processes the resulting weighted sum. This processed value is passed to the output layer using an output function. To minimize the error, the model adjusts its weights using backpropagation.

Artificial neural networks (ANNs) can be categorized into two main architectural types: feedforward neural networks and recurrent neural networks (RNNs). In a feedforward ANN, signals flow in a single direction—from the input layer, through any hidden layers, and finally to the output layer. These networks, which may contain one or more hidden layers, are commonly used for pattern recognition tasks. A popular example of a feedforward architecture is the multilayer perceptron (MLP), which consists of three types of layers: input, hidden, and output.

On the other hand, recurrent neural networks (RNNs) have a different structure that incorporates internal memory. They include loops within the hidden layers, allowing information to flow both forward and backward through the network. This design enables RNNs to maintain context from previous inputs, making them well-suited for tasks involving sequences or time-series data. By leveraging past outputs, RNNs improve their predictive accuracy.

5.2 Levenberg-Marquardt Algorithm

The Levenberg-Marquardt (LM) algorithm, developed by Donald Marquardt and Kenneth Levenberg, is a fast and stable mathematical method for minimizing nonlinear functions. It is employed to train artificial neural networks to resolve problems ranging from small to medium scale. Error backpropagation (EBP), also known as steepest descent, is a significant breakthrough in neural network training. Although it is well known for its slow convergence, there are two primary reasons: step sizes must be appropriate to the gradients, whereas the curvature is not the same in all directions on the error surface, culminating in the classic "error valley" problem. The prescribed Gauss-Newton algorithm can improve the moderate convergence of the steepest-descent method by using 2^{nd}-order derivatives to estimate the curvature of the error surface. This algorithm can achieve suitable step sizes in all directions and

consequently converge rapidly, particularly for the quadratic error surface. However, this enhancement occurs when the approximation to the quadratic error function is valid; otherwise, the Gauss-Newton algorithm would be predominantly divergent. [51]

The Levenberg-Marquardt (LM) algorithm combines the Gauss-Newton algorithm and the steepest-descent method, offering two-fold advantages: speed and stability. It provides greater robustness than the Gauss-Newton algorithm, as it can converge swiftly even in the presence of complex error surfaces. Furthermore, it converges faster than the steepest descent method. The devised algorithm implements a combined training process that alternates between the steepest-descent and Gauss-Newton algorithms until the local curvature is sufficiently flat to warrant a quadratic approximation. This approach significantly speeds up convergence. [51]

5.2.1 Derivation of Levenberg-Marquardt Algorithm

This section entails the derivation of the Levenberg-Marquardt algorithm that will be described in the following four segments: (1) Newton's method, (2) Gauss–Newton's algorithm, (3) Steepest descent algorithm, and (4) Levenberg–Marquardt algorithm. Some commonly used indices are introduced ahead of the derivation:

- m is related to the index of outputs, ranging from 1 to M, where M represents the number of outputs.
- p is related to the index of patterns, ranging from 1 to P, where P represents the number of patterns.
- k is the index of iterations.
- i and j are the weight indices, from 1 to N, where N represents the aggregate of weights.

The sum of squared error (SSE) is used to analyze the training process. It is quantified for network outputs and training patterns as

$$E(\mathbf{x}, \theta) = \frac{1}{2} \sum_{p=1}^{P} \sum_{m=1}^{M} e_{p,m}^2 \tag{5.3}$$

where, $\mathbf{x}$ and θ are denoting the input vector and weight vector respectively, $e_{p,m}$ is the training error at output m when applying pattern p and it is explained as

$$e_{p,m} = d_{p,m} - o_{p,m} \tag{5.4}$$

where, $\mathbf{d}$ is expressing the vector of desired output, and, $\mathbf{o}$ is denoting the vector of actual output

5.2.2 Steepest Descent Algorithm

The steepest-descent algorithm is a 1^{st}-order algorithm because it uses the first-order partial derivative of the total error function to find minima in the error space. The gradient $\mathbf{g}$ is described as the first-order partial derivative of the total error function:

$$\mathbf{g} = \frac{\partial E(\mathbf{x}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial E}{\partial \theta_1} & \frac{\partial E}{\partial \theta_2} & \cdots & \frac{\partial E}{\partial \theta_N} \end{bmatrix}^T \tag{5.5}$$

Keeping in view the above-mentioned definition of the gradient $\mathbf{g}$, the rule of updation for the steepest descent algorithm can be described as:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \alpha \mathbf{g}_k \tag{5.6}$$

where α denotes the step size learning constant.

The entire training process ensures asymptotic convergence of the steepest-descent algorithm. Approaching the vicinity of the solution, each element of the gradient vector tends to become very minor, and there could be a minimal change in weight.

5.2.3 Newton's Method

Newton's method presumes that the entirety of weights are linearly independent and all the gradient elements $g_1, g_2, \ldots, g_N$ are weight functions expressed as:

$$\begin{cases} g_1 = F_1(\theta_1, \theta_2, \cdots, \theta_N) \\ g_2 = F_2(\theta_1, \theta_2, \cdots, \theta_N) \\ \quad \vdots \\ g_N = F_N(\theta_1, \theta_2, \cdots, \theta_N) \end{cases} \tag{5.7}$$

Where $F_1, F_2, \ldots, F_N$ are nonlinear relationships among related gradient components and weights. Simplifying each g_i $(i = 1, 2, \ldots, N)$ in Equation 5.7 by the Taylor series and assuming the approximation of first-order:

$$\begin{cases} g_1 \approx g_{1,0} + \dfrac{\partial g_1}{\partial \theta_1}\Delta\theta_1 + \dfrac{\partial g_1}{\partial \theta_2}\Delta\theta_2 + \cdots + \dfrac{\partial g_1}{\partial \theta_N}\Delta\theta_N \\ g_2 \approx g_{2,0} + \dfrac{\partial g_2}{\partial \theta_1}\Delta\theta_1 + \dfrac{\partial g_2}{\partial \theta_2}\Delta\theta_2 + \cdots + \dfrac{\partial g_2}{\partial \theta_N}\Delta\theta_N \\ \quad \vdots \\ g_N \approx g_{N,0} + \dfrac{\partial g_N}{\partial \theta_1}\Delta\theta_1 + \dfrac{\partial g_N}{\partial \theta_2}\Delta\theta_2 + \cdots + \dfrac{\partial g_N}{\partial \theta_N}\Delta\theta_N \end{cases} \tag{5.8}$$

By synthesizing the description of the gradient vector $\mathbf{g}$ in Equation 5.5, the following term can be established as:

$$\frac{\partial g_i}{\partial \theta_j} = \frac{\partial}{\partial \theta_j}\left(\frac{\partial E}{\partial \theta_i}\right) = \frac{\partial^2 E}{\partial \theta_j \partial \theta_i} \tag{5.9}$$

By inserting Equation 5.9 to 5.8:

$$\begin{cases} g_1 \approx g_{1,0} + \dfrac{\partial^2 E}{\partial \theta_1^2}\Delta\theta_1 + \dfrac{\partial^2 E}{\partial \theta_1 \partial \theta_2}\Delta\theta_2 + \cdots + \dfrac{\partial^2 E}{\partial \theta_1 \partial \theta_N}\Delta\theta_N \\ g_2 \approx g_{2,0} + \dfrac{\partial^2 E}{\partial \theta_2 \partial \theta_1}\Delta\theta_1 + \dfrac{\partial^2 E}{\partial \theta_2^2}\Delta\theta_2 + \cdots + \dfrac{\partial^2 E}{\partial \theta_2 \partial \theta_N}\Delta\theta_N \\ \quad \vdots \\ g_N \approx g_{N,0} + \dfrac{\partial^2 E}{\partial \theta_N \partial \theta_1}\Delta\theta_1 + \dfrac{\partial^2 E}{\partial \theta_N \partial \theta_2}\Delta\theta_2 + \cdots + \dfrac{\partial^2 E}{\partial \theta_N^2}\Delta\theta_N \end{cases} \tag{5.10}$$

Compared with the steepest-descent method, the second-order derivatives of the total error function are constrained to be estimated for all components of the gradient vector. However, to obtain the minima for the total error function E, each element of the gradient vector necessitates that it should be presumed as zero. Consequently, left hand sides of the Equations (5.10) are all zero, then

$$\begin{cases} 0 = g_{1,0} + \dfrac{\partial^2 E}{\partial \theta_1^2}\Delta\theta_1 + \dfrac{\partial^2 E}{\partial \theta_1 \partial \theta_2}\Delta\theta_2 + \cdots + \dfrac{\partial^2 E}{\partial \theta_1 \partial \theta_N}\Delta\theta_N \\ 0 = g_{2,0} + \dfrac{\partial^2 E}{\partial \theta_2 \partial \theta_1}\Delta\theta_1 + \dfrac{\partial^2 E}{\partial \theta_2^2}\Delta\theta_2 + \cdots + \dfrac{\partial^2 E}{\partial \theta_2 \partial \theta_N}\Delta\theta_N \\ \quad \vdots \\ 0 = g_{N,0} + \dfrac{\partial^2 E}{\partial \theta_N \partial \theta_1}\Delta\theta_1 + \dfrac{\partial^2 E}{\partial \theta_N \partial \theta_2}\Delta\theta_2 + \cdots + \dfrac{\partial^2 E}{\partial \theta_N^2}\Delta\theta_N \end{cases} \tag{5.11}$$

By combining the Equations 5.5 and 5.11, we get:

$$\begin{cases} -\dfrac{\partial E}{\partial \theta_1} = -g_{1,0} \approx \dfrac{\partial^2 E}{\partial \theta_1^2}\Delta\theta_1 + \dfrac{\partial^2 E}{\partial \theta_1 \partial \theta_2}\Delta\theta_2 + \cdots + \dfrac{\partial^2 E}{\partial \theta_1 \partial \theta_N}\Delta\theta_N \\ -\dfrac{\partial E}{\partial \theta_2} = -g_{2,0} \approx \dfrac{\partial^2 E}{\partial \theta_2 \partial \theta_1}\Delta\theta_1 + \dfrac{\partial^2 E}{\partial \theta_2^2}\Delta\theta_2 + \cdots + \dfrac{\partial^2 E}{\partial \theta_2 \partial \theta_N}\Delta\theta_N \\ \qquad\qquad \vdots \\ -\dfrac{\partial E}{\partial \theta_N} = -g_{N,0} \approx \dfrac{\partial^2 E}{\partial \theta_N \partial \theta_1}\Delta\theta_1 + \dfrac{\partial^2 E}{\partial \theta_N \partial \theta_2}\Delta\theta_2 + \cdots + \dfrac{\partial^2 E}{\partial \theta_N^2}\Delta\theta_N \end{cases} \tag{5.12}$$

There are N parameters for defining N equations, so that each of the Δw_i can be computed. Given the solutions, the weight space can be updated iteratively. The

matrix form of Equation 5.12 is:

$$\begin{bmatrix} -g_1 \\ -g_2 \\ \vdots \\ -g_N \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 E}{\partial \theta_1^2} & \frac{\partial^2 E}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2 E}{\partial \theta_1 \partial \theta_N} \\ \frac{\partial^2 E}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 E}{\partial \theta_2^2} & \cdots & \frac{\partial^2 E}{\partial \theta_2 \partial \theta_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial \theta_N \partial \theta_1} & \frac{\partial^2 E}{\partial \theta_N \partial \theta_2} & \cdots & \frac{\partial^2 E}{\partial \theta_N^2} \end{bmatrix} \begin{bmatrix} \Delta\theta_1 \\ \Delta\theta_2 \\ \vdots \\ \Delta\theta_N \end{bmatrix} \tag{5.13}$$

The Hessian square matrix for second-order partial derivatives is defined as:

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 E}{\partial \theta_1^2} & \frac{\partial^2 E}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2 E}{\partial \theta_1 \partial \theta_N} \\ \frac{\partial^2 E}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 E}{\partial \theta_2^2} & \cdots & \frac{\partial^2 E}{\partial \theta_2 \partial \theta_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial \theta_N \partial \theta_1} & \frac{\partial^2 E}{\partial \theta_N \partial \theta_2} & \cdots & \frac{\partial^2 E}{\partial \theta_N^2} \end{bmatrix} \tag{5.14}$$

By combining the gradient equation with the Hessian matrix, we can write:

$$-\boldsymbol{g} = \mathbf{H}\Delta\boldsymbol{\theta} \tag{5.15}$$

Solving for the update step:

$$\Delta\boldsymbol{\theta} = -\mathbf{H}^{-1}\boldsymbol{g} \tag{5.16}$$

Consequently, the update rule of Newton's method is described as:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \mathbf{H}_k^{-1}\boldsymbol{g}_k \tag{5.17}$$

The total error function is composed of second-order derivatives, and the Hessian matrix $\mathbf{H}$ provides an accurate assessment of the changes in the gradient. By comparing the gradient descent rule with Newton's update, one sees that optimal step sizes are scaled upon obtaining the inverse of the Hessian matrix.

5.2.4 Guass-Newton Algorithm

For weight updation, as long as Newton's method is applied, to compute the Hessian matrix **H**, the derivatives of the second-order need to be evaluated for the total error function. However, this entire process can be very complex and computationally expensive. The Jacobian matrix **J** is defined to simplify the calculation. The Jacobian matrix consists of first-order derivatives of the error terms concerning the weights:

$$\begin{bmatrix} \frac{\partial e_{1,1}}{\partial \theta_1} & \frac{\partial e_{1,1}}{\partial \theta_2} & \cdots & \frac{\partial e_{1,1}}{\partial \theta_N} \\ \frac{\partial e_{1,2}}{\partial \theta_1} & \frac{\partial e_{1,2}}{\partial \theta_2} & \cdots & \frac{\partial e_{1,2}}{\partial \theta_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_{P,M}}{\partial \theta_1} & \frac{\partial e_{P,M}}{\partial \theta_2} & \cdots & \frac{\partial e_{P,M}}{\partial \theta_N} \end{bmatrix} \tag{5.18}$$

By integrating the Equations 5.3 and 5.5, the calculation of gradient vector elements is performed as

$$g_i = \frac{\partial E}{\partial \theta_i} = \frac{\partial}{\partial \theta_i}\left(\frac{1}{2}\sum_{p=1}^{P}\sum_{m=1}^{M} e_{p,m}^2\right) = \sum_{p=1}^{P}\sum_{m=1}^{M}\left(\frac{\partial e_{p,m}}{\partial \theta_i} e_{p,m}\right) \tag{5.19}$$

By synthesizing Equations 5.18 and 5.19, the Jacobian matrix **J** relation with gradient vector **g** is described as

$$\mathbf{g} = \mathbf{J}\mathbf{e} \tag{5.20}$$

where **e** is error vector.

By inserting Equation 5.3 into 5.14, the Hessian matrix entries at ith row and jth column can be calculated as

$$h_{i,j} = \frac{\partial^2 E}{\partial \theta_i \partial \theta_j} = \frac{\partial^2}{\partial \theta_i \partial \theta_j}\left(\frac{1}{2}\sum_{p=1}^{P}\sum_{m=1}^{M} e_{p,m}^2\right) = \sum_{p=1}^{P}\sum_{m=1}^{M}\left(\frac{\partial e_{p,m}}{\partial \theta_i}\frac{\partial e_{p,m}}{\partial \theta_j}\right) + S_{i,j} \tag{5.21}$$

where $S_{i,j}$ is equivalent to

$$S_{i,j} = \sum_{p=1}^{P}\sum_{m=1}^{M}\frac{\partial^2 e_{p,m}}{\partial \theta_i \partial \theta_j} e_{p,m} \tag{5.22}$$

The Hessian matrix $\mathbf{H}$ relationship with the Jacobian matrix $\mathbf{J}$ is modified by assuming that $S_{i,j}$ is nearly zero

$$\mathbf{H} \approx \mathbf{J}^T\mathbf{J} \tag{5.23}$$

By synthesizing Equations 5.17, 5.20, and (5.23), the updating rule for the Gauss–Newton algorithm can be expressed as

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \left(\mathbf{J}_k^T\mathbf{J}_k\right)^{-1}\mathbf{J}_k\mathbf{e}_k \tag{5.24}$$

It is evident that the Gauss–Newton algorithm offers an advantage when compared with standard Newton's method (Equation 5.17) as it avoids the calculation of derivatives of second-order for the total error function, and alternatively uses the Jacobian matrix $\mathbf{J}$. However, convergence issues arise in both the Gauss-Newton and Newton algorithms when minimizing the error space. The problem mentioned above can be mathematically interpreted as the matrix $\mathbf{J}^T\mathbf{J}$ is not necessarily invertible.

5.2.5 Levenberg-Marquardt Algorithm

The Levenberg–Marquardt algorithm provides an alternative approximation to ensure that the estimated Hessian matrix $\mathbf{J}^T\mathbf{J}$ is invertible:

$$\mathbf{H} \approx \mathbf{J}^T\mathbf{J} + \mu\mathbf{I} \tag{5.25}$$

where

- μ is termed a combination coefficient, and it is always positive.
- $\mathbf{I}$ is the corresponding Identity operator

According to the Equation (5.25), it is evident that the entries corresponding to the main diagonal of the estimated Hessian matrix will be nonzero. Hence, with the said approximation, it can be guaranteed that the matrix $\mathbf{H}$ is always invertible.

By combining Equations (5.24) and (5.25), the update rule for the

Levenberg–Marquardt algorithm can be expressed as

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \left(\mathbf{J}_k^T \mathbf{J}_k + \mu \mathbf{I}\right)^{-1} \mathbf{J}_k^T \mathbf{e}_k \tag{5.26}$$

The Levenberg-Marquardt algorithm combines the steepest descent and Gauss-Newton algorithms. During training, it alternates between the two algorithms mentioned above. When the combination coefficient μ advances toward zero, Equation (5.26) approaches Equation (5.24) and the Gauss–Newton algorithm is employed. On the contrary, Equation (5.26) approximates Equation 5.6, when the combination coefficient μ is very large, and the steepest descent technique is utilized accordingly.

Moreover, the combination coefficient μ in Equation (5.26) can be defined as the learning coefficient:

$$\alpha = \frac{1}{\mu} \tag{5.27}$$

Table 5.1: Technical Aspects of Different Algorithms

Algorithms	**Rules for Parameter Update**	**Convergence**	**Derivative Information Required**
EBP algorithm	$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \alpha \mathbf{g}_k$	Slow but stable	Gradient
Newton algorithm	$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \mathbf{H}_k^{-1} \mathbf{g}_k$	Fast but unstable	Hessian and Gradient
Gauss Newton algorithm	$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - (\mathbf{J}_k^T \mathbf{J}_k)^{-1} \mathbf{J}_k \mathbf{e}_k$	Fast but unstable	Jacobian
Levenberg Marquardt algorithm	$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - (\mathbf{J}_k^T \mathbf{J}_k + \mu \mathbf{I})^{-1} \mathbf{J}_k \mathbf{e}_k$	Fast and stable	Jacobian

5.3 Simulation Results of Adaptive Back Propagation SMC (ABPSMC)

The BP optimization scheme is implemented on a feedforward network. Similar to SMC, the performance of the Feedforward NN using the BP optimization scheme is also evaluated in Simulink/Matlab. The system parameters are the same as those used in the SMC evaluation. The initial conditions, as well as the simulation settings, are also the same as those used in the SMC evaluation.

The network has two hidden layers, each with 16 neurons. Learning rate α has the value of 0.001, and the maximum number of epochs is 1000. Moreover, it could also stop if the gradient fell below 1e-6. The total dataset comprises 1 million samples, with two input features (error and error dot), and one output feature: Adaptive Modulation Gain. The dataset was split into three portions: a 70 percent training set, a 15 percent testing set, and a 15 percent validation set. A total of six neural networks are trained, with one independent NN trained for each state (x, y, z, roll, pitch, yaw).

5.3.1 Simulation Results

Similar to SMC, the performance of the Feedforward NN using the LM optimization scheme is also evaluated in Simulink/Matlab. The system parameters are the same as those used in the SMC evaluation. The initial conditions and simulation settings are identical to those used in the SMC evaluation.

The graph in Figure 5.2 shows the performance of the Feedforward NN, trained using the BP optimization scheme for longitudinal position control of quadrotor UAV over a 100-second time frame window. The solid red line represents the desired longitudinal position, while the dotted black line represents the actual longitudinal position achieved by the quadrotor. As depicted by the plot, the error between the actual and desired trajectories is minimal. The actual position trajectory is almost identical to the desired position throughout the entire time frame, indicating the controller's high accuracy and robustness. Furthermore, no phase lag is found, and overshoot is nearly zero. Overall, the system remains stable, no divergence, oscillation

buildup, or drift is observed.

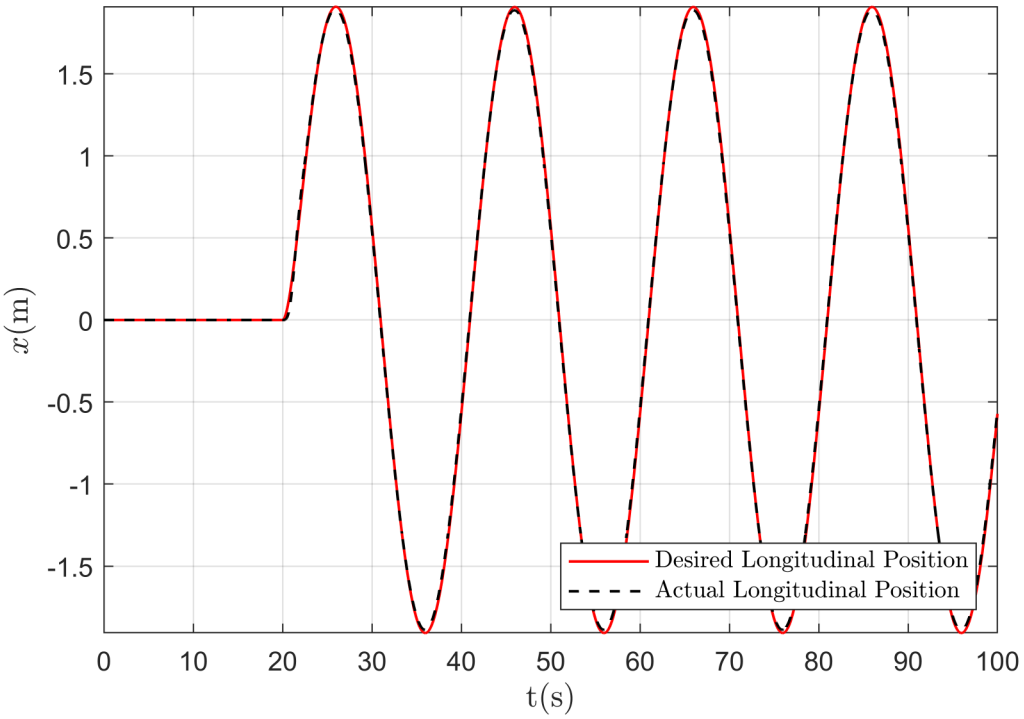

Figure 5.2: Longitudinal Position Control Using ABPSMC

The graph in Figure 5.3 shows the performance of the Feedforward NN, trained using the BP optimization scheme for lateral position control of quadrotor UAV over a 100-second time frame window. The solid red line represents the desired lateral position, while the dotted black line represents the actual lateral position achieved by the quadrotor. As depicted by the plot, the error between the actual and desired trajectories is minimal. The actual position trajectory is almost identical to the desired position throughout the entire time frame, indicating the controller's high accuracy and robustness. Furthermore, no phase lag is found, and overshoot is nearly zero. Overall, the system remains stable, no divergence, oscillation buildup, or drift is observed. The plot in Figure 5.4 represents the performance of altitude control for a quadrotor UAV. The solid red line represents the desired altitude, while the dotted black line represents the actual altitude attained by the quadrotor. The desired result is achieved by FNN using the BP optimization scheme. No delay, overshoot, oscillation buildup, or drift is observed at any point in the simulation.

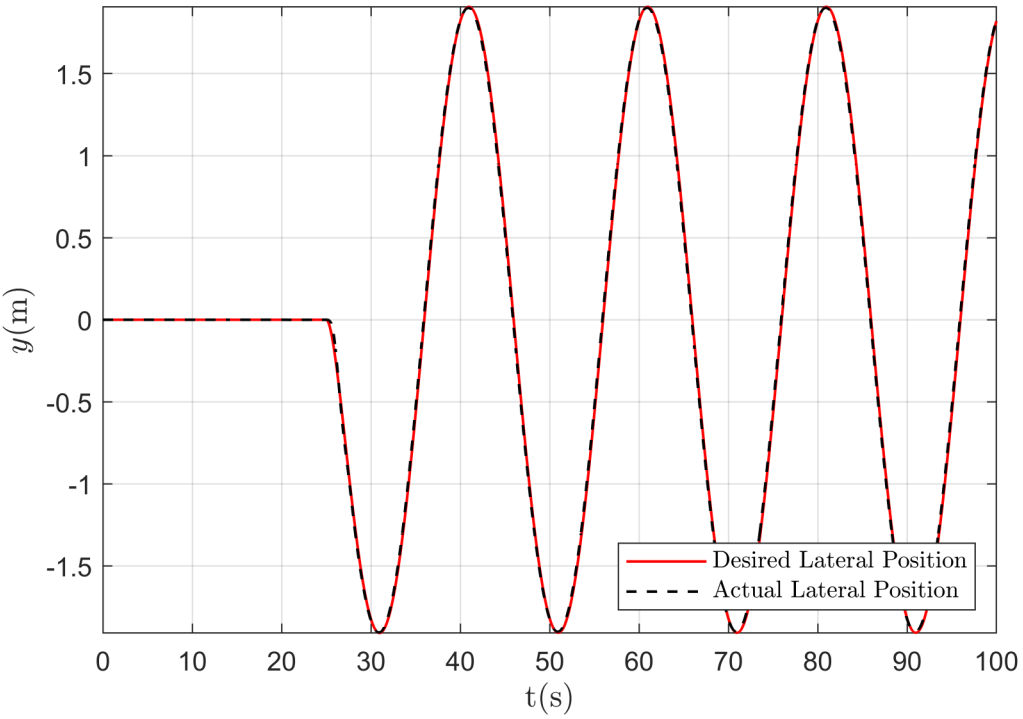

Figure 5.3: Lateral Position Control Using ABPSMC

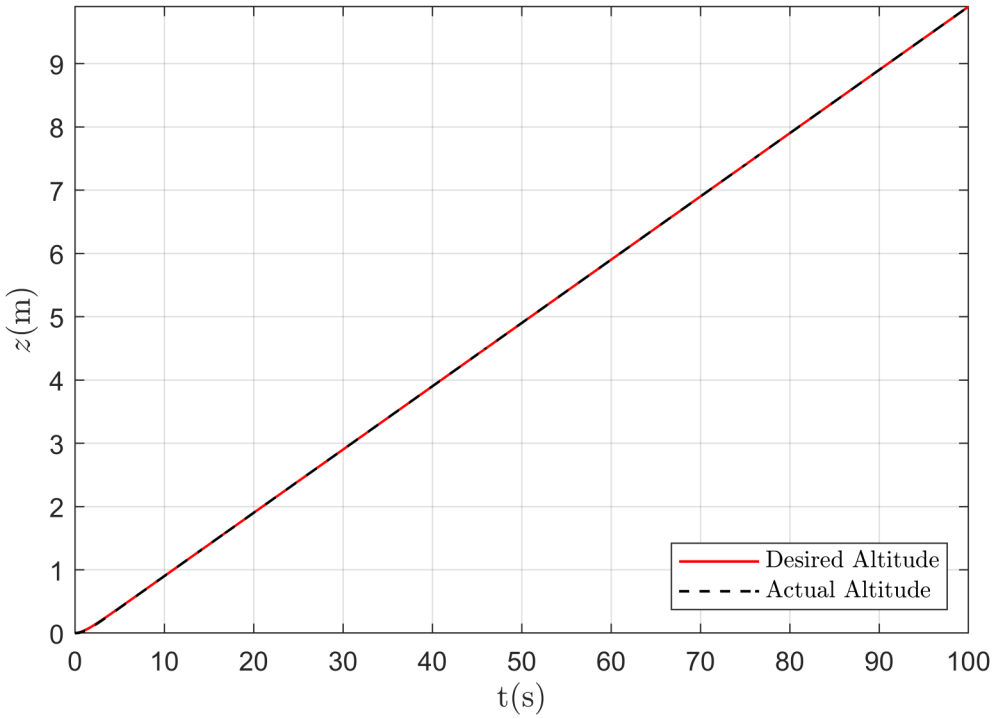

Figure 5.4: Altitude Control Using ABPSMC

The plot in Figure 5.5 represents the performance of yaw angle control for a quadrotor UAV. The solid red line represents the desired yaw angle, while the dotted black line represents the actual yaw angle attained by the quadrotor. The yaw angle is traced perfectly by the Feedforward NN, trained using the BP optimization scheme. No delay, overshoot, oscillation buildup, or drift is observed at any point in the simulation.

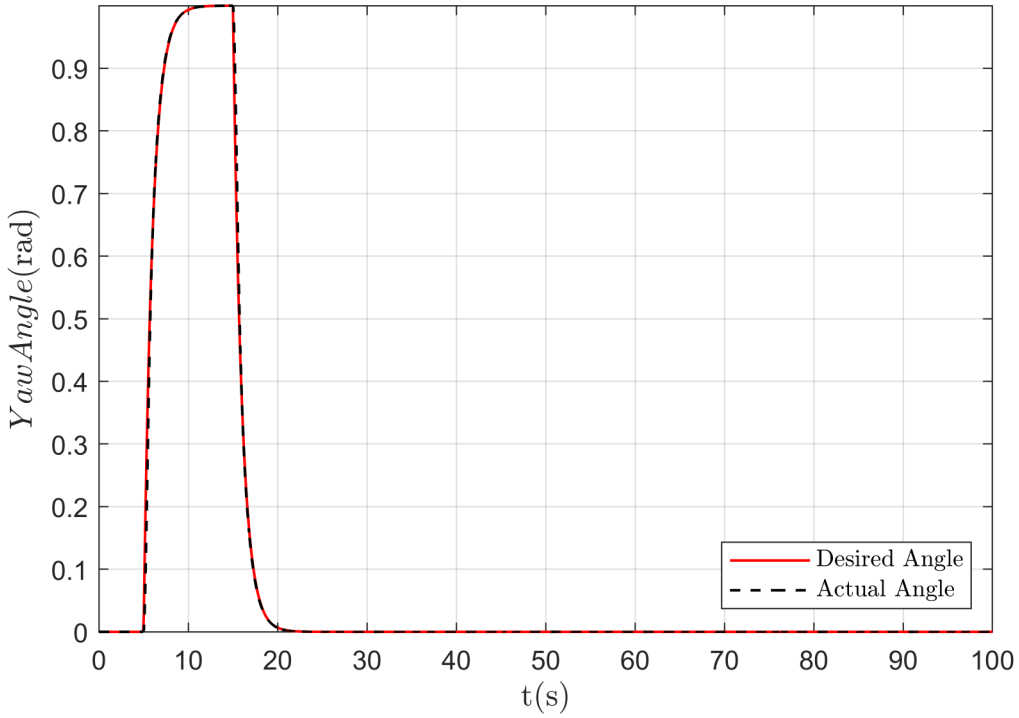

Figure 5.5: Yaw Angle Control Using ABPSMC

The 3D trajectory plot in Figure 5.6 depicts the performance of quadrotor UAV tracking in a helical/spiral path in the x,y,z (3D) space. The solid red line shows the desired trajectory, while the dotted black line illustrates the actual trajectory followed by the UAV. From the plot in Figure 5.6, it is evident that the quadrotor has followed the desired 3D trajectory accurately. No delay, overshoot, oscillation buildup, or drift is observed at any point in the simulation.

Plot in Figure 5.7 represents the control input $U_1(N)$, which is the total trust generated by the rotors to counteract gravity and control vertical motion (altitude). As shown in Figure, initial spike of approximately $10.88N$ is required to counteract

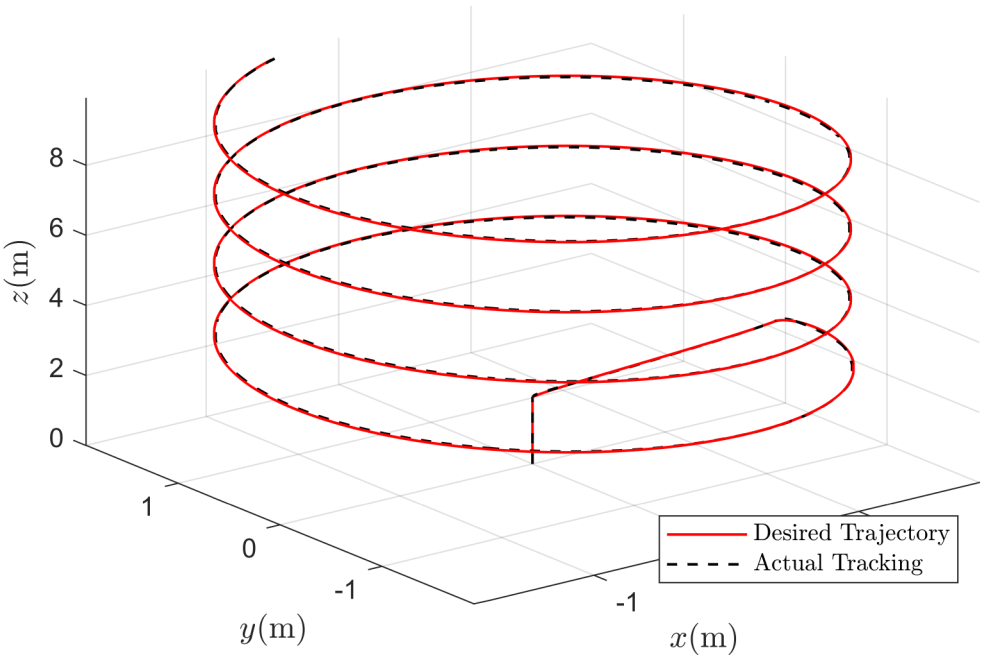

Figure 5.6: 3D Maneuvering of Quadrotor UAV using ABPSMC

gravity ($m = 1.1\,\text{kg}$ x $g = 9.81\,\text{m/s}^2 = 10.79N$) and initiate vertical motion. In the current scenario, the quadrotor is not only ascending vertically but also moving along a circular x-y path, resulting in an overall 3D helical trajectory. Furthermore, a quadrotor is a coupled system, exhibiting a strong coupling between the x, y, roll, and pitch states. At time $t = 20\,\text{sec}$ the desired longitudinal position x_d raises from 0 to a positive value as depicted in Figure 5.2, while, lateral position y_d drops from 0 to a negative value at the same time as can be seen in Figure 5.3. To generate the motions in x & y directions, the quadrotor tilts. As a result of this tilt, the thrust vector is no longer purely vertical. Therefore, more thrust is required to maintain upward acceleration. Once this phase is over, the control input U_1 settles into a periodic oscillation pattern around the nominal thrust level (approximately $10.79\,\text{Nm}$).

Figure 5.8 shows the evolution of control effort U_2 (N.m) with respect to time t (in seconds) for a quadrotor UAV. In quadrotor UAV U_2, typically represents roll torque, which adjusts its orientation about its longitudinal axis. Plot depicts that from $t = 0$ to $t \approx 20$ seconds, the control input U_2 remains 0. It implies that no corrective roll

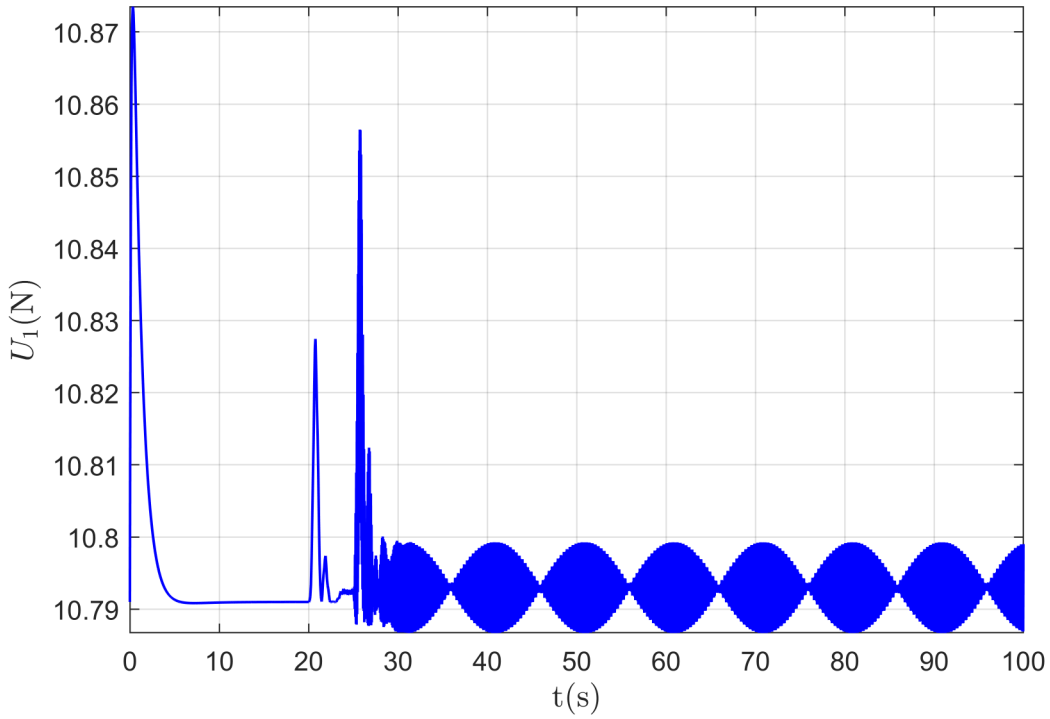

Figure 5.7: ABPSMC Control Input: U_1

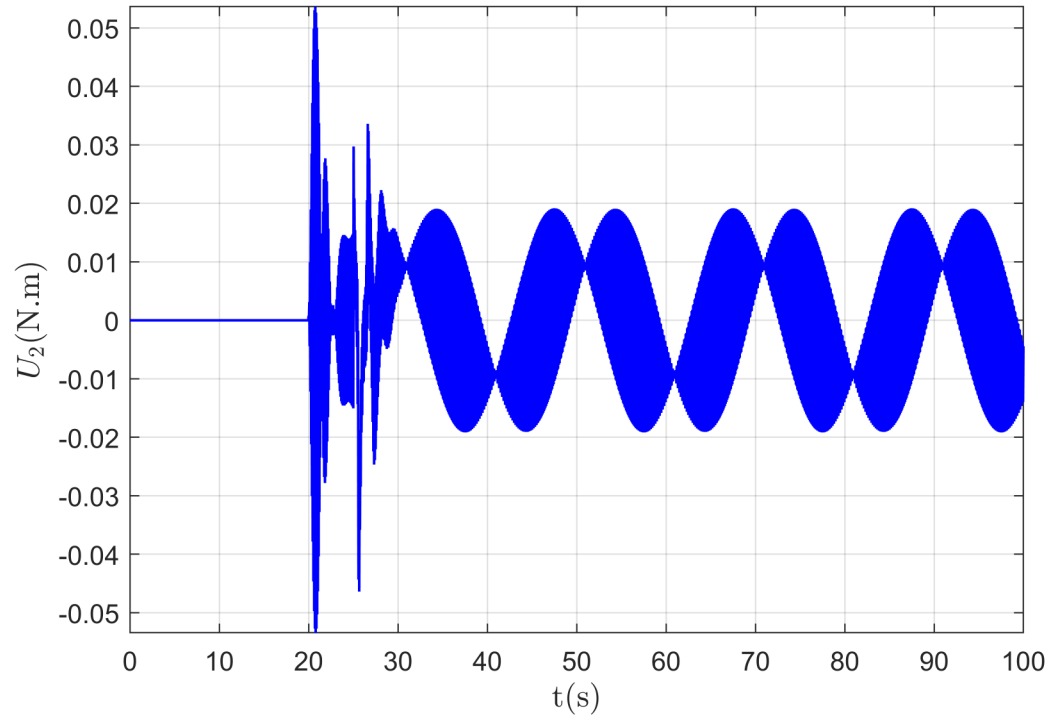

Figure 5.8: ABPSMC Control Input: U_2

torque is required around the longitudinal axis, as cross verifies from Figure 5.2. At $t = 20$ seconds, sharp control action occurs as the desired longitudinal position rapidly changes at that instant. This indicates the need for immediate torque correction. From $t > 25$ to $t = 100$ seconds, damped oscillations can be observed which settle into a more periodic manner. These oscillations indicate the tracking of a sinusoidal trajectory. The rapid damping indicates effective control and stabilization. The rapid damping indicates effective control and stabilization. The periodic oscillations exhibit a smooth and regular waveform, indicating a stable controller response. The waveform's constant amplitude over time in this region suggests that the system is neither drifting nor diverging.

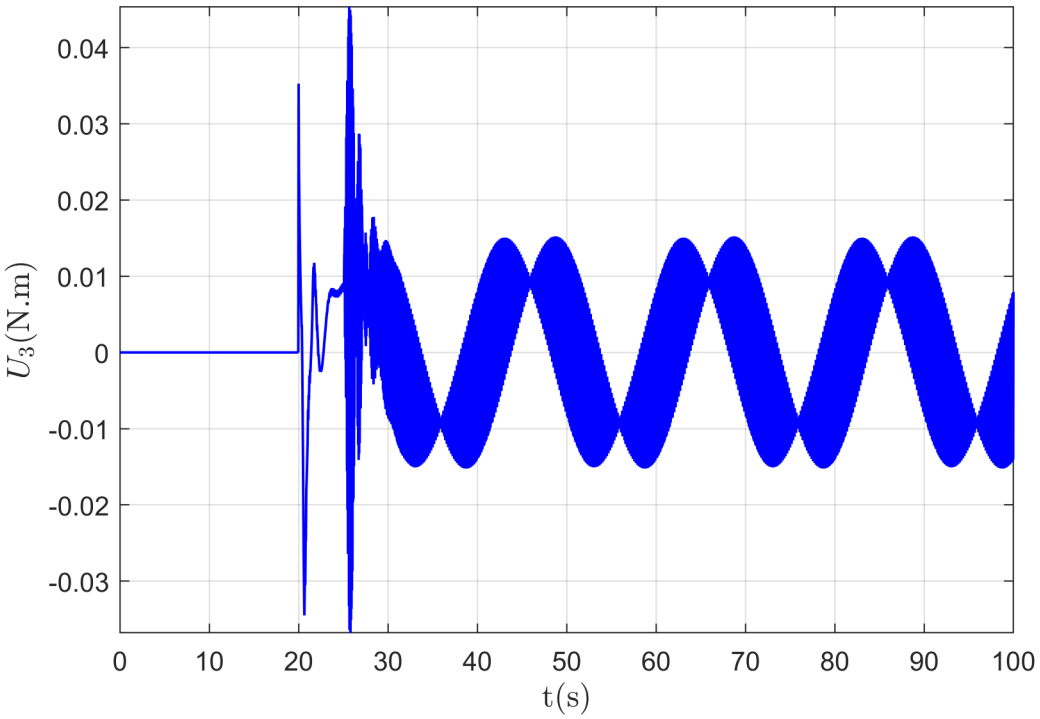

Figure 5.9: ABPSMC Control Input: U_3

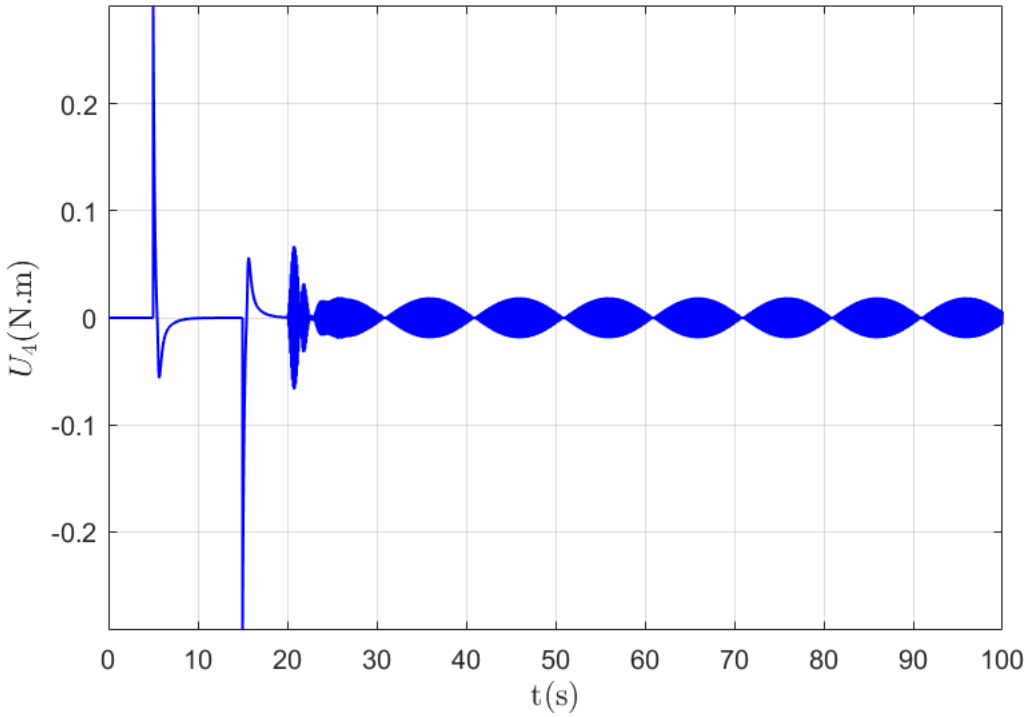

Figure 5.10: ABPSMC Control Input: U_4

5.4 Simulation Results of Adaptive Levenberg-Marquardt SMC (ALMSMC)

The LM optimization scheme is implemented on a feedforward network. The network has one hidden layer with 16 neurons. Combination coefficient μ has the value of 0.01, and the maximum number of epochs is 1000. Moreover, it could also stop if the gradient fell below 1e-6. The total dataset comprises 1 million samples, with two input features (error and error dot), and one output feature: Adaptive Modulation Gain. The dataset was split into three portions: a 70 percent training set, a 15 percent testing set, and a 15 percent validation set. A total of six neural networks are trained, with one independent NN trained for each state (x, y, z, roll, pitch, yaw). The output dataset for each NN training is generated from the respective adaptive law equations, defined by Equation 4.71. Similarly, the input dataset for each NN is tracking error of state and its derivative, denoted by e and $\dot{e}$, respectively.

5.4.1 Simulation Results

Similar to SMC, the performance of the Feedforward NN using the LM optimization scheme is also evaluated in Simulink/Matlab. The system parameters are the same as those used in the SMC evaluation. The initial conditions and simulation settings are identical to those used in the SMC evaluation.

The graph in Figure 5.11 shows the performance of the Feedforward NN, trained using the LM optimization scheme for longitudinal position control of quadrotor UAV over a 100-second time frame window. The solid red line represents the desired longitudinal position, while the dotted black line represents the actual longitudinal position achieved by the quadrotor. As depicted by the plot, the error between the actual and desired trajectories is minimal. The actual position trajectory is almost identical to the desired position throughout the entire time frame, indicating the controller's high accuracy and robustness. Furthermore, no phase lag is found, and overshoot is nearly zero. Overall, the system remains stable, no divergence, oscillation buildup, or drift is observed.

The graph in Figure 5.12 shows the performance of the Feedforward NN, trained using the LM optimization scheme for lateral position control of quadrotor UAV over a 100-second time frame window. The solid red line represents the desired lateral position, while the dotted black line represents the actual lateral position achieved by the quadrotor. As depicted by the plot, the error between the actual and desired trajectories is minimal. The actual position trajectory is almost identical to the desired position throughout the entire time frame, indicating the controller's high accuracy and robustness. Furthermore, no phase lag is found, and overshoot is nearly zero. Overall, the system remains stable, no divergence, oscillation buildup, or drift is observed. The plot in Figure 5.13 represents the performance of yaw angle control for a quadrotor UAV. The solid red line represents the desired yaw angle, while the dotted black line represents the actual yaw angle attained by the quadrotor. The yaw angle is traced perfectly by SMC. No delay, overshoot, oscillation buildup, or drift is observed at any point in the simulation.

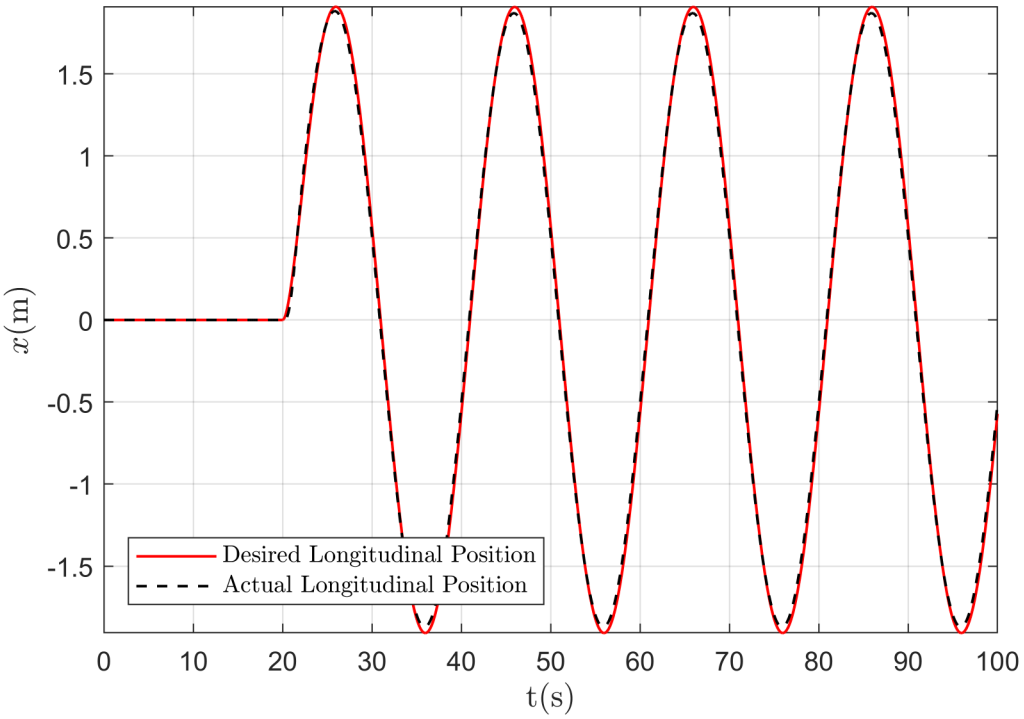

Figure 5.11: Longitudinal Position Control Using ALMSMC

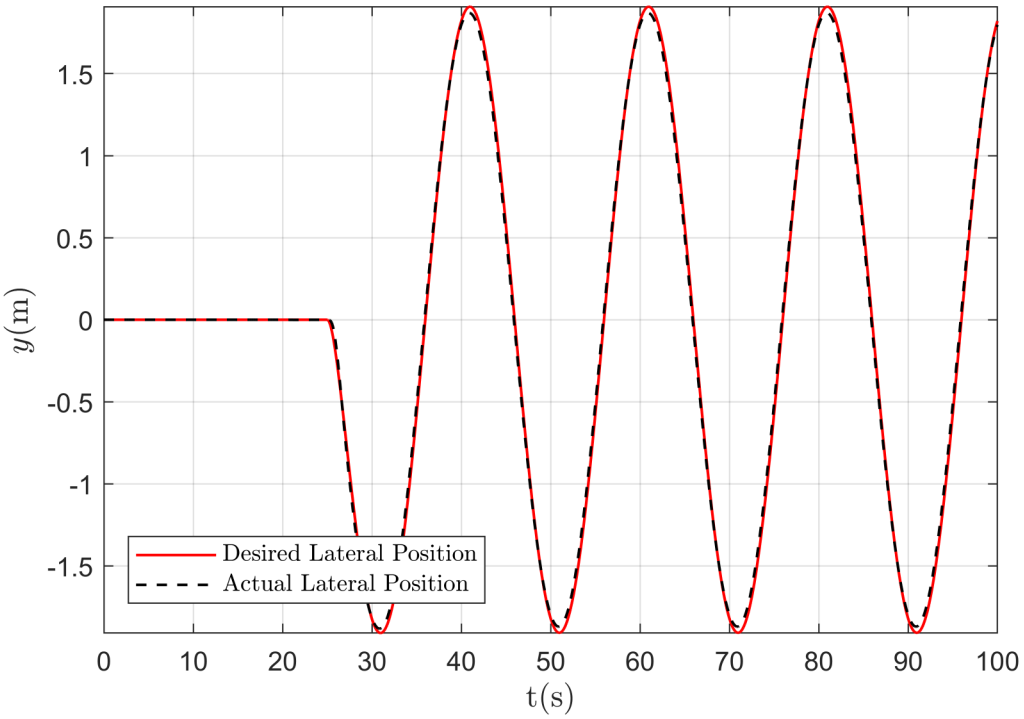

Figure 5.12: Lateral Position Control Using ALMSMC

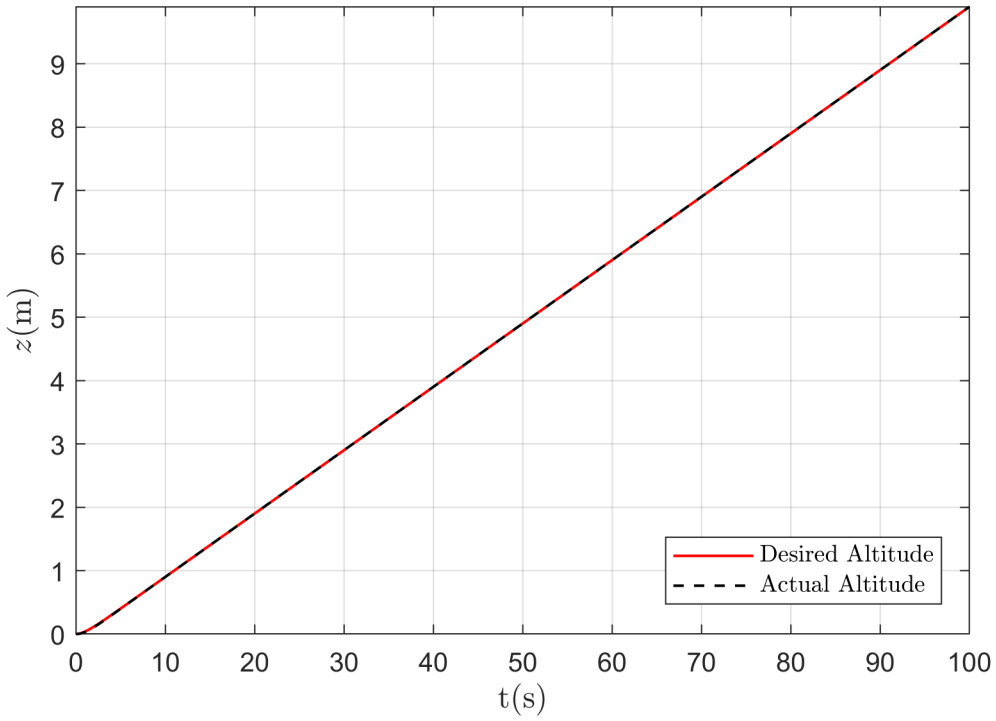

Figure 5.13: Altitude Control Using ALMSMC

The plot in Figure 5.14 represents the performance of yaw angle control for a quadrotor UAV. The solid red line represents the desired yaw angle, while the dotted black line represents the actual yaw angle attained by the quadrotor. The yaw angle is traced perfectly by SMC. No delay, overshoot, oscillation buildup, or drift is observed at any point in the simulation.

The 3D trajectory plot in Figure 5.15 depicts the performance of quadrotor UAV tracking in a helical/spiral path in the x,y,z (3D) space. The solid red line shows the desired trajectory, while the dotted black line illustrates the actual trajectory followed by the UAV. From the plot in Figure 5.15, it is evident that the quadrotor has followed the desired 3D trajectory accurately. No delay, overshoot, oscillation buildup, or drift is observed at any point in the simulation.

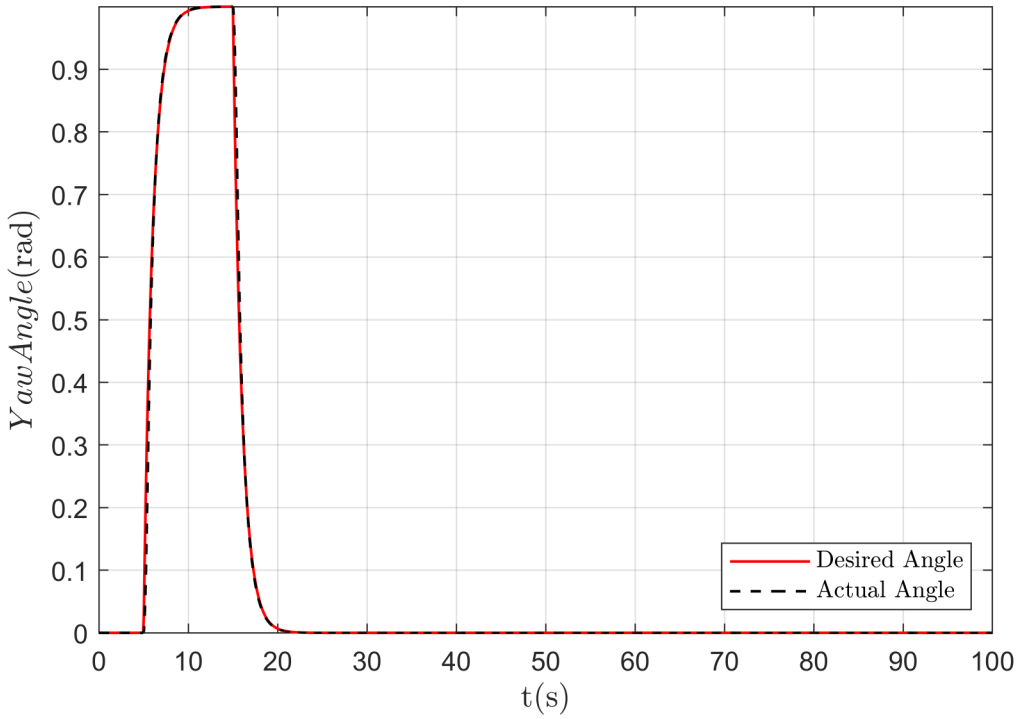

Figure 5.14: Yaw Angle Control Using ALMSMC

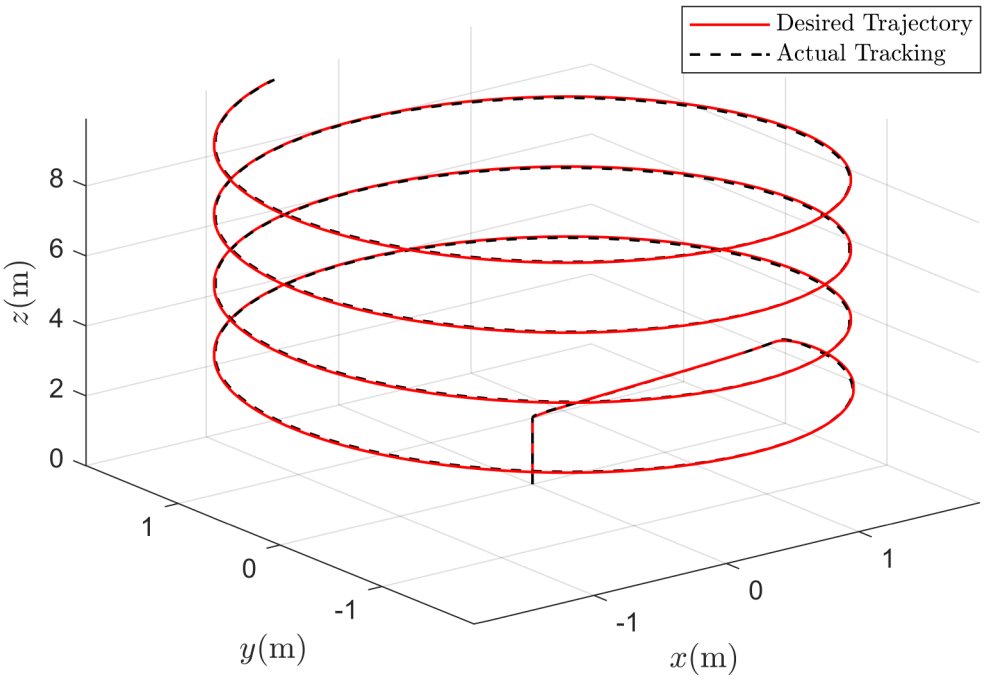

Figure 5.15: 3D Maneuvering of Quadrotor UAV using ALMSMC

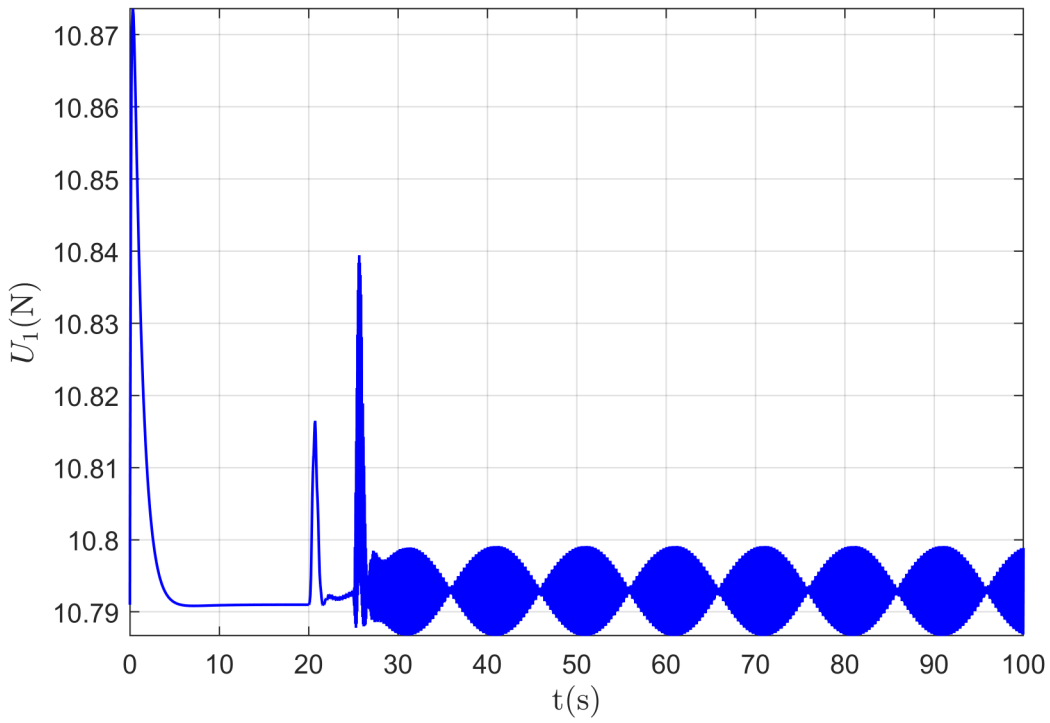

Figure 5.16: ALMSMC Control Input: U_1

Plot in Figure 5.25 represents the control input U_1, which is the total trust generated by the rotors to counteract gravity and control vertical motion (altitude). As shown in Figure, initial spike of approximately $10.88N$ is required to counteract gravity ($m = 1.1\,\text{kg}$ x $g = 9.81\,\text{m/s}^2 = 10.79\text{N}$) and initiate vertical motion. In the current scenario, the quadrotor is not only ascending vertically but also moving along a circular x-y path, resulting in an overall 3D helical trajectory. Furthermore, a quadrotor is a coupled system, exhibiting a strong coupling between the x, y, roll, and pitch states. At time $t = 20\,\text{sec}$ the desired longitudinal position $x_d(t)$ raises from 0 to a positive value as depicted in Figure 5.11, while, lateral position $y_d(t)$ drops from 0 to a negative value at the same time as can be seen in Figure 5.12. To generate the motions in x & y directions, the quadrotor tilts. As a result of this tilt, the thrust vector is no longer purely vertical. Therefore, more thrust is required to maintain upward acceleration. Once this phase is over, the control input U_1 settles into a periodic oscillation pattern around the nominal thrust level (approximately $10.79\,\text{Nm}$).

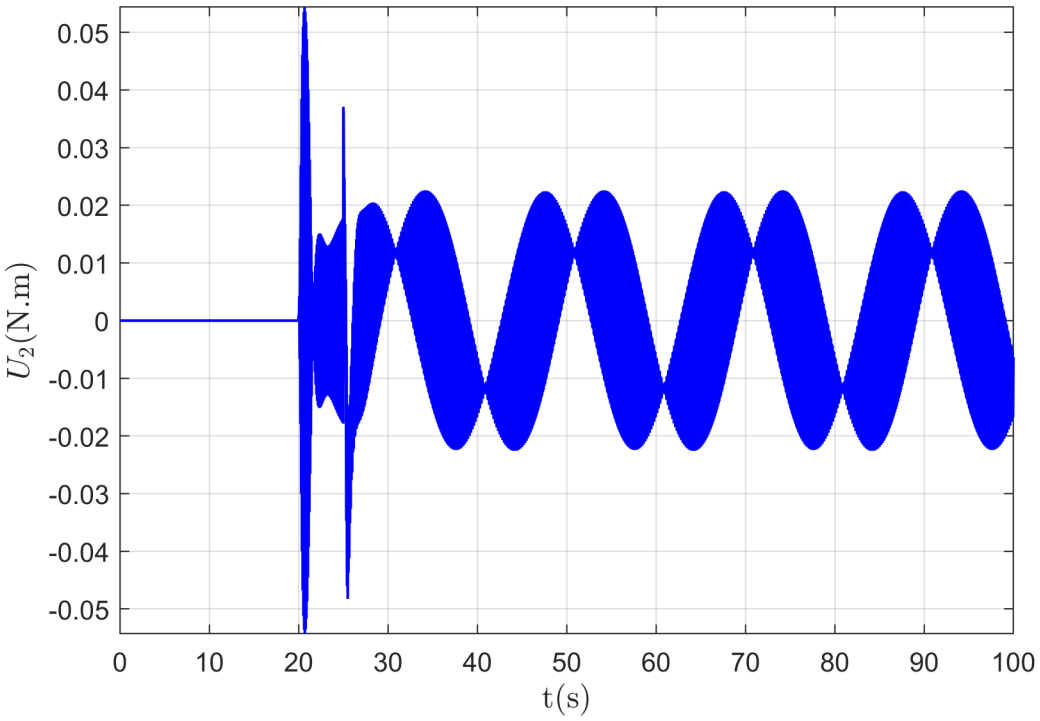

Figure 5.17: ALMSMC Control Input: U_2

Figure 5.26 shows the evolution of control effort U_2 (N.m) with respect to time t (in seconds) for a quadrotor UAV. In quadrotor UAV U_2, typically represents roll torque, which adjusts its orientation about its longitudinal axis. Plot depicts that from $t = 0$ to $t \approx 20$ seconds, the control input U_2 remains 0. It implies that no corrective roll torque is required around the longitudinal axis, as cross verifies from Figure 5.11. At $t = 20$ seconds, sharp control action occurs as the desired longitudinal position rapidly changes at that instant. This indicates the need for immediate torque correction. From $t > 25$ to $t = 100$ seconds, oscillations can be observed which settle into a more periodic manner. These oscillations exhibit a smooth and regular waveform, indicating that periodic oscillations reflect a stable controller response. The waveform's constant amplitude over time in this region suggests that the system is neither drifting nor diverging.

Figure 5.27 shows the evolution of control effort U_3 (N.m) with respect to time t (in seconds) for a quadrotor UAV. In a quadrotor UAV U_3, typically represents pitch torque, which adjusts its orientation about its lateral axis. Plot depicts that

from $t = 0$ to $t \approx 20$ seconds, the control input U_3 remains 0. It implies that no corrective pitch torque is required around the lateral axis, as cross-verified from Figure 5.12. At $t = 20$ seconds, there is a sharp spike in U_3. This indicates the need for immediate torque correction, as the desired lateral position rapidly changes at that instant. From $t > 25$ seconds onward, oscillations can be observed which settle into a more periodic manner. These oscillations indicate the tracking of a sinusoidal trajectory. The periodic oscillations exhibit a smooth and regular waveform, indicating a stable controller response. The waveform's constant amplitude over time in this region suggests that the system is neither drifting nor diverging.

Figure 5.28 shows the evolution of control effort U_4 (N.m) with respect to time t (in seconds) for a quadrotor UAV. In the quadrotor UAV U_4, responsible for yaw control, which adjusts its orientation about the vertical z axis. This affects the UAV's heading. As shown in the Figure, the Initial two positive and negative spikes, each approximately 0.25 N $\cdot$ m in magnitude, within the first 20 seconds, represent a sudden change in the yaw setpoint. This rapid change in yaw angle control can be counterchecked from Figure 5.14. It can be seen that the desired yaw angle abruptly changes from 0 to 1 radian and back to 0 radian over the first 20 seconds. To meet the yaw-angle target, the control input U_4 is applied. Time from $t \approx 20$ onward, the control input U_4 remains stable with small periodic oscillations centered around 0 N.m. These minor yaw corrections indicate that the UAV maintains a stable heading while exhibiting only minor periodic motions, possibly due to coupled roll or pitch motions.

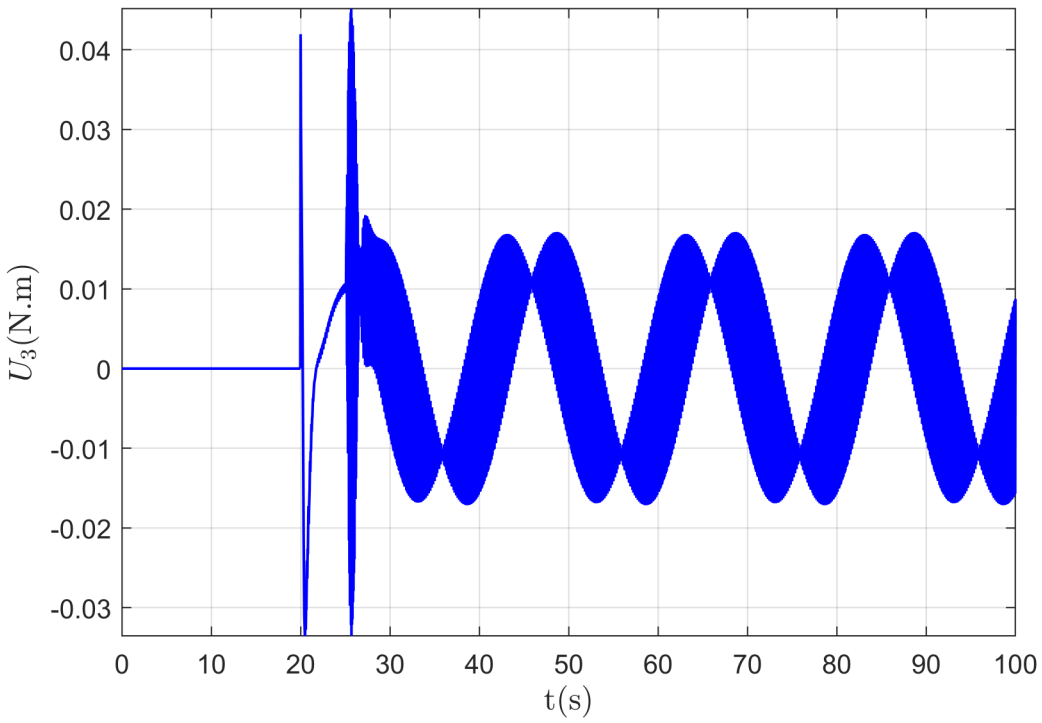

Figure 5.18: ALMSMC Control Input: U_3

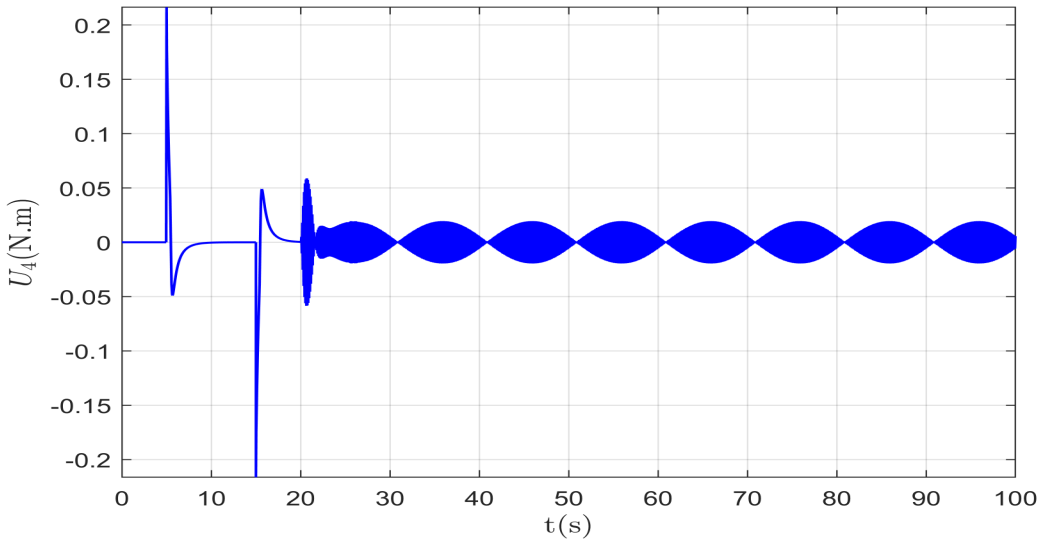

Figure 5.19: ALMSMC Control Input: U_4

5.5 Simulation Results of Adaptive Radial Basis Function Sliding Model Control (ARBFSMC)

Similar to SMC, the performance of the ARBFSMC scheme is also evaluated in Simulink/Matlab. The system parameters are the same as those used in the SMC evaluation.

The graph in Figure 5.20 shows the performance of the RBFNN scheme for longitudinal position control of quadrotor UAV over a 100-second time frame window. The solid red line represents the desired longitudinal position, while the dotted black line represents the actual longitudinal position achieved by the quadrotor.

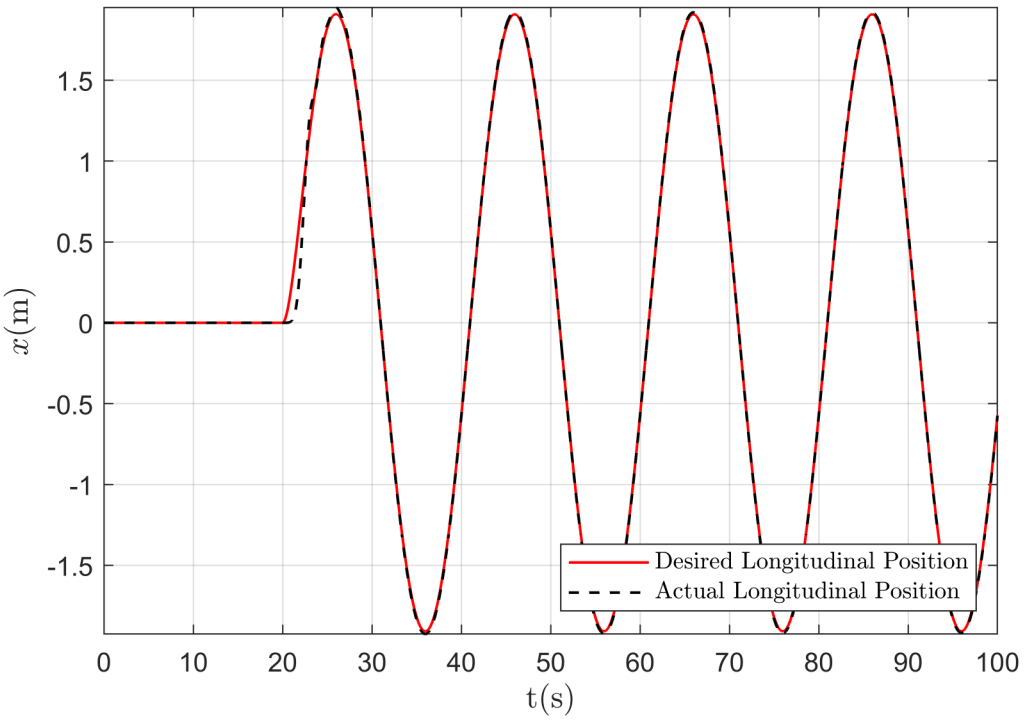

Figure 5.20: Longitudinal Position Control Using ARBFSMC

As depicted by the plot, for the time interval from $t = 0$ to $t = 20$ seconds, the desired trajectory remains constant at 0. From $t > 20$ seconds onward, as the desired trajectory changes, the actual trajectory initially shows a slight lag. This is due to system dynamics (response delay). After a short transient, the actual trajectory

closely matches the desired trajectory throughout the entire time frame, indicating the controller's high accuracy and robustness. Furthermore, no phase lag is found, and overshoot is nearly zero. Overall, the system remains stable, with no divergence, oscillation buildup, or drift observed.

The graph in Figure 5.21 shows the performance of the RBFNN scheme for lateral position control of quadrotor UAV over a 100-second time frame window. As depicted by the plot, the error between the actual and desired trajectories is minimal. The actual position trajectory is almost identical to the desired position throughout the entire time frame, indicating the controller's high accuracy and robustness. Furthermore, no phase lag is found, and overshoot is nearly zero. Overall, the system remains stable, with no divergence, oscillation buildup, or drift observed.

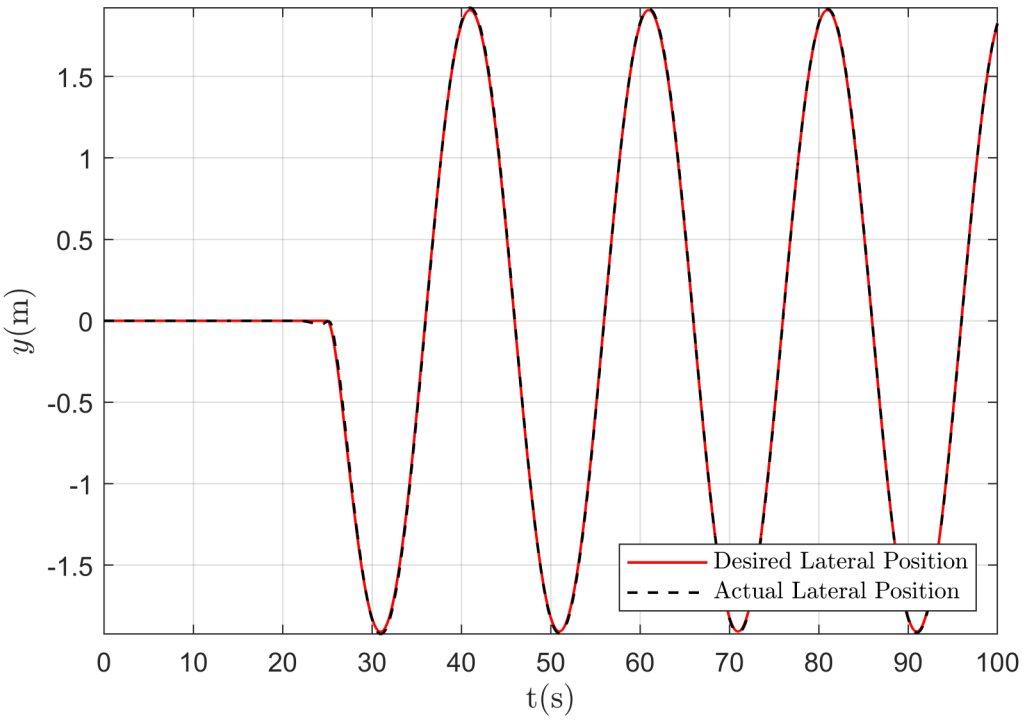

Figure 5.21: Lateral Position Control Using ARBFSMC

The plot in Figure 5.22 represents the performance of the RBFNN scheme-based altitude controller. The desired altitude is achieved perfectly. No delay, overshoot, oscillation buildup, or drift is observed at any point in the simulation.

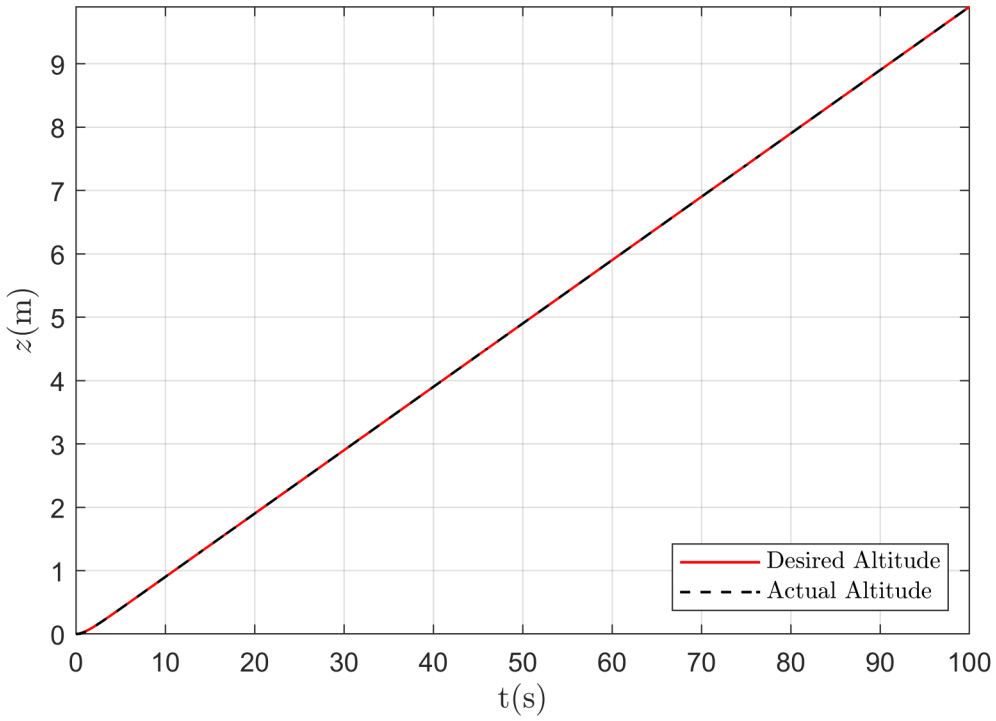

Figure 5.22: Altitude Control Using ARBFSMC

The plot in Figure 5.23 represents the performance of yaw angle control for a quadrotor UAV. The yaw angle is traced perfectly by SMC. No delay, overshoot, oscillation buildup, or drift is observed at any point in the simulation.

The 3D trajectory plot in Figure 5.24 depicts the performance of quadrotor UAV tracking in a helical/spiral path in the x,y,z (3D) space. The solid red line shows the desired trajectory, while the dotted black line illustrates the actual trajectory followed by the UAV. From the plot in Figure 5.24, it is evident that the quadrotor has followed the desired 3D trajectory accurately. No delay, overshoot, oscillation buildup, or drift is observed at any point in the simulation.

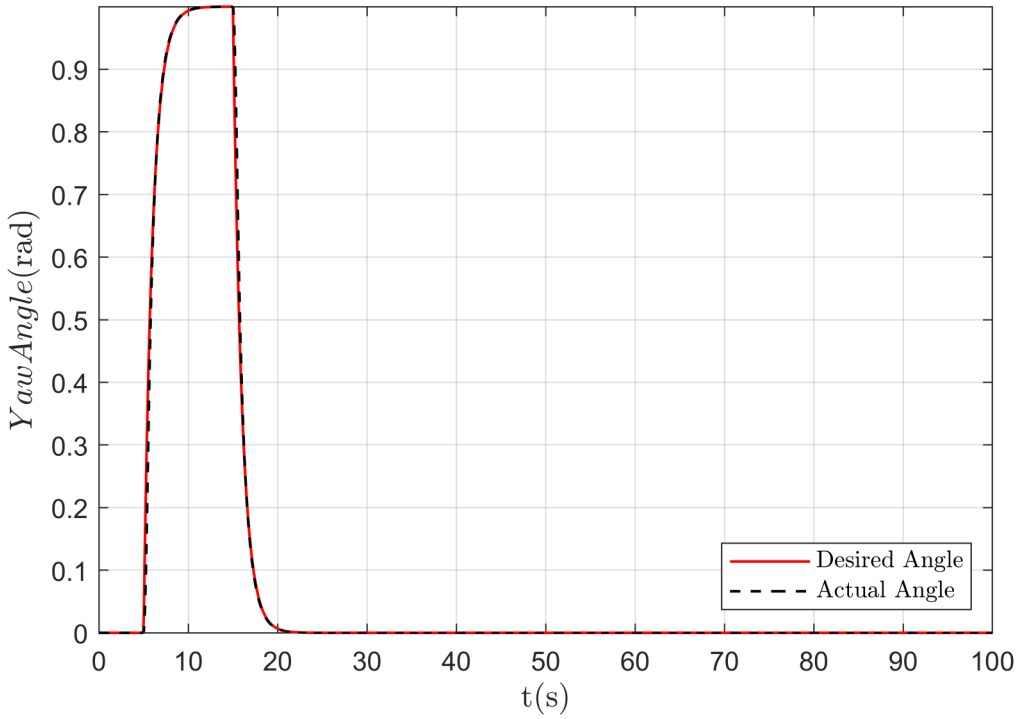

Figure 5.23: Yaw Angle Control Using ARBFSMC

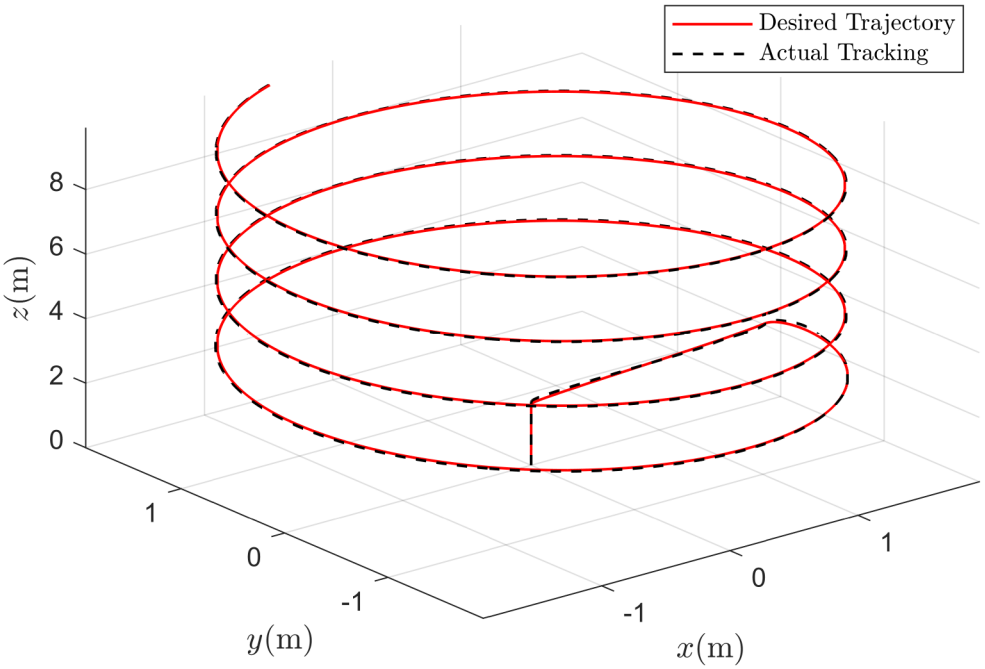

Figure 5.24: 3D Maneuvering of Quadrotor UAV Using ARBFSMC

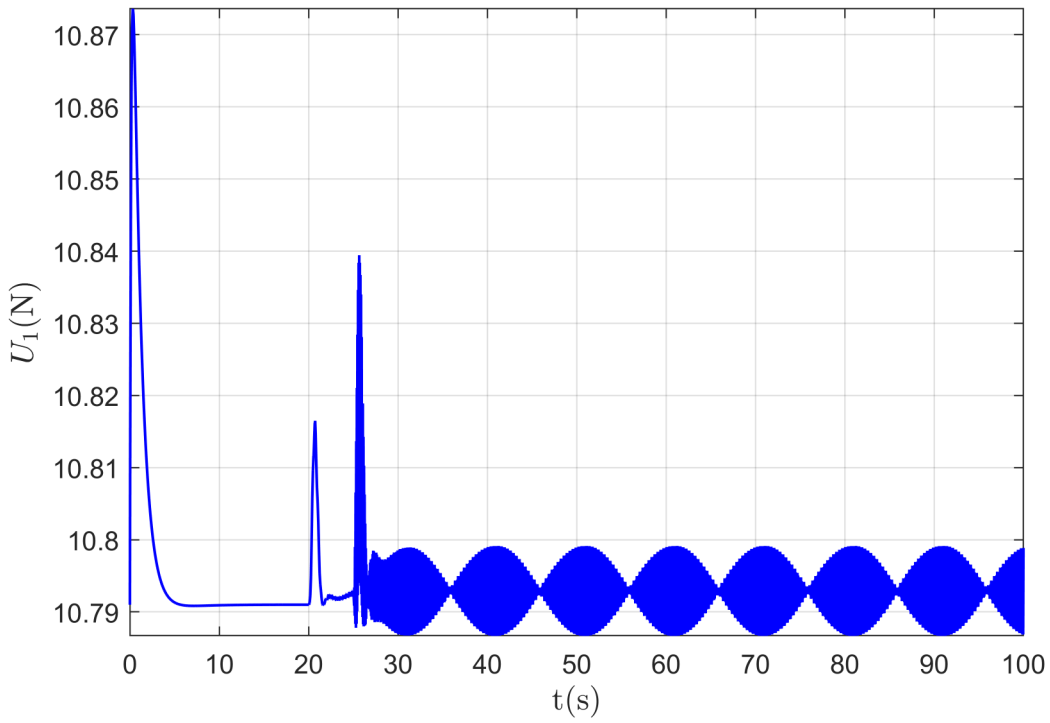

Figure 5.25: ARBFSMC Control Input: U_1

Plot in Figure 5.25 represents the control input U_1, which is the total trust generated by the rotors to counteract gravity and control vertical motion (altitude). As shown in Figure, initial spike of approximately 10.88N is required to counteract gravity ($m = 1.1\,\text{kg}$ x $g = 9.81\,\text{m/s}^2 = 10.79\text{N}$) and initiate vertical motion. In the current scenario, the quadrotor is not only ascending vertically but also moving along a circular x-y path, resulting in an overall 3D helical trajectory. Furthermore, a quadrotor is a coupled system, exhibiting a strong coupling between the x, y, roll, and pitch states. At time $t = 20\,\text{sec}$, the desired longitudinal position $x_d(t)$ changes from 0 to a positive value as depicted in Figure 5.20. Similarly, the lateral position $y_d(t)$ drops from 0 to a negative value at $t = 25\,\text{sec}$, the same can be seen in Figure 5.21. To generate the motions in x & y directions, the quadrotor tilts. As a result of this tilt, the thrust vector is no longer purely vertical. Therefore, more thrust is required to maintain upward acceleration. Once this phase is over, the control input U_1 settles into a periodic oscillation pattern around the nominal thrust level (approximately 10.79 N).

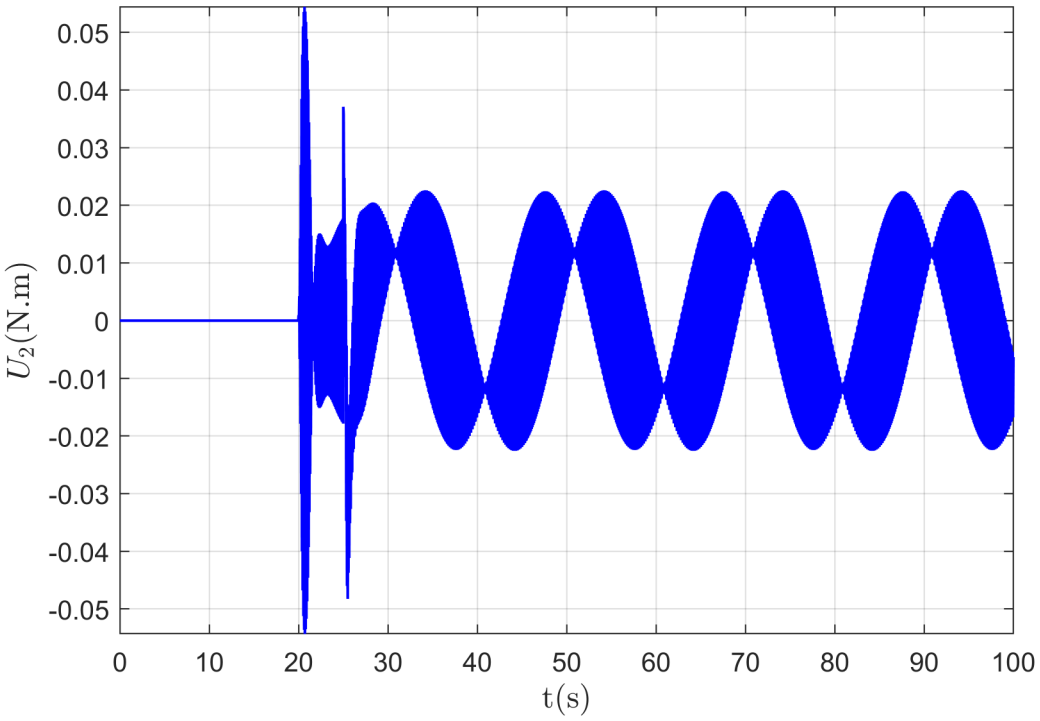

Figure 5.26: ARBFSMC Control Input: U_2

Figure 5.26 shows the evolution of control input U_2 (N.m) with respect to time t (in seconds) for a quadrotor UAV. In quadrotor UAV U_2, typically represents roll torque, which adjusts its orientation about its longitudinal axis. Plot depicts that from $t = 0$ to $t \approx 20$ seconds, the control input U_2 remains 0. It implies that no corrective roll torque is required around the longitudinal axis, as cross verifies from Figure 5.11. At $t = 20$ seconds, there is a sharp spike in U_2. This indicates the need for immediate torque correction, as the desired longitudinal position rapidly changes at that instant. From $t > 25$ to $t = 100$ seconds, oscillations can be observed which settle into a more periodic manner. These oscillations indicate the tracking of a sinusoidal trajectory. The periodic oscillations exhibit a smooth and regular waveform, indicating a stable controller response. The waveform's constant amplitude over time in this region suggests that the system is neither drifting nor diverging.

Figure 5.27 shows the evolution of control effort U_3 (N.m) with respect to time t (in seconds) for a quadrotor UAV. In a quadrotor UAV U_3, typically represents pitch torque, which adjusts its orientation about its lateral axis. Plot depicts that from

$t = 0$ to $t \approx 20$ seconds, the control input U_3 remains 0. It implies that no corrective pitch torque is required around the lateral axis, as cross-verified from Figure 5.12. At $t = 20$ seconds, there is a sharp spike in U_3. This indicates the need for immediate torque correction, as the desired lateral position rapidly changes at that instant. From $t > 25$ seconds onward, oscillations can be observed which settle into a more periodic manner. These oscillations indicate the tracking of a sinusoidal trajectory. The periodic oscillations exhibit a smooth and regular waveform, indicating a stable controller response. The constant amplitude of the waveform over time in this region suggests that the system is neither drifting nor diverging.

Figure 5.28 shows the evolution of control effort U_4 (N.m) with respect to time t (in seconds) for a quadrotor UAV. In the quadrotor UAV U_4, responsible for yaw control, which adjusts its orientation about the vertical z axis. This affects the UAV's heading. As shown in the Figure, the Initial two positive and negative spikes, each approximately 0.25 N $\cdot$ m in magnitude, within the first 20 seconds, represent a sudden change in the yaw setpoint. This rapid change in yaw angle control can be counterchecked from Figure 5.14. It can be seen that the desired yaw angle abruptly changes from 0 to 1 radian and back to 0 radian over the first 20 seconds. To meet the yaw angle target, the control input U_4 is applied. Time from $t \approx 20$ onward, the control input U_4 remains stable with small periodic oscillations centered around 0 N.m. These minor yaw corrections indicate that the UAV maintains a stable heading while exhibiting only minor periodic motions, possibly due to coupled roll or pitch motions.

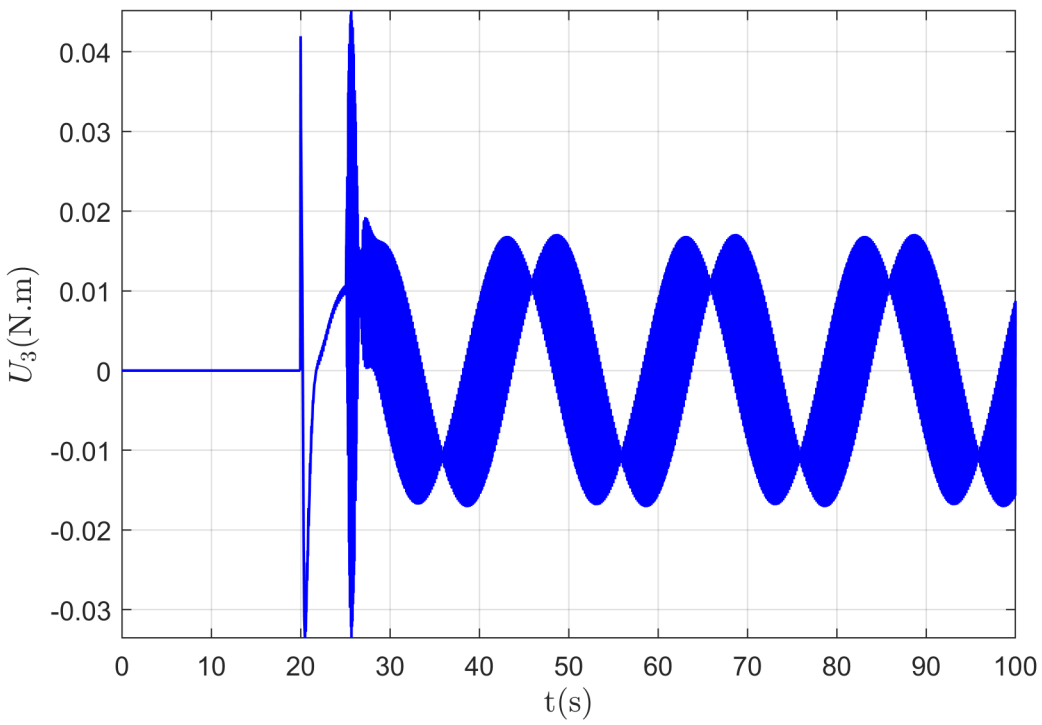

Figure 5.27: ARBFSMC Control Input: U_3

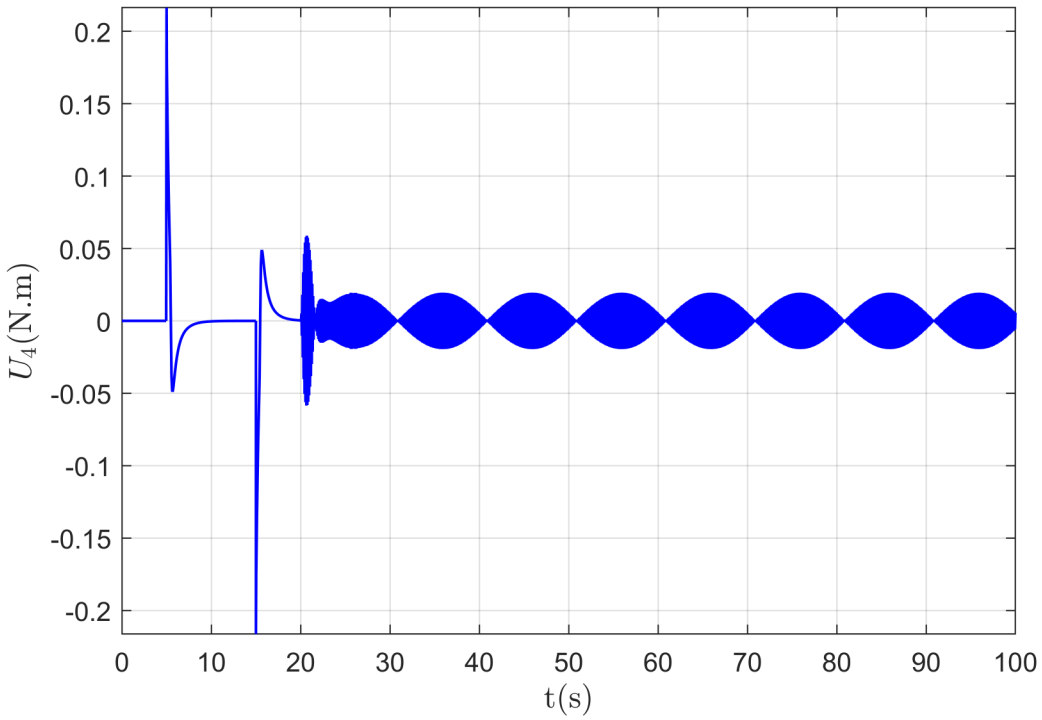

Figure 5.28: ARBFSMC Control Input: U_4

5.6 Adaptve Gains

The Figure 5.29 illustrates the time responses of the nonlinear adaptive gains obtained from the ARBFSMC scheme to mitigate the effects of lumped disturbances. The adaptive gains K_1 and K_2 regulate the longitudinal and lateral translational dynamics, respectively, while K_3 governs the altitude position. The rotational dynamics about the roll, pitch, and yaw axes are controlled by the adaptive gains K_4, K_5, and K_6, respectively.

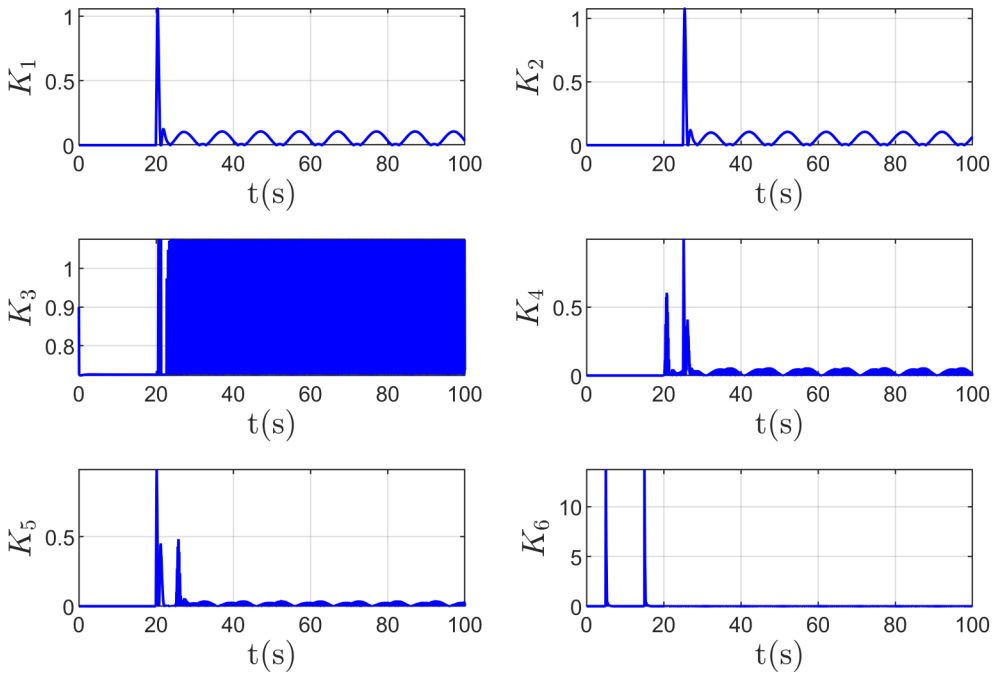

Figure 5.29: Adaptive Gains

Summary

This chapter presents adaptive NN–based control schemes for nonlinear gain estimation within the SMC framework. Three NN-based controllers, ABPSMC, ALMSMC, and ARBFSMC, are employed to estimate the adaptive nonlinear gains online. Extensive simulations are conducted for each controller to evaluate its longitudinal, lateral,

altitude, roll, pitch, and yaw control performance. The results demonstrate improved tracking accuracy, smoother system response, reduced chattering amplitude, and reduced control effort compared to conventional nonlinear control schemes.

Chapter 6
SMC and NN-Based Controller Comparison

The Figure 6.1 represents the performance of longitudinal position control for four different control schemes. One of them is the conventional SMC, which has fixed gains, while the other three are NN-based schemes. Plots in the Figure confirm the effectiveness of neural network–based controllers in improving adaptive SMC modulation gain optimization. The SMC exhibits the most significant tracking deviations/errors, particularly near the peaks of the sinusoidal input, due to the high fixed gains used to address uncertain disturbances. In contrast, ABPSMC and ALMSMC exhibit greater tracking precision than SMC. ALMSMC offers balanced, stable performance. The ARBFSMC provides the closest approximation to the reference trajectory, thereby validating its superior accuracy, though a slight overshoot is observed in some instances. These results corroborate the quantitative error analysis presented in Table 6.1.

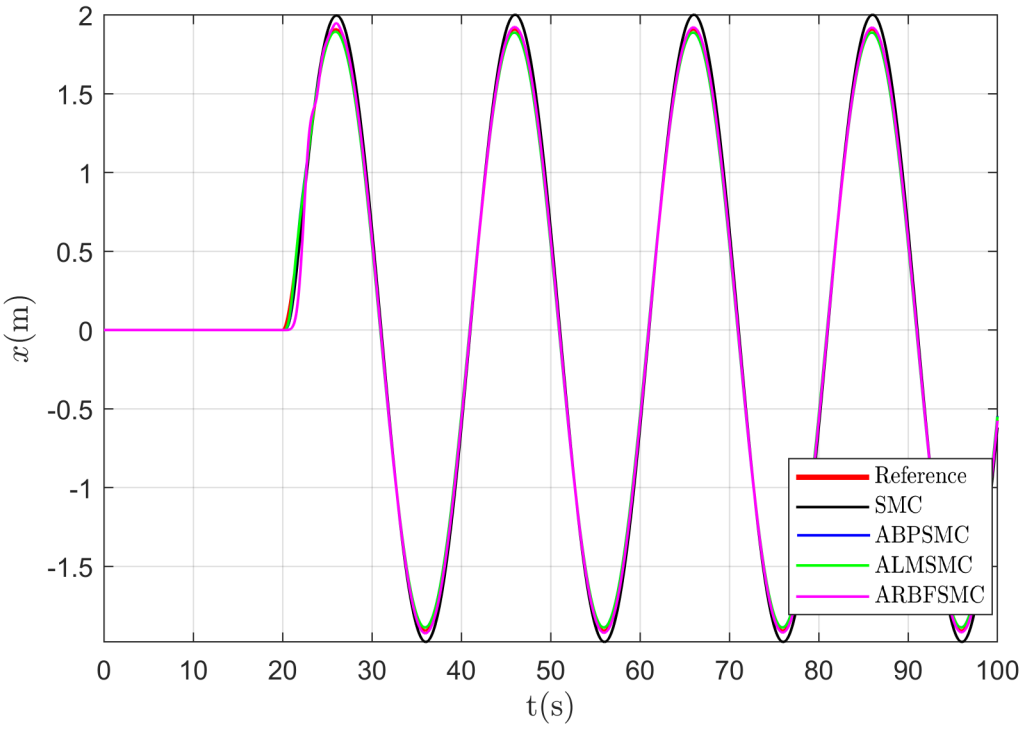

Figure 6.1: Longitudinal Position Control

The Figure 6.2 represents the performance of lateral position control for four different control schemes. One of them is the conventional SMC, which has fixed gains, while the other three are NN-based schemes. Plots in the Figure confirm the effectiveness of neural network–based controllers in improving adaptive SMC modulation gain optimization. The SMC exhibits the most significant tracking deviations/errors, particularly near the peaks of the sinusoidal input, due to the high fixed gains used to address uncertain disturbances. In contrast, ABPSMC and ALMSMC exhibit greater tracking precision than SMC. ALMSMC offers balanced, stable performance. The ARBFSMC provides the closest approximation to the reference trajectory, thereby validating its superior accuracy, though a slight overshoot is observed in some instances. These results validate the quantitative error analysis presented in Table 6.1.

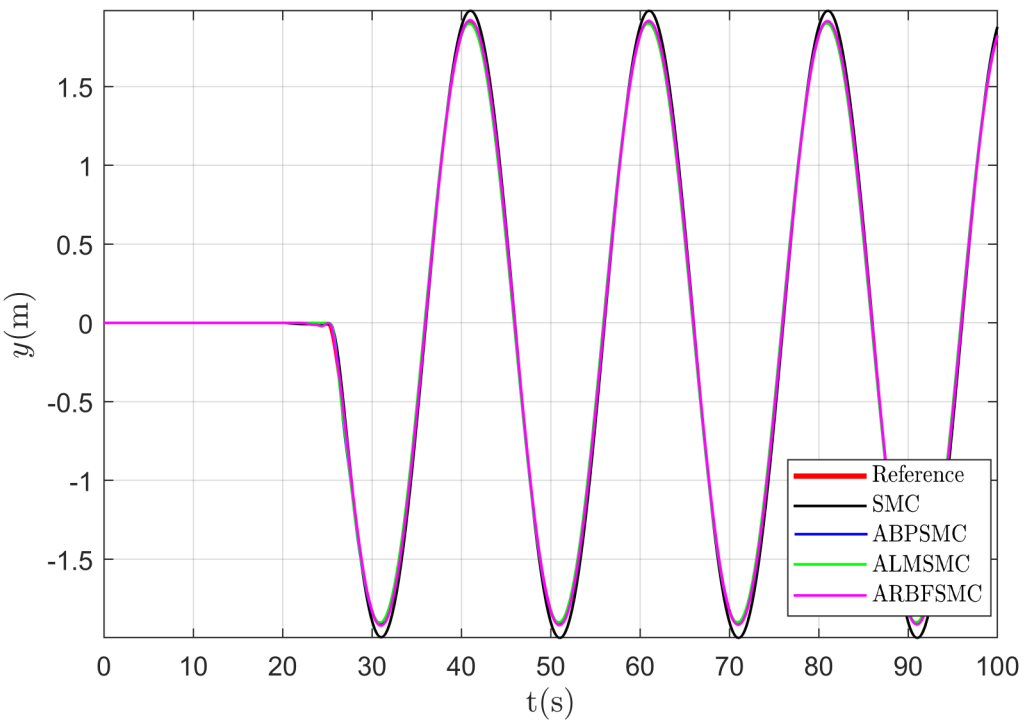

Figure 6.2: Lateral Position Control

Figure 6.3 illustrates the altitude tracking performance of various control strategies, including SMC, ABPSMC, ALMSMC, and ARBFSMC. Altitude tracking of these control schemes is compared to the Reference trajectory. From the plot, it is evident that all four controller schemes successfully achieved the desired altitude perfectly. No delay, overshoot, oscillation buildup, or drift is observed at any point in the simulation. The graph almost overlaps with the Reference.

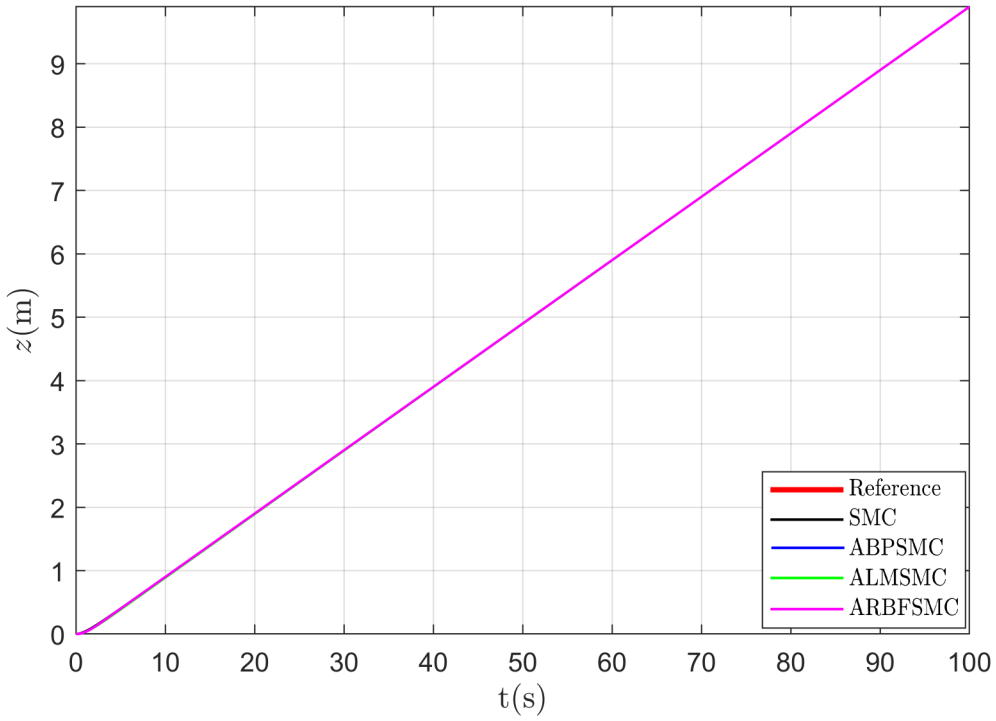

Figure 6.3: Altitude Control

Figure 6.4 shows the yaw angle response for various control strategies against the Reference trajectory. All controllers accurately followed the reference command, indicating effective tracking performance and robustness. Among these, SMC achieves good accuracy but exhibits relatively sharper control action, while ABPSMC and ALMSMC provides smoother responses with reduced fluctuations. The ARBFSMC performed the best among all control schemes, maintaining the closest tracking to the Reference with minimal error and superior smoothness.

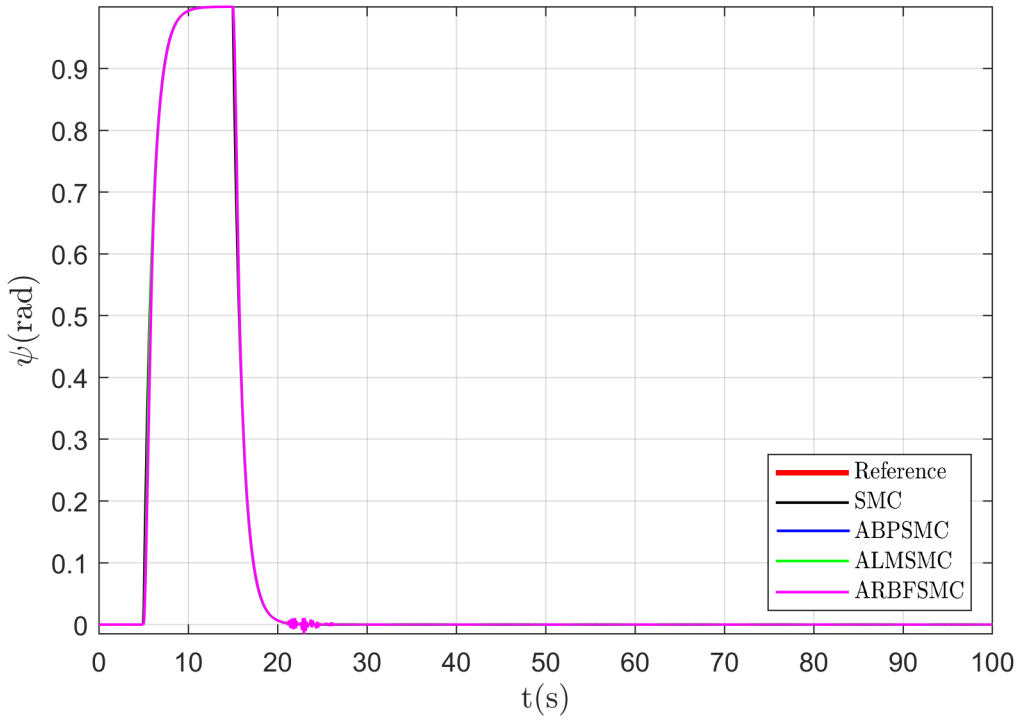

Figure 6.4: Yaw Angle Control

The Figure 6.5 presents the control input response U_1 generated by SMC, ABPSMC, ALMSMC, and ARBFSMC over a simulation time of 100 seconds. Initially, all controllers exhibit an adjustment phase, stabilizing near the operating point of approximately 10.8 N. The SMC response exhibits oscillatory behavior with relatively high amplitude chattering, a well-known drawback of classical SMC. In contrast, three NN-based controllers, ABPSMC, ALMSMC, and ARBFSMC, effectively reduced chattering. Among these, ABPSMC and ALMSMC provide smoother control efforts after the transient phase. However, ARBFSMC introduces some high spikes in the early stage (around 20–30 seconds) before settling into an oscillatory pattern. Overall, the results indicate that NN-based adaptive SMC strategies can mitigate the excessive chattering typically associated with classical SMC, thereby achieving smoother control inputs. However, ARBFSMC exhibits higher transient fluctuations compared to ABPSMC and ALMSMC, suggesting a trade-off between learning capability and transient robustness.

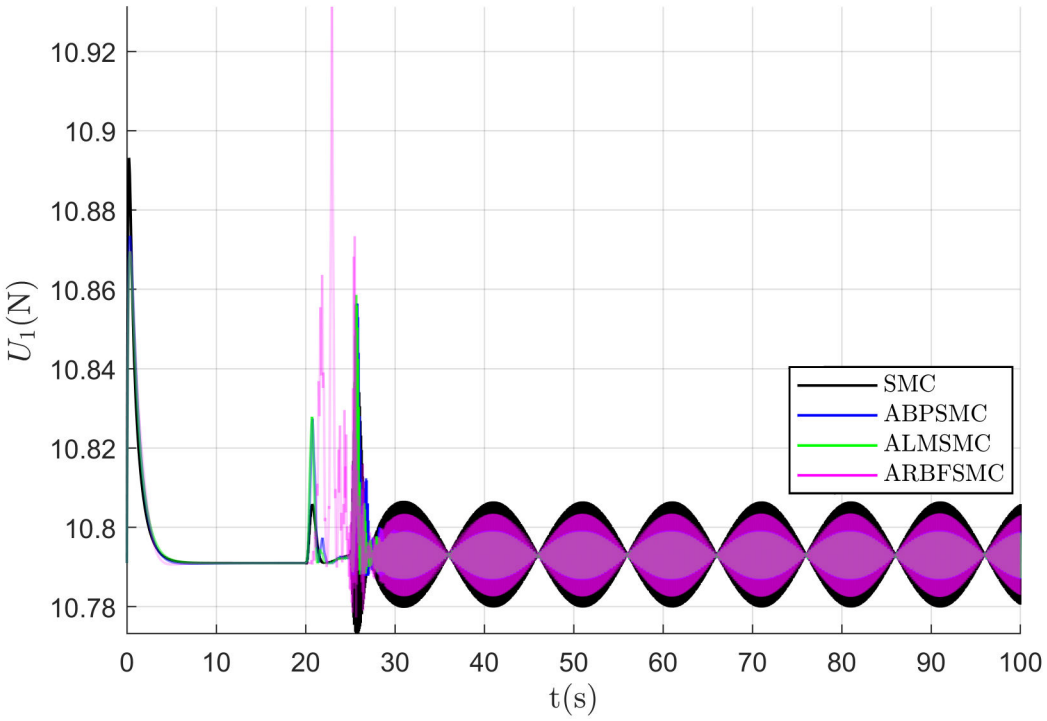

Figure 6.5: Control Input: U_1

The Figure 6.6 shows the control input U_2 generated by classical SMC, and three ASMC schemes, namely ABPSMC, ALMSMC, ARBFSMC, for adaptive modulation gain optimization, over a simulation time of 100 seconds. The classical SMC exhibits high-amplitude oscillations and persistent chattering throughout the simulation, with the torque oscillating between 2.5 and 3 $\mathrm{N} \cdot \mathrm{m}$ after 20 seconds. This behavior is due to fixed controller gains in SMC, which enforce a high control effort even when the external disturbances disappear or when no significant correction in roll and pitch is required, thereby reducing efficiency and increasing energy consumption. In contrast, the neural network–based controllers produce smoother control signals with significantly reduced amplitudes. Both ABPSMC and ALMSMC generate negligible torque demand, remaining close to zero after the transient phase, highlighting their efficiency and stability. The ARBFSMC, while maintaining a much smaller amplitude than SMC, exhibits periodic oscillations and transient fluctuations between 20–30 seconds before stabilizing into a lower-amplitude pattern. These results demonstrate that integrating neural networks with SMC effectively suppresses the chattering

phenomenon and reduces control effort, thereby improving actuator efficiency. Among the tested strategies, ABPSMC and ALMSMC yield the smoothest and most stable control responses, whereas ARBFSMC achieves acceptable performance with some additional oscillatory behavior.

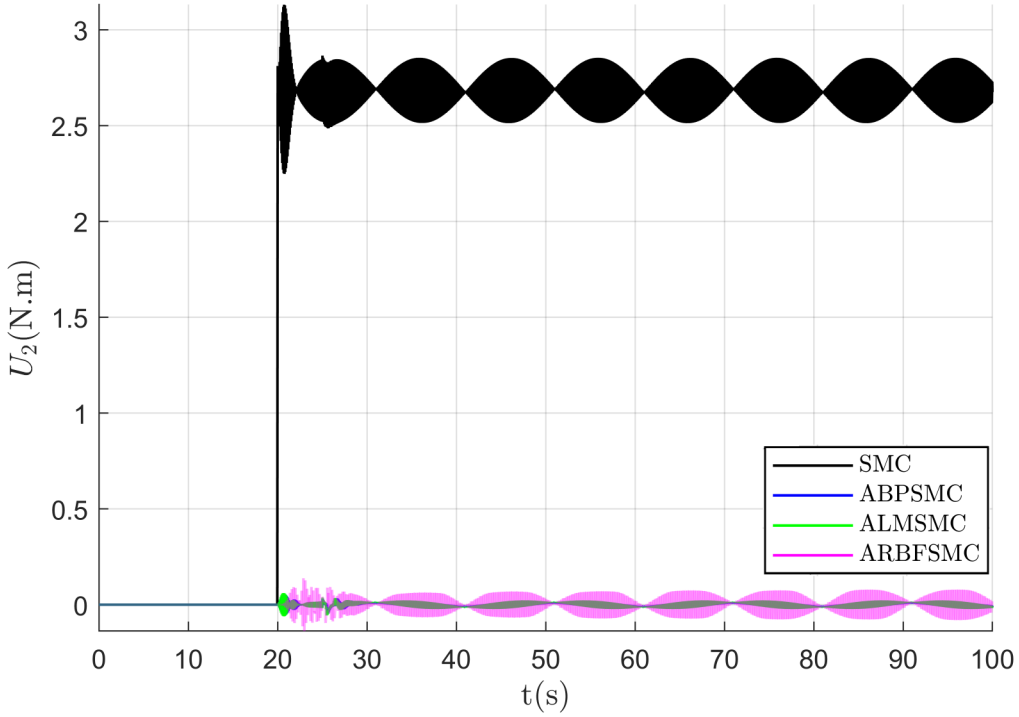

Figure 6.6: Control Input: U_2

This Figure 6.7 illustrates the control input response of U_3 under different control strategies. U_3 corresponds to the pitch torque. At the moment time $t = 25$ seconds, the classical SMC produces a very high control effort, resulting in a sharp spike in U_3. This indicates the need for immediate torque correction, as the desired lateral position rapidly changes at that instant. From $t > 25$ seconds onward, oscillations can be observed which settle into a more periodic manner. These oscillations indicate the tracking of a sinusoidal trajectory. The periodic oscillations exhibit a smooth and regular waveform, indicating a stable controller response. Moreover, due to the fixed gains of SMC, the torque remains high and oscillatory even when the disturbance vanishes or no further corrective yaw action is required. This results in

unnecessary actuator activity and increased energy consumption. On the other hand, the neural network–based approaches demonstrate significantly smoother control responses. Both the ABPSMC and ALMSMC yields stable torque outputs with negligible oscillations, ensuring efficient yaw regulation without imposing excessive actuator demand. The ARBFSMC response also shows improved behavior compared to SMC, though with small-amplitude fluctuations. Overall, these results confirm that neural network–augmented controllers are more effective at reducing chattering and maintaining control efficiency than conventional SMC.

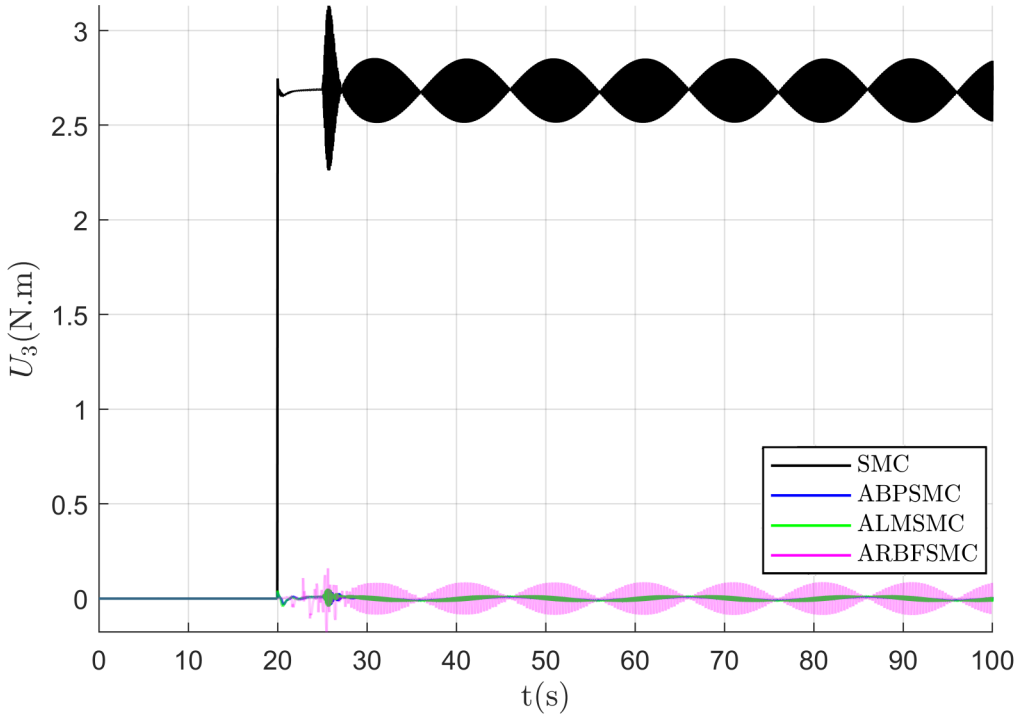

Figure 6.7: Control Input: U_3

The Figure 6.8 shows the control input U4, corresponding to the yaw torque, generated by the classical SMC, and three NN-based ASMC schemes (ABPSMC, ALMSMC, ARBFSMC) for adaptive modulation-gain optimization over a simulation time of 100 seconds. The yaw reference signal is a step command lasting 5 to 15 seconds. To follow the reference, conventional SMC exhibits sharp and discontinuous spikes in control effort. This behavior reflects the SMC's inherent sensitivity to abrupt

reference transitions and its reliance on fixed gains; consequently, the classical SMC produces high control input spikes even when minimal yaw correction is required or when external disturbances are negligible. As a result, the actuator remains active with an unnecessarily high torque demand, leading to increased energy consumption and potential wear and tear. In contrast, NN-based controllers exhibit substantially smoother, more adaptive control behavior. Both ABPSMC and ALMSMC effectively attenuates the transient oscillations triggered by the step change in yaw reference and exhibits rapid convergence, approximating zero control effort after the reference is deactivated, indicating efficient disturbance rejection and minimal chattering. The ARBFSMC also provides a significant reduction in oscillatory behavior compared to SMC, although minor fluctuations persist during the initial transient. In general, the neural network demonstrates clear superiority in regulating the yaw torque, effectively mitigating chattering, reducing control effort, and enhancing robustness against both reference and disturbance variations.

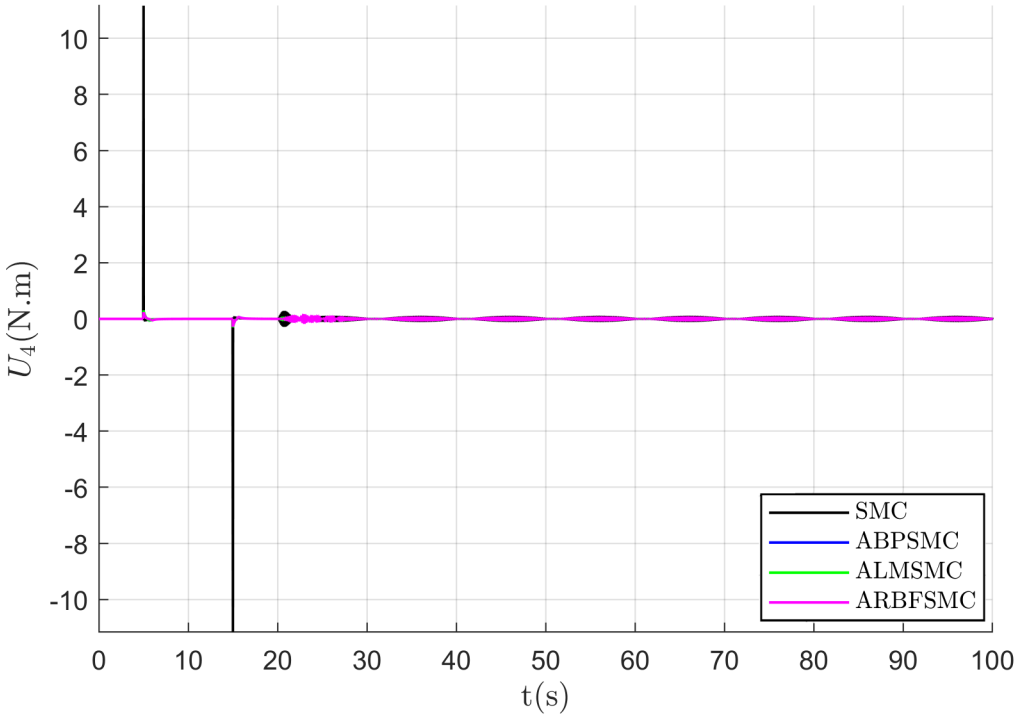

Figure 6.8: Control Input: U_4

The performance comparison of the SMC and the NN-based adaptive SMCs (ABPSMC, ALMSMC, and ARBFSMC) is presented in Table 6.1. The evaluation is performed across the four control axes (X, Y, Z, and Yaw) using four error metrics. Integral Absolute Error (IAE), Integral Time Absolute Error (ITAE), and Root Mean Square Error (RMSE).

Table 6.1: Comparison of SMC and NN–Based Controllers Using Error Indices

Axis	**Controller**	**IAE**	**ITAE**	**RMSE**
X	SMC	4.6284	277.4674	0.0577
	ABPSMC	1.5166	89.4698	0.0193
	ALMSMC	1.3525	80.2496	0.0170
	ARBFSMC	1.2893	53.1388	0.0390
Y	SMC	4.2920	264.8235	0.0552
	ABPSMC	1.8697	117.0607	0.0242
	ALMSMC	1.7117	107.8144	0.0221
	ARBFSMC	0.7324	38.6584	0.0101
Z	SMC	0.0315	0.1138	0.0015
	ABPSMC	0.0392	0.1611	0.0015
	ALMSMC	0.4137	10.7319	0.0056
	ARBFSMC	0.0034	0.0256	0.0002
Yaw	SMC	0.0025	0.0518	0.0002
	ABPSMC	0.1475	2.2409	0.0097
	ALMSMC	0.1284	2.0285	0.0087
	ARBFSMC	0.1812	2.3230	0.0122

For the longitudinal control, the conventional SMC exhibits the highest error values across all metrics, indicating larger tracking deviations and less effective regulation

during translational motion tracking. The integration of neural networks with the adaptive SMC framework significantly improves performance, resulting in substantial reductions in both IAE and ITAE. Among the neural network–based controllers, the ARBFSMC achieves the lowest IAE (1.2893) and ITAE (53.1388), demonstrating its strong nonlinear mapping capability and efficient learning of system dynamics. The ALMSMC and ABPSMC exhibit notable improvements over the conventional SMC, with the ALMSMC performing slightly better than the ABPSMC. Overall, ARBFSMC shows superior tracking accuracy and faster error convergence.

For the lateral control, the conventional SMC again yields the highest error values across all metrics, reflecting greater tracking deviations and slower convergence in lateral motion. The incorporation of neural networks into the adaptive SMC framework significantly improves performance, resulting in substantial reductions in both transient and steady-state errors. Among the neural network–based controllers, the ARBFSMC achieves the lowest IAE (0.7324) and ITAE (38.6584), indicating superior nonlinear approximation capability and smoother control actions. The ALMSMC and ABPSMC also show improved results compared to the conventional SMC, with the ALMSMC performing slightly better than the ABPSMC. Overall, the ARBFSMC exhibits the most precise and stable tracking performance along the Y-axis.

For altitude control, all controllers exhibit high precision due to the relatively decoupled nature of vertical dynamics. However, the ARBFSMC achieves the lowest error across all metrics, with IAE of 0.0034 and ITAE of 0.0256, and RMSE of 0.0002, demonstrating exceptional accuracy and fast convergence. The ALMSMC and ABPSMC controllers perform satisfactorily but exhibit slightly higher ITAE values, indicating a slower decay of residual errors. The conventional SMC maintains acceptable performance but is comparatively less precise than the NN–based controllers in handling vertical motion.

For the Yaw angle control, the conventional SMC exhibits the lowest errors across all metrics, with an IAE of 0.0025 and RMSE of 0.0002, indicating superior precision and effective orientation control. The neural network–based adaptive SMC controllers also perform satisfactorily, maintaining low error magnitudes with minor variations.

Among them, the ALMSMC exhibits slightly better accuracy than the ABPSMC and ARBFSMC, reflecting its strong learning efficiency for rotational dynamics. The marginal difference between the SMC and NN-based controllers suggests that the conventional SMC is already well-tuned for yaw regulation, leaving limited scope for further improvement through neural adaptation.

Summary

This chapter presents a comprehensive comparison between conventional SMC and NN-based controllers, including ABPSMC, ALMSMC, and ARBFSMC. The comparison is performed using quantitative performance indices for longitudinal, lateral, altitude, and yaw dynamics. Error-based metrics show that NN-based controllers consistently achieve improved tracking accuracy, with lower tracking errors and reduced control effort compared to classical SMC. The comparative results show that ARBFSMC yields the lowest error indices in most motion axes, reflecting its superior nonlinear gain adaptation capability in the presence of external disturbances. Overall, the results confirm the effectiveness of NN-based adaptive control schemes.

Chapter 7
Conclusion and Future Work

7.1 Conclusion

This research presents the design & development of three NN-enhanced ASMC schemes, namely ABPSMC, ALMSMC, and ARBFSMC. The three developed controllers are evaluated in a performance comparison with one another and with the conventional SMC. The comparative analysis demonstrated that integrating neural networks with the adaptive framework significantly enhances control accuracy and robustness across all control axes. The proposed adaptive architectures effectively minimized tracking errors, improved transient performance, and mitigated chattering while maintaining strong robustness against parameter uncertainties and external disturbances.

Among the evaluated models, the ARBFSMC demonstrated superior performance in most cases. This highlights the efficacy of data-driven learning in adaptively tuning the nonlinear controller gain, thereby enhancing the precision, adaptability, and stability of the nonlinear UAV control system. Importantly, integrating ASMC with NN-based intelligent mechanisms provides a robust approach to controlling complex dynamic process plants subject to nonlinearities and time-varying perturbations.

7.2 Future Work

Although the proposed framework demonstrated significant improvements, several promising research extensions can be envisioned. Future work may focus on integrating advanced deep learning architectures, such as long short-term memory (LSTM) networks and transformer-based models, to better capture temporal dependencies and enhance the adaptive estimation of controller gains under rapidly varying dynamics. In addition, the integration of reinforcement learning and hybrid data-driven strategies could facilitate autonomous tuning of controller parameters, enabling real-time self-optimization across diverse operational and environmental conditions.

Another promising avenue is to extend the proposed methodology to cooperative multi-agent UAV systems by utilizing data-driven ASMC for tasks such as formation control, coordinated navigation, and fault-tolerant operations. Additionally, future research may explore scaling the framework to larger and more complex aerial or robotic platforms to demonstrate its robustness, computational efficiency, and practical applicability in autonomous flight and intelligent control missions.

References

[1] S. Bouabdallah, "Design and control of quadrotors with application to autonomous flying," Ph.D. dissertation, Epfl, 2007.

[2] M. Zhang, W. Li, M. Wang, S. Li, and B. Li, "Helicopter–uavs search and rescue task allocation considering uavs operating environment and performance," *Computers & Industrial Engineering*, vol. 167, p. 107994, 2022.

[3] M. T. Nguyen, L. H. Truong, and T. T. Le, "Video surveillance processing algorithms utilizing artificial intelligent (ai) for unmanned autonomous vehicles (uavs)," *MethodsX*, vol. 8, p. 101472, 2021.

[4] A. D. Boursianis, M. S. Papadopoulou, P. Diamantoulakis, A. Liopa-Tsakalidi, P. Barouchas, G. Salahas, G. Karagiannidis, S. Wan, and S. K. Goudos, "Internet of things (iot) and agricultural unmanned aerial vehicles (uavs) in smart farming: A comprehensive review," *Internet of Things*, vol. 18, p. 100187, 2022.

[5] S. Gokool, M. Mahomed, R. Kunz, A. Clulow, M. Sibanda, V. Naiken, K. Chetty, and T. Mabhaudhi, "Crop monitoring in smallholder farms using unmanned aerial vehicles to facilitate precision agriculture practices: A scoping review and bibliometric analysis," *Sustainability*, vol. 15, no. 4, 2023.

[6] H.-S. Lee, B.-S. Shin, J. A. Thomasson, T. Wang, Z. Zhang, and X. Han, "Development of multiple uav collaborative driving systems for improving field phenotyping," *Sensors*, vol. 22, no. 4, 2022.

[7] U. F. Ukaegbu, L. K. Tartibu, M. O. Okwu, and I. O. Olayode, "Development of a light-weight unmanned aerial vehicle for precision agriculture," *Sensors*, vol. 21, no. 13, 2021.

[8] H. E. Mohamadi, N. Kara, and M. Lagha, "Heuristic-driven strategy for boosting aerial photography with multi-uav-aided internet-of-things platforms," *Engineering Applications of Artificial Intelligence*, vol. 112, p. 104854, 2022.

[9] T. F. Olivatto, F. F. Inguaggiato, and F. N. Stanganini, "Urban mapping and impacts assessment in a brazilian irregular settlement using uav-based imaging," *Remote Sensing Applications: Society and Environment*, vol. 29, p. 100911, 2023.

[10] A. Corbett and B. Smith, "A study of a miniature tdlas system onboard two unmanned aircraft to independently quantify methane emissions from oil and gas production assets and other industrial emitters," *Atmosphere*, vol. 13, no. 5, 2022.

[11] M. Sajid, H. Mittal, S. Pare, and M. Prasad, "Routing and scheduling optimization for uav assisted delivery system: A hybrid approach," *Applied Soft Computing*, vol. 126, p. 109225, 2022.

[12] H.-W. Lee, "Research on multi-functional logistics intelligent unmanned aerial vehicle," *Engineering Applications of Artificial Intelligence*, vol. 116, p. 105341, 2022.

[13] N. P. Nguyen, N. X. Mung, H. L. N. N. Thanh, T. T. Huynh, N. T. Lam, and S. K. Hong, "Adaptive sliding mode control for attitude and altitude system of a quadcopter uav via neural network," *IEEE Access*, vol. 9, pp. 40 076–40 085, 2021.

[14] T. Jiang, T. Song, and D. Lin, "Integral sliding mode based control for quadrotors with disturbances: Simulations and experiments," *International Journal of Control, Automation and Systems*, vol. 17, pp. 1987–1998, 2019.

[15] A. Eltayeb, M. F. Rahmat, M. A. M. Basri, M. M. Eltoum, and M. S. Mahmoud, "Integral adaptive sliding mode control for quadcopter uav under variable payload and disturbance," *IEEE access*, vol. 10, pp. 94 754–94 764, 2022.

[16] T. Huang, D. Huang, Z. Wang, X. Dai, and A. Shah, "Generic adaptive sliding mode control for a quadrotor uav system subject to severe parametric uncertainties and fully unknown external disturbance," *International Journal of Control, Automation and Systems*, vol. 19, no. 2, pp. 698–711, 2021.

[17] J.-J. Xiong and G.-B. Zhang, "Global fast dynamic terminal sliding mode control for a quadrotor uav," *ISA transactions*, vol. 66, pp. 233–240, 2017.

[18] D. Lee, H. Jin Kim, and S. Sastry, "Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter," *International Journal of control, Automation and systems*, vol. 7, pp. 419–428, 2009.

[19] S. B. F. Asl and S. S. Moosapour, "Adaptive backstepping fast terminal sliding mode controller design for ducted fan engine of thrust-vectored aircraft," *Aerospace Science and Technology*, vol. 71, pp. 521–529, 2017.

[20] K. Mei and S. Ding, "Second-order sliding mode controller design subject to an upper-triangular structure," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 497–507, 2018.

[21] O. Mofid and S. Mobayen, "Adaptive sliding mode control for finite-time stability of quad-rotor uavs with parametric uncertainties," *ISA transactions*, vol. 72, pp. 1–14, 2018.

[22] H. Razmi and S. Afshinfar, "Neural network-based adaptive sliding mode control design for position and attitude control of a quadrotor uav," *Aerospace Science and technology*, vol. 91, pp. 12–27, 2019.

[23] Y. Yang and Y. Yan, "Attitude regulation for unmanned quadrotors using adaptive fuzzy gain-scheduling sliding mode control," *Aerospace Science and Technology*, vol. 54, pp. 208–217, 2016.

[24] E.-H. Zheng, J.-J. Xiong, and J.-L. Luo, "Second order sliding mode control for a quadrotor uav," *ISA transactions*, vol. 53, no. 4, pp. 1350–1356, 2014.

[25] J.-J. Xiong and E.-H. Zheng, "Position and attitude tracking control for a quadrotor uav," *ISA transactions*, vol. 53, no. 3, pp. 725–731, 2014.

[26] S. Mondal and C. Mahanta, "Chattering free adaptive multivariable sliding mode controller for systems with matched and mismatched uncertainty," *ISA transactions*, vol. 52, no. 3, pp. 335–341, 2013.

[27] F. Muñoz, I. González-Hernández, S. Salazar, E. S. Espinoza, and R. Lozano, "Second order sliding mode controllers for altitude control of a quadrotor uas: Real-time implementation in outdoor environments," *Neurocomputing*, vol. 233, pp. 61–71, 2017.

[28] J. Zhang, Z. Ren, C. Deng, and B. Wen, "Adaptive fuzzy global sliding mode control for trajectory tracking of quadrotor uavs," *Nonlinear Dynamics*, vol. 97, pp. 609–627, 2019.

[29] S. Zeghlache, H. Mekki, A. Bouguerra, and A. Djerioui, "Actuator fault tolerant control using adaptive rbfnn fuzzy sliding mode controller for coaxial octorotor uav," *ISA transactions*, vol. 80, pp. 267–278, 2018.

[30] Q. Xu, Z. Wang, and Z. Zhen, "Adaptive neural network finite time control for quadrotor uav with unknown input saturation," *Nonlinear dynamics*, vol. 98, pp. 1973–1998, 2019.

[31] Y. Yin, H. Niu, and X. Liu, "Adaptive neural network sliding mode control for quad tilt rotor aircraft," *Complexity*, vol. 2017, no. 1, p. 7104708, 2017.

[32] G. X. Wu, Y. Ding, T. Tahsin, and I. Atilla, "Adaptive neural network and extended state observer-based non-singular terminal sliding modetracking control for an underactuated usv with unknown uncertainties," *Applied Ocean Research*, vol. 135, p. 103560, 2023.

[33] Y. Zhang, Q. Yao, B. Luo, and N. Chen, "Robust control of uncertain asymmetric hysteretic nonlinear systems with adaptive neural network disturbance observer," *Applied Soft Computing*, vol. 167, p. 112387, 2024.

[34] H. Zhou, W. Chen, J. Liu, L. Cheng, and M. Xia, "Trustworthy and intelligent fault diagnosis with effective denoising and evidential stacked gru neural network," *Journal of Intelligent Manufacturing*, vol. 35, no. 7, pp. 3523–3542, 2024.

[35] H. Razmi and A. M. Kashtiban, "Nonlinear pid-based analog neural network control for a two link rigid robot manipulator and determining the maximum load

carrying capacity," *International Journal of Soft Computing and Engineering*, vol. 2, no. 1, pp. 228–234, 2012.

[36] S. Raiesdana, "Control of quadrotor trajectory tracking with sliding mode control optimized by neural networks," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 234, no. 10, pp. 1101–1119, 2020.

[37] S. C. Yogi, V. K. Tripathi, and L. Behera, "Adaptive integral sliding mode control using fully connected recurrent neural network for position and attitude control of quadrotor," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 12, pp. 5595–5609, 2021.

[38] X. Zhang, Y. Wang, G. Zhu, X. Chen, and C.-Y. Su, "Discrete-time adaptive neural tracking control and its experiments for quadrotor unmanned aerial vehicle systems," *IEEE/ASME Transactions on Mechatronics*, vol. 28, no. 3, pp. 1201–1212, 2021.

[39] Z. Ma, J. Ding, Y. Zhou, and T. Yuan, "Neural network based finite-time super twisting sliding mode control for quad-rotor uav with parameter uncertainty and mismatched disturbance," in *2022 41st Chinese Control Conference (CCC)*. IEEE, 2022, pp. 516–521.

[40] S. C. Yogi, L. Behera, and S. Nahavandi, "Adaptive intelligent minimum parameter singularity free sliding mode controller design for quadrotor," *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 2, pp. 1805–1823, 2023.

[41] W. Le, P. Xie, and J. Chen, "Disturbance rejection control of the agricultural quadrotor based on adaptive neural network," *Information Processing in Agriculture*, 2024.

[42] T. Omkar, A. Nitinkumar, P. Supriya, and S. Suraj, "Quadcopter: Design, construction and testing," *International Journal of Research in Engineering Applications and Management (IJREAM)*, vol. 4, pp. 1–7, 2018.

[43] E. Getiye, "Model predictive control of unmanned aerial vehicle for locust detection and bio-pesticide spraying," MSc thesis, Addis Ababa University, School of Electrical and Computer Engineering, Industrial Control Engineering, Addis Ababa, Ethiopia, 2021, [Online]. [Online]. Available: <InserttheactualURLhere>

[44] S. Bouabdallah, P. Murrieri, and R. Siegwart, "Design and control of an indoor micro quadrotor," *IEEE International Conference on Robotics and Automation*, pp. 4393–4398, 2004.

[45] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 20–32, 2012.

[46] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*. Prentice Hall, 1991.

[47] V. I. Utkin, *Sliding Modes in Control and Optimization*. Springer, 1992.

[48] C. Edwards and S. Spurgeon, *Sliding Mode Control: Theory and Applications*. CRC Press, 1998.

[49] R. Patnaik and P. Dash, "Fast adaptive finite-time terminal sliding mode power control for the rotor side converter of the dfig based wind energy conversion system," *Sustainable Energy, Grids and Networks*, vol. 1, pp. 63–84, 2015.

[50] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*. PMLR, 2016, pp. 1050–1059.

[51] H. Yu and B. M. Wilamowski, "Levenberg–marquardt training," in *Intelligent Systems*, B. M. Wilamowski and J. D. Irwin, Eds. CRC Press, 2018, pp. 12–1–12–16.