Addis Ababa University

Technology and Built Environment

School of Electrical and Computer Engineering

Adaptive Neuro-Fuzzy Inference System based Sliding Mode Control in the Presence of External Disturbances and Parameter Variation for Quadcopter UAV.

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

By

Daniel Fikadu

Advisor

Lebsework Negash (PhD)

April 30, 2025

Addis Ababa, Ethiopia

Addis Ababa University

Technology and Built Environment

School of Electrical and Computer Engineering

Adaptive Neuro-Fuzzy Inference System based Sliding Mode Control in the Presence of External Disturbances and Parameter Variation for Quadcopter UAV.

By

Daniel Fikadu

APPROVED BY BOARD OF EXAMINERS

Dean, School of Graduate Committee	Signature	Date
Dr. Bisrat Derebssa	_____	_____
Advisor	Signature	Date
Dr. Lebsework Negash	_____	_____
External Examiner	Signature	Date
Dr. Chala Merga	_____	27.June.2027
Internal Examiner	Signature	Date
Mr. Teshome Hambissa	_____	_____

Declaration

I hereby declare that except where the acknowledgment is made, the MSc research titled *" Adaptive Neuro-Fuzzy Inference System based Sliding Mode Control in the Presence of External Disturbances and Parameter Variation for Quadcopter UAV."* presented here is original and has never been submitted to any other institutions.

Student Name — Signature

Daniel Fikadu _____

Place:

Technology and Built Environment
Addis Ababa University, AAU
Addis Ababa, Ethiopia.

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

Advisor	Signature	Date
Dr. Lebsework Negash	_____	_____

Acknowledgment

I would like to express my heartfelt gratitude to all those who have supported and contributed to the completion of this thesis. I would like to offer an appreciation to my thesis advisor, **D**r. **L**ebsework **N**egash for your guide, and unwavering support have been invaluable throughout this journey. Your wisdom and dedication to the pursuit of knowledge have inspired me greatly. I wish to acknowledge the Technology and Built Environment for providing the necessary resources, including access to libraries, laboratories, and research materials, which have been instrumental in the successful completion of this research. My sincere thanks go to my wife and my family for their unwavering encouragement and understanding during the ups and downs of this academic endeavor. Your belief in my abilities has been a constant source of motivation. Finally, I extend my appreciation to all the individuals and participants who contributed their time and knowledge for enabling this research to come to fruition. This work would not have been possible without the collective support and collaboration of all those mentioned above. Thank you for being an integral part of this academic endeavor.

Daniel Fikadu

Abstract

Quadrotor UAVs become more and more essential for surveillance, military and defense, crop spraying, rescue missions, and to assess areas of destruction that are too dangerous for humans to be engaged. But quadrotor is challenging in its control mechanism because of its underactuated and coupled system due to these reasons different control methods have been a focus of many researches.

It is shown through review of the literature that When it comes to handling nonlinearity, underactuation, and coupled systems, sliding mode control (SMC) offers numerous benefits over alternative control techniques. The Adaptive Neuro Fuzzy Inference System (ANFIS), an intelligent controller, can minimize the chattering phenomena that may affect system performance. In this work, Adaptive Neuro Fuzzy Inference System based sliding mode control (ANFIS-SMC) is proposed to tackle the issue and to improve the trajectory tracking performance. Combining the ANFIS with the SMC technology proposes the intelligent robust controller scheme that is composed of controllers, which includes position, altitude and attitude which enables tracking control of quadcopters, which depends on the quality of the training data i.e the error input, and the output of the SMC control signals, then the ANFIS based SMC controller to be able to adapt changes by minimizing the chattering effect.

To validate the performance of the proposed ANFIS tuning SMC Controller, a comparison is illustrated between the conventional Sliding mode controller with proposed SMC based ANFIS controller using same system. From the result, ANFIS-SMC controller remove the chattering and gives better performance compared with conventional SMC methods developed in the study. The proposed control scheme is verified by developing simulation results for the quadcopter using MATLAB/SIMULINK software.

Keywords : Quad copter; Adaptive Neuro Fuzzy Inference System (ANFIS); Sliding Mode Control (SMC); Unmanned Aerial Vehicle (UAV); Trajectory tracking

Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligent
AFSMC	Adaptive Fuzzy Sliding Mode Controller
ANFIS	Adaptive Neuro-Fuzzy Inference System
ANFIS-SMC	Adaptive Neuro-Fuzzy Inference System based Sliding Mode Controller
AR	Augmented Reality
DOF	Degrees of Freedom
FLS	Fuzzy Logic Systems
IACO	Improved Ant Colony Optimization
LQR	Linear Quadratic Regulator
NFSMC	Neuro Fuzzy Sliding Mode Controller
NFS	Neuro-Fuzzy Systems
NN	Neural Networks
PID	Proportional Integral Derivative
SMC	Sliding Mode Control
UAV	Unmanned Aerial Vehicles

Chapter 1

Introduction

1.1 Overview

Unmanned Aerial Vehicles (UAV) have extensive attention due to their vast potential in both military and civilian applications [1]. They play a crucial role in a wide range of domains, including disaster rescue, intelligent surveillance, smart agriculture, aerial photography and mapping, industrial inspection, logistics, wild fire protection, and crop monitoring. UAV's are becoming a crucial component in technological solutions due to their adaptability and improvements in their capabilities. Because of their distinct benefits, quadcopters have become the most popular type of UAV among the others. Quadcopters mostly applied in research areas because of their autonomous operation, low maintenance costs, ease of construction, onboard vision capabilities, vertical takeoff and landing, hovering ability, and maneuverability [2].

Quadcopters are able to carry out difficult tasks including autonomous decision-making, real-time navigation, and object detection due to these cutting-edge technologies. UAVs are becoming more dependable for mission-critical applications because to AI-driven control techniques that enable them to adjust to changing conditions. AI-powered UAVs, for example, can evaluate topography, find survivors, and optimize flight paths in real-time during search and rescue missions. Similar to this, quadcopters with AI-based sensors in smart agriculture may optimize pesticide use, monitor crop health, and identify diseases, resulting in more environmentally friendly agricultural methods. More complex control methods are required to further improve UAV performance as a result of these developments, which have created new research and development opportunities.

With four rotors configured in a combination of clockwise and counterclockwise rotations, quadcopters can spin in six different directions and accomplish translational motion in Cartesian space. Additionally, they perform rotational movements through yaw, pitch, and roll angles around the axes of an inertial reference frame, which are derived from the Newton-Euler equations [3]. The highly nonlinear dynamics of quadcopters present significant challenges in control system design. Researchers have explored various control techniques, including Sliding Mode Control (SMC), Proportional Integral Derivative (PID), backstepping control, nonlinear H-infinity control, and Linear Quadratic Regulator (LQR), to stabilize and regulate quadcopter flight. However, traditional control methods often struggle to handle uncertainties, external disturbances, and sudden changes in environmental conditions. As a result, model-free intelligent control techniques have gained attention, offering improved adaptability and robustness in uncertain environments.

To address these challenges, researchers have developed intelligent control techniques such as Neural Networks (NN), Fuzzy Logic Systems (FLS), and Neuro-Fuzzy Systems (NFS) to improve UAV stability and tracking performance [4]. These techniques leverage machine learning and adaptive algorithms to adjust control parameters dynamically, ensuring optimal performance even in the presence of uncertainties. The integration of AI-driven controllers allows quadcopters to make real-time adjustments based on sensor data, enhancing their ability to navigate complex terrains, avoid obstacles, and perform precision maneuvers. Despite these advancements, designing an advanced control system for quadcopters remains a challenging task due to their underactuated and strongly coupled dynamics. Environmental disturbances, unpredictable payload variations, and aerodynamic effects further complicate the control problem. Therefore, continuous research efforts are required to develop more robust, adaptive, and efficient control strategies that can ensure reliable UAV operation across diverse applications.

Quadrotor control has made considerable use of sliding mode control (SMC). This method has the advantage of being robust to disturbances able to stabilize the system, and insensitive to model mistakes and parametric uncertainties [5]. One of the drawbacks for Sliding Mode Control (SMC) is the chattering effect, which is the high frequency switching of the controller output. Neural Network (NN) is type of Artificial Intelligent (AI) technique that is used to approximate the function represented by a given data. It is considered as a complex adaptive system which can change its internal struc-

ture based on the information passing through it. Fuzzy logic systems are very efficient for high-level reasoning and decision and often used in complex systems to overcome the limitations of conventional mathematical tools. They have been used in sliding mode control to replace the discontinuous switching function [6].

The primary tool in this work is a neuro-fuzzy method known as the Adaptive Neuro Based Fuzzy Inference System (ANFIS). Neural networks and fuzzy inference systems are fused in the neurofuzzy technique Adaptive Neuro-Fuzzy Inference System (ANFIS) [7][8][9]. While the neural network offers the system a sense of adaptability, fuzzy logic accounts for the system's imprecision and uncertainty. With the aid of the rules taken from the input-output data of the system being modeled, a preliminary fuzzy model and its input variables are first created using this hybrid approach. The final ANFIS model of the system is then created by fine-tuning the rules of the original fuzzy model using a neural network. ANFIS is used as powerful tool for modeling and prediction of non-linear systems. So, in this study the ANFIS based SMC is implemented to attain desired system dynamics as well as to minimize the chattering effect .

Figure 1.1: Quadcopter Model

1.2 Problem Statement

Quadcopter inherent instability, non-linearity, and underactuation, six degrees of freedom must be designed by combining three translational and three rotational motions to produce the desired trajectory. A quadcopter's under-actuated characteristics make automatic control challenging; it is challenging to regulate all six states with just four control inputs. With the increasing complexity in quadcopter dynamics, big challenges arise in the precise control of the quadcopters, especially in an unpredictable environment. Therefore, there is an overwhelming necessity to have a design for advanced control strategies that can be applied in adaptive management of quadcopter behavior in real-time.

To mitigate these difficulties, the combination of Adaptive Neuro-Fuzzy Inference System (ANFIS) and Sliding Mode Control (SMC) used in modeling the quadcopter non-linear dynamics, hence provide adaptability in learning and decision-making. On the other hand, SMC gives robust against system uncertainties and external disturbances, guaranteeing the stability and performance of a quadcopter. However, SMC exhibit chattering, which is rapid switching of the control signal and demand a high control effort.

In summary, this study aims to development of an ANFIS tuning SMC control system for quadcopters. This research, hence, tries to push the boundaries of those limitations of conventional control techniques and to open ways for more adaptive control in aerial vehicles, which are able to carry out different tasks.

1.3 Objective of the Study

1.3.1 General Objective

The main goal of this study is to design and simulate Adaptive Neuro-Fuzzy Inference System based Sliding Mode Controller (ANFIS-SMC) to improve trajectory tracking performance of quadcopter UAV.

1.3.2 Specific Objectives

- Develop a non-linear dynamics model of the quadcopter UAV using Newton-Euler method.
- Design sliding mode control to provide robust performance against disturbances,

ensuring the quadcopter adheres to its intended trajectory.

- Implement ANFIS to tune SMC's control law and minimize chattering effects for quadcopters.
- Validate the controller's performance via MATLAB-Simulink simulations.

1.4 Significance of the Study

The primary goal of this paper is to develop an adaptive intelligent control system for stabilizing a quadrotor in the presence of system uncertainties and disturbances. To achieve this, we replace conventional nonlinear controllers with intelligent control due to their adaptive nature, lower computational cost, and ability to approximate complex nonlinear systems.

We propose a neural network-based fuzzy logic system trained using sliding mode control (SMC) data. By integrating an both controllers in parallel with the system, the controller effectively handles uncertainties and disturbances while ensuring stability during the SMC training phase. This strategy keeps system states within a safe region, allowing accurate training of the ANN and fuzzy inference system. Overall, the proposed control method reduces chattering, enhances robustness, and minimizes control effort, leading to improved quadrotor performance.

1.5 Methodology

The study will be achieved through the following methodology:

- To begin, study the literature, which involves reading books, journal articles, and theses, in order to gather the necessary data, concepts, and ideas that will help you concentrate on your thesis.
- developing the quadcopter's mathematical model.
- The design of SMC is resilient and impervious to unmodeled dynamics, parameter uncertainty, and external disturbances.
- In order to change the premise and ensuing parameters, the ANFIS system scheme was trained using SMC input/output data and a hybrid learning algorithm (which combines back propagation and least square estimate).

- In order to guarantee robustness, parameter uncertainty, and insensitivity to external disturbances while tracking the intended reference trajectory of the quadcopter, ANFIS-based SMC has been evaluated under various conditions.
- Trajectory tracking performance for convectional sliding mode control has been compared using an adaptive neurofuzzy inference system using MATLAB/Simulink simulation.

1.6 Scope of the thesis

This thesis aims to establish a mathematical model, design, and simulation of ANFIS tuning SMC for effective trajectory tracking and stabilization for the dynamic quadcopter model. The MATLAB/Simulink environment will be used to implement the controller and system design.

1.7 Outlines of the thesis

The thesis is divided into six chapters, with chapter 1 serving as the introduction.

1 Chapter 2: explains the quadcopter's fundamental alignment, structure, and flying principles.

2 Chapter 3: Deals with mathematical modelling of the quadcopter design.

3 Chapter 4: Design conventional sliding mode controller and train the data for neuro fuzzy system.

4 Chapter 5:summarizes the findings and provides a brief discussion of them.

5 Chapter 6: recommendations for future work to be done after drawing conclusions from the study done in this thesis.

Chapter 2

Literature Review

2.1 Overview of quadcopter

Model-based controllers such as SMC are inherently nonlinear and are designed to handle variations in system dynamics. However, their robustness is constrained by the accuracy of the system model. In contrast, intelligent controllers like Neural Networks (NN) and Fuzzy Logic (FL) do not rely on precise mathematical models. As a result, they are often more effective in adapting to unexpected changes and uncertainties, offering greater flexibility than conventional control methods.

Few studies in the literature have been used to control the quadcopter's movement using an ANFIS controller. The authors in reference [4] studies the non-linear modeling of the lateral, longitudinal, and vertical dynamics of the Augmented Reality (AR) Drone. They examined how the findings enhance overall performance by contrasting the attitude results of the controller with the performance of the linear system identification techniques in the absence of position control validation.

The neuro-fuzzy controller using the Raspberry Pi 3 platform for quadcopter in geosensor networks are provided by the authors in reference [9]. In order to stabilize a quadcopter's attitude and/or altitude during flight, the suggested neuro-fuzzy controller was created for the Raspberry Pi-3 platform and tested on a real quadrotor UAV. The results showed that it performed better than the PID controller. However, as disadvantage of ANFIS on a Raspberry Pi platform is that it might be challenging to have real-time performance due to high processing delays that could interfere with the control system.

According to reference [10], by combining the FLC with conventional SMC, the hybrid position control of robotic manipulators operating in uncertain environments. The

well-established Adaptive Fuzzy Sliding Mode Controller (AFSMC) requires the fewest data on the structure and properties of the manipulator when compared to other hybrid control approaches that are currently available. At last, simulation results showed how well the established controller performed when dealing with improbabilities. The control strategy requires significant processing witch need computational power for optimal performance.

Research presented in [11], [12] ,used SMC to acquire the problem of attitude control with external disturbances using the non-linear sliding mode control. This approach has the capacity to stabilize the quad-rotor in any desired position with an appropriate switching law even in a given noise, but there is a serious of chattering problem and the gains of the controller are adjusted manually.The results of the simulation demonstrate how effectively the control strategy preserves tracking performance and stability, significantly advancing the field of aerial robots and opening the door for future studies that concentrate on experimental validations and additional improvements. The tilting-rotor UAV is more complex mathematical model with coupled nonlinear equations, making the design and implementation of the SMC significantly more challenging.

Muhammet[13] presents a detailed exploration of adaptive neuro-fuzzy inference systems (ANFIS) applied to the control of aerial vehicles, specifically focusing on quadrotors. The thesis investigates both Type-1 and Type-2 fuzzy logic systems, developing new training models and testing them with specific equations to improve control applications. The research begins with an examination of various fuzzy inference systems (FIS), such as Sugeno and Mamdani systems, which are popular in control systems.The thesis highlights the advantages of the Sugeno model due to its reduced computational load compared to the Mamdani model, which requires a more complex defuzzification process. One significance of the thesis is the gathering of new training algorithms for these fuzzy systems. These algorithms were tested on a quadrotor model, demonstrating improved performance in position and orientation control. The modified ANFIS controller showed excellent results in single-layer structures, achieving precise control without overshoot and with minimal settling time. Furthermore, the thesis addresses the challenge of real-time parameter adaptation in neuro-fuzzy controllers. This involves continuously updating the control parameters based on system feedback, which is crucial for maintaining optimal performance in dynamic environments. The proposed methods were compared with existing approaches, showing notable improvements in terms of accuracy and efficiency and as a drawback used NFCs with three inputs (error, output derivative, sum of

error) which need more computational time.

Selma [14] uses the Improved Ant Colony Optimization (IACO) method in this paper to optimize the ANFIS controller's settings. By increasing the convergence rate and guaranteeing greater trajectory tracking accuracy, this improvement seeks to lower learning errors and improve controller performance. Because of its capacity to identify the best routes and adjust to shifting conditions, ant colony optimization—which draws inspiration from ant's which is ideally suited for challenging optimization tasks. The study compares the IACO tuned ANFIS controller's performance to that of conventional PID and standard ANFIS controllers. The findings show that in trajectory tracking tasks, the hybrid ANFIS-IACO controller performs noticeably better than both the traditional ANFIS and PID controllers. Decreased error rates and more accuracy in intended flight path are clear indicators of this improvement. The efficacy of the approach is largely dependent on the ACO parameters, which can be difficult to properly tune.

According to Darwito [15], the suggested controller combines SMC and ANFIS. ANFIS facilitates dynamic control parameter adjustment, enabling the system to adjust to system performance. ANFIS is used in the suggested system to modify the control gains with a predetermined rule sets. The study's conclusions offer important new information for creating UAV control systems that are more efficient. It is concluded that the ANFIS-based SMC approach greatly enhances the trajectory tracking and overall performance of quadcopters in the presence of disturbances. This combination of ANFIS and SMC can be especially helpful in dynamic environments, helping to develop robust and responsive UAV control strategies. This approach presents a viable way to address the control issues that UAVs encounter in a range of applications. As a disadvantage Vectorial distance may not appropriately represent how velocities, accelerations, or outside disturbances change in dynamic systems like UAVs when evaluating real-time trajectory tracking performance.

In this paper develops an Adaptive Neuro-Fuzzy Inference System (ANFIS) to control a quadrotor UAV [16] the study employs an ANFIS, which combines the learning capabilities of neural networks with the human-like reasoning of fuzzy logic.The ANFIS controller parameters are optimized using advanced algorithms to ensure precise control over the quadrotor's attitude. The paper contrasts the performance of the ANFIS controller with conventional PID controllers, highlighting the improvements achieved.The ANFIS controller demonstrates superior performance in stabilizing the quadrotor's attitude compared to conventional PID controllers and the simulation result show that the

ANFIS controller provides better handling of input disturbances and parameter variations, ensuring more stable and reliable flight.The paper by Rezazadeh and colleagues provides a detailed exploration of using ANFIS for optimal attitude control of quadrotor UAVs. By leveraging the adaptive and learning capabilities of ANFIS, the proposed controller outperforms traditional methods, ensuring enhanced stability and reliability in UAV operations. This research is instrumental for engineers and researchers aiming to develop advanced control systems for UAVs. The paper only work on the attitude control it does not consider the position control of the system.

As [17] the development of a Neuro Fuzzy Sliding Mode Controller (NFSMC) for controlling the speed of switched reluctance motors (SRM) in electric vehicles marks a notable improvement in control systems. It merges the strength of SMC with the adaptive features of neuro-fuzzy systems. This method effectively tackles the nonlinear dynamics and uncertainties that are typical in SRMs, leading to better speed regulation and enhanced overall system performance. By integrating fuzzy logic, the controller can utilize expert knowledge, while the neural network aspect offers adaptive learning, allowing it to respond to changing operating conditions and load disturbances. By boosting efficiency and responsiveness, the NF-SMC not only improves acceleration and deceleration profiles but also optimizes energy use, thus extending the vehicle's operational range. In summary, this cutting-edge control strategy stands out as a promising option for contemporary electric vehicle applications, fostering both performance and sustainability. As a drawback does not fully address the challenges related to the real-time performance such as the dynamics of both the motor and the electric vehicle may vary with changes in load, temperature, and voltage.

This work proposes the implementation of an ANFIS based SMC to supervise quadcopter trajectory tracking. The effectiveness and efficacy of this strategy are demonstrated by simulated studies carried out in Matlab/Simulink in the study.

Chapter 3

Modeling and Model Verification

3.1 Modeling Approach

This chapter will create a quadcopter kinematics and dynamics models using a Newton-Euler formalism under the following presumptions:

- The quadcopter center of mass and the body frame origin are the same.
- Thrust and Torque is proportional to the square of rotor speed.
- The propellers are rigid.
- The quad-copter is a rigid structure and has a symmetric.

The rotor dynamics of the actuators and the aerodynamic effects operating on the quadrotor body will be explored after the kinematics and dynamics models of the quadrotor have been derived. The chapter will conclude with the development of a state space model for the quadrotor system, which will be used in the following modeling chapter.

3.2 The basic structure of the Quadcopter

A quadcopter is a type of unmanned aerial vehicle that relies on four rotors each located at the end of the arms extending from the frame. The motors are responsible for spinning the propellers and generating lift typically controlled electronically, allowing for precise adjustments in speed. Its unique design allows for stable flight, maneuverability, and versatility in various applications.

There are two types of design configurations for a quadcopter with four rotors: plus "+" and cross "x" configuration. While the roll and pitch dynamics are different for both, the thrust and yaw dynamics are similar. Put another way, in "+" configurations, the roll and pitch moments are produced by two rotors alone, but in "x" configurations, these moments are produced by four motors. For "x" and "+" configurations, 3.1 take note of the axes' directions in relation to each setup.

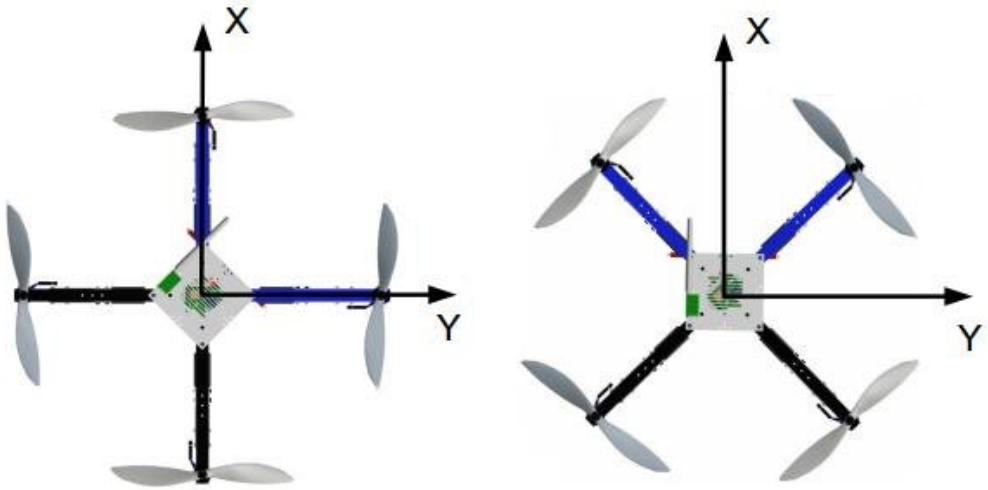


Figure 3.1: Quadcopter "+" Configuration and "X" Configuration.

In both arrangements, increasing rotor speed produces vertical upward motion, while decreasing rotor speed produces vertical downward motion. Roll and Pitch rotation in the "+" and "x"configuration is achieved by adjusting the speeds of the front and rear rotors, while roll rotation is accomplished by adjusting the speeds of the left and right rotors. This thesis adopts the "x" configuration under study so, four rotors' speeds are adjusted for "x" configurations based on the desired rotation. For the following analysis, the cross configuration will be used because of the following two main advantages:-

- The cross configuration gives provides better balance and simplifies the control mechanism, i.e. if one of the rotors fails happens, the quad-copter moves stably due to the second rotor in case of cross configuration.
- For a camera mounted on a quadcopter, the cross arrangement ensures that there is no blockage to the camera.

3.2.1 Working Principle of Quadcopter

Quadrotor related to the four basic movements which allow the quadcopter to reach a certain altitude and attitude. The description of these basic movements [18]:

1. Thrust:- From the figure 3.2 Increasing or Decrease all four rotors simultaneously with The quadcopter is raised or lowered by a vertical force of the same amount. The inertial frame vertical direction coincides with the quadcopter's horizontal position. If not, the given push causes the inertial frame to accelerate both vertically and horizontally. The rotor generates the same lift force Since the torque generated by each arm is identical in magnitude but directed in the opposite direction, the net torque simply cancels out before stabilizing vertically. The quadcopter will lift off vertically upward when the total force of its four rotors exceeds the body's gravity.

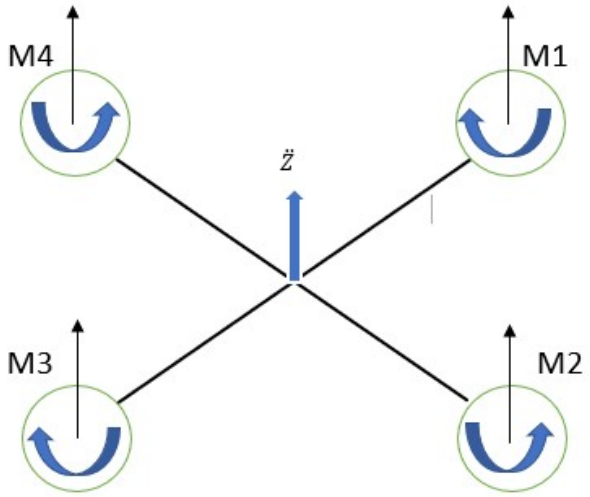


Figure 3.2: Thrust Movement

2. Roll:-Either minimizing the speed of the two left-side motors (M3, M4) while maximizing the speed of the remaining right-side propellers (M1, M2) or increasing the speed of the left-side propellers (M2, M3) while decreasing the speed of the remaining right-side propellers (M1, M4) will create Roll movement which leads x_B axis torque. Maintaining the total effective thrust during a roll command is crucial to ensuring that only roll acceleration takes place, allowing the vehicle to move along the body reference frame, as depicted in 3.3.

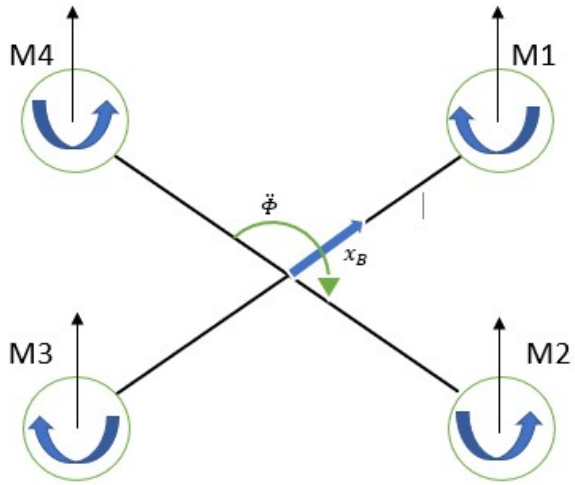


Figure 3.3: Roll Movement

3. Pitch:- The rear propeller speed can be increased or decreased, and the front propeller speed can be decreased or increased. The quadrotor turns as a result of the torque it produces with regard to the y_B axis. This command merely accelerates the pitch angle because the total vertical thrust is the same as when hovering. (in first approximation) i.e resulting in four-wing aircraft front and back ends of unbalanced lift, so that the quadcopter body flip to the front and back as shown in 3.4.

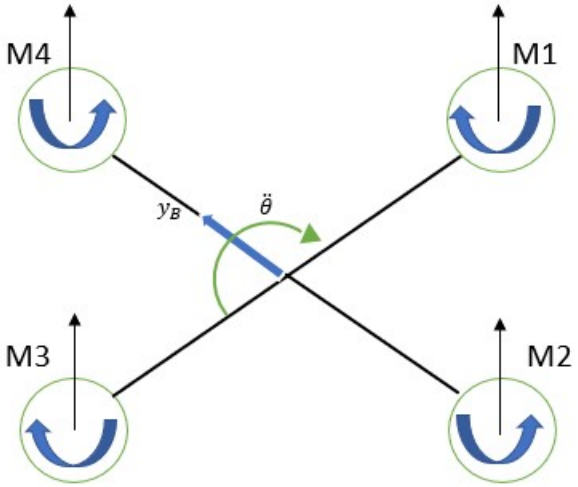


Figure 3.4: Pitch Movement

3. Yaw:-The speed of the front-rear propellers increases or decreases in response to 3.5, and the left-right motor speed decreases or increases as well. The quadrotor

turns as a result of the torque it produces with regard to the z_B axis. Because the front-rear propellers revolve counterclockwise while the left-right propellers rotate clockwise, the yaw movement is produced. This instruction merely accelerates the yaw angle because the total vertical thrust is the same as hovering.

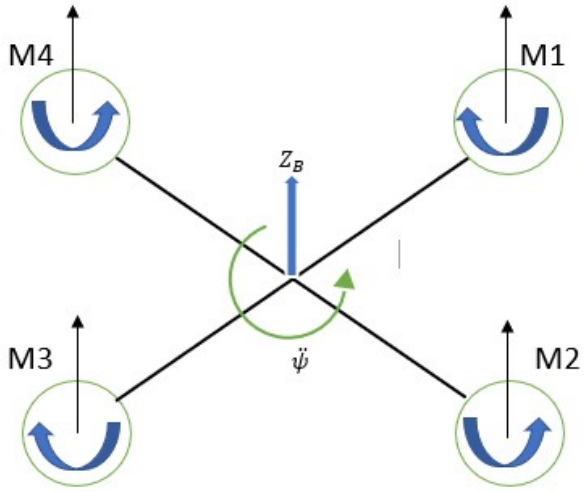


Figure 3.5: Yaw Movement

3.2.2 Kinematic model

The inerial frame and the quadcopter frame are the two coordinate frames that will be employed in the modeling. The quadcopter frame is positioned in the middle of the quadcopter body, whereas the inertial frame is an ground level are shown in 3.6. this paper utilize the ENU (East, North, Up) reference frame as the global coordinate system when building controllers for quadrotors. Roll, pitch, and yaw angles are represented as Euler angles (Roll (ψ), Pitch (θ), and Yaw (ϕ)). Figure 3.6 displays the quadrotor's coordinate systems.

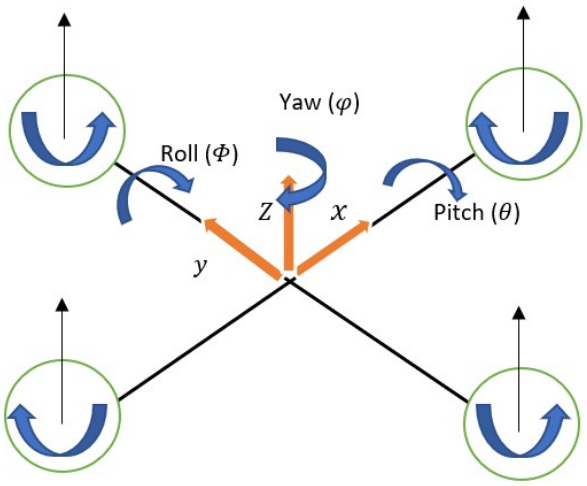


Figure 3.6: Coordinate frames of the quadrotor

The body frame is $B-(Oxyz)$, and the earth frame is $E-(Oxyz), wherem, g$, and l descibe the rotor mass, gravity acceleration, and the distance of each rotor to centre of gravity of rotor respectively. The orientation of the rotor from body frame to earth frame is represented by rotation matrices of $R : e \longrightarrow B$, where these matrices depend on the Euler angle (ϕ, θ, ψ).

Rotation matrix

The orientation of a quadrotor in three dimensions is described by its rotational matrix. It is often represented as a 3x3 rotation matrix derived from the Euler angles or quaternion representation and without singularities, quaternions offer a representation of attitude. Quaternionic attitude representation uses a single axis and a rotation angle as opposed to Euler angles, which involve three angles. For the purposes of this thesis, Euler angle representation is adopted because the application of this thesis doesn't require acrobatic movements.

The rotation of a rigid body in space can be characterized using the Euler angle, which is a mathematical description of three consecutive rotations about a possible axes. Implementing a right-hand oriented coordinate system yaw (ψ), pitch (θ), and roll (ϕ) as the Euler angles, the overall rotational matrix can be expressed as 3.1:

$$R = R_x(\psi)R_y(\theta)R_z(\phi) \tag{3.1}$$

- $R(x, \phi)$, rotation around x -axis.
- $R(y, \theta)$, rotation around y -axis.

- $R(z,\psi)$, rotation around z -axis.

They are represented by:

$$R(x,\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \tag{3.2}$$

$$R(y,\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\theta \end{bmatrix} \tag{3.3}$$

$$R(z,\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

$$R = \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi - \sin\phi\cos\psi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \tag{3.5}$$

3.3 Dynamics model

In most of literature's quadrotors, are modeled as a rigid-body vehicles. Quadrotors can be expressed with six Degrees of Freedom (DOF) and it will be represented by six differential equations. Finding these differential equations in between the two reference frames is done using the Newton-Euler's Equations as stated on the previous section. Newton–Euler is the processes of grouping Euler's two laws of motion for a quadrotor into a single equation with 6 component matrices. These laws relate the center of gravity motion of a quadrotor with the sum of forces and torques acting on the quadrotor. The quadcopter consists of two subsystems, The first one is the angular subsystem which consists of the three angles $\phi, \theta and \psi$, while the second subsystem consists of the altitude in x, y and z positions. The equations with respect to a body frame are:

$$\begin{bmatrix} mJ_3 & 0 \\ 0 & J \end{bmatrix} \begin{bmatrix} \dot{V} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} \omega \times mV \\ \omega \times J\omega \end{bmatrix} = \begin{bmatrix} F \\ \tau \end{bmatrix} \tag{3.6}$$

Where:

$F =$ linear forces acting on the center of mass of a quadrotor in (x, y, z)

$\tau =$ torque acting about the center of mass

$m =$ mass of the quadrotor

$J_3 = 3 \times 3$ inertia matrix

$\dot{V} =$ linear accelerations

$V =$ the body linear speed vector

$J_c =$ moment of inertia

$\omega =$ angular velocity of the quadrotor

$\dot{\omega} =$ angular acceleration of the quadrotor

The attitude and position of the quadrotor defined in inertial reference frame written as a vector of P_A :

$$P_A = \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^T \tag{3.7}$$

For complete kinematic equations the transitional matrix for rotational positions and velocities from body frame to inertial frame is used and represented as a matrix $T_{(\phi,\theta)}$;

$$T_{(\phi,\theta)} = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta) \end{bmatrix} \tag{3.8}$$

where the transformation matrix's inverse will be:

$$T^{-1}_{(\phi,\theta)} = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \cos(\theta)\sin(\phi) \\ 1 & -\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix} \tag{3.9}$$

Since the body frame represents the quadrotor's center of mass, let's use V_B and ω_B to denote the linear and angular velocities, respectively.

$$\begin{aligned} V_B &= \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^T \\ \omega_B &= \begin{bmatrix} \delta & \beta & \gamma \end{bmatrix}^T \end{aligned} \tag{3.10}$$

The transformation and rotation matrices can now be used to convert velocities from one frame to the other frame and vice versa. Thus, the quadrotor final kinematic model may be described in an inertial reference frame will be;

$$\begin{aligned}[\dot{x},\dot{y},\dot{z}]^T &= R^T_{(\phi,\theta,\psi)}[v_x,v_y,v_z]^T \\ [\dot{\phi},\dot{\theta},\dot{\psi}]^T &= T_{(\phi,\theta)}[\delta,\beta,\gamma]^T\end{aligned} \tag{3.11}$$

The quadrotor's kinematic model was created by multiplying the angular and linear speeds in the inertial frame;

$$\begin{aligned}\dot{x} &= v_x[c(\psi)c(\theta)] - v_y\left[s\left((\phi - c(\psi)s(\theta)s(\phi)\right] + v_z[s((\phi + c(\psi)s(\theta)c(\phi)\right] \\ \dot{y} &= v_x[s(\psi)c(\theta)] - v_y\left[c\left((\phi + s(\psi)s(\theta)s(\phi)\right] - v_z[c((\phi - s(\psi)s(\theta)c(\phi)\right] \\ \dot{z} &= -v_x[s(\theta)] + v_y\left[c\left(\theta s(\phi\right] + v_z[c(\theta)c(\phi)\right] \\ \dot{\phi} &= \delta + \beta[s(\phi)\tan(\theta)] + \gamma[c(\phi)\tan(\theta)] \\ \dot{\theta} &= \beta c(\phi) - \gamma s(\phi) \\ \dot{\psi} &= \beta[s(\phi)\sec(\theta)] + \gamma[c(\phi)\sec(\theta)]\end{aligned} \tag{3.12}$$

Where: sine, cosine, tangent, and secant are represented by the variables s, c, tan, and sec.

The body movement is determined by varying the angular speed of the four motors positioned on the four cross ("x") configuration, each of which is located distant from the center by length l, using the quadrotor's configuration and neglecting gyroscopic force. Thrust is represented by F_{thrust} and is produced by the idea that all motors provide an upward force in the z direction which counteracts the gravitational force.

$$F_{\text{thrust}} = k\left(\omega_1{}^2 + \omega_2{}^2 + \omega_3{}^2 + \omega_4{}^2\right) \tag{3.13}$$

The equivalent actuating torques are produced by each motor allowing the quadrotor to rotate about its own axes (x, y, z). The moment from the difference in angular speeds of motors 2 and 4, which are oriented perpendicular to the axis x, generates torque about the x axis.

$$\tau_{ac,x} = kl\left(\omega_4{}^2 - \omega_2{}^2\right) \tag{3.14}$$

The moment from the difference in the angular speeds of motors 3 and 1, which are oriented perpendicular to the axis y, generates the torque about the y axis.

$$\tau_{ac,y} = kl\left(\omega_3^2 - \omega_1^2\right) \tag{3.15}$$

Motors spinning in the same direction (M1, M3) and in the opposite direction (M2, M4) produce torque about the z axis.

$$\tau_{ac,z} = b\left(\omega_1^2 + \omega_3^2 - \omega_2^2 - \omega_4^2\right) \tag{3.16}$$

Where: The lift constant, arm length from center to each motor, and drag constant are denoted by k, l, and b, respectively. The moments of inertia for the quadrotor is calculated assuming a spherical dense mass M located at the center having radius of r and point masses m placed at a distance l on the four motors position. Here the quadrotor is symmetrical about all axes. The spherical mass (m) moment of inertia given by formula; $I = (2/5)\mathrm{mr}^2$, where r and m are the radius and mass of the sphere [19]. The total moment of inertia in 3D becomes;

$$J = \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 1 & 0 & J_z \end{bmatrix} \tag{3.17}$$

Where:

$$\begin{aligned} J_x &= (2/5)Mr^2 + 2ml^2 \\ J_y &= (2/5)Mr^2 + 2ml^2 \\ J_z &= (2/5)Mr^2 + 4ml^2 \end{aligned} \tag{3.18}$$

where l and m are the distance from center and mass of the sphere respectively.

The expression for the net linear forces and moments due to acceleration will be;

$$F_{\text{net}} = \begin{bmatrix} F_{\text{net},x} \\ F_{\text{net},y} \\ F_{\text{net},z} \end{bmatrix} = m \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \tag{3.19}$$

The translational force resulting from both rotational and translational acceleration in the body frame using Newton's second law is as follows:

$$F_{\text{net},B} = ma_t \tag{3.20}$$

Where: a_t is translational acceleration

$$a_t = \frac{dV_B}{dt} + \frac{d(\omega_B \times s)}{dt} \tag{3.21}$$

The time differential operator is simplified as follows: x represents the cross product and angular speeds maintained constant during translational motion at maneuver.

$$a_t = \dot{V}_B + \omega_B \times \dot{s} \tag{3.22}$$

The body frame's translational speed V_B, simplifies the translational acceleration and force.

$$\begin{aligned} a_t &= \dot{V}_B + \omega_B \times V_B \\ F_{net,B} &= m\left(\dot{V}_B + \omega_B \times V_B\right) \end{aligned} \tag{3.23}$$

$$F_{net,B} = \begin{bmatrix} F_{net,x} \\ F_{net,y} \\ F_{net,z} \end{bmatrix} = m \begin{bmatrix} \dot{V}_x \\ \dot{V}_y \\ \dot{V}_z \end{bmatrix} + m \begin{bmatrix} \delta \\ \beta \\ \gamma \end{bmatrix} \times \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \tag{3.24}$$

$$\begin{aligned} F_{net,x} &= m\left(\dot{V}_z + v_y\gamma - v_y\beta\right) \\ F_{net,y} &= m\left(\dot{V}_x + v_x\gamma - v_z\delta\right) \\ F_{net,z} &= m\left(\dot{V}_y + v_x\beta - v_x\delta\right) \end{aligned} \tag{3.25}$$

A rotating reference frame with its axes fixed to the quadrotor and parallel to its principal axis of inertia is used to apply Euler's equations to the rotation of a quadrotor. So, the equation yields the net torque:

$$\tau_{\text{net}} = I\dot{\omega}_B + \omega_B \times (I\omega_B) \tag{3.26}$$

$$\tau_{\text{net}} = \begin{bmatrix} \tau_{\text{net},x} \\ \tau_{\text{net},y} \\ \tau_{\text{net},z} \end{bmatrix} = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \dot{\delta} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} + \begin{bmatrix} \delta \\ \beta \\ \gamma \end{bmatrix} \times \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \delta \\ \beta \\ \gamma \end{bmatrix} \tag{3.27}$$

$$\begin{aligned} \tau_{\text{net},x} &= I_x\dot{\delta} + I_z\beta\gamma - I_y\beta\gamma \\ \tau_{\text{net},y} &= I_y\dot{\beta} + I_x\delta\gamma - I_z\delta\gamma \\ \tau_{\text{net},z} &= I_z\dot{\gamma} + I_y\delta\beta - I_x\delta\gamma \end{aligned} \tag{3.28}$$

All external forces as gravity, wind, and net acceleration force must be balanced for

the quadrotor to be controlled, neglecting the gyroscopic force and also holds true for external torques:

$$\begin{aligned} F_{\text{net}} &= F_{\text{applied}} - F_{\text{external}} \\ \tau_{\text{net}} &= \tau_{\text{applied}} - \tau_{\text{external}} \end{aligned} \tag{3.29}$$

The external force and torque are represented by the following equation, assuming that the applied forces are caused by gravity, thrust (actuating), and torque due to wind and actuation:

$$\begin{aligned} F_{\text{external}} &= F_{\text{mg}} + F_{\text{wind}} + F_{\text{thrust}} - F_{\text{net}} \\ \tau_{\text{external}} &= \tau_{\text{actuating}} + \tau_{\text{wind}} - \tau_{\text{net}} \end{aligned} \tag{3.30}$$

Where Equation 3.27 provides the actuating torque and thrust, while the wind component drag forces and torques are simply column vectors in (x, y, z) that are treated as disturbances and have the form of;

$$\begin{bmatrix} F_{\text{wind}} & \tau_{\text{wind}} \end{bmatrix}^T = \begin{bmatrix} F_{\text{wind},x} & F_{\text{wind},y} & F_{\text{wind},z} & \tau_{\text{wind},x} & \tau_{\text{wind},y} & \tau_{\text{wind},z} \end{bmatrix}^T \tag{3.31}$$

Using the rotation matrix and unit vector along the z axis from the inertial reference plane to the body frame converted by the equation. so, the applied force due to gravity will be:

$$\begin{aligned} F_{mg} &= mgR_{(\phi,\theta,\psi)}\hat{u}_z \\ F_{mg} &= \begin{bmatrix} -mgs(\theta) & mgc(\theta)s(\phi) & mgc(\theta)c(\phi) \end{bmatrix}^T \end{aligned} \tag{3.32}$$

A quadrotor needs to balance its net external forces and torques in order to execute a controlled maneuver ($F_{\text{ext}} = 0, \tau_{\text{ext}} = 0$). A comprehensive dynamic model is created, which includes force and torque equations resulting from wind, actuation, and acceleration;

$$\begin{aligned} m\left(\dot{v}_x - v_y\gamma + v_z\beta\right) &= -mg[s(\theta)] + F_{\text{wind},,x} \\ m\left(\dot{v}_y + v_x\gamma - v_z\delta\right) &= mg[c(\theta)s(\phi)] + F_{\text{wind,y}} \\ m\left(\dot{v}_z - v_x\beta + v_y\delta\right) &= mg[c(\theta)c(\phi)] + F_{\text{wind},z} - k\left(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2\right) \\ I_x\dot{\delta} + I_z\beta\gamma - I_y\beta\gamma &= kl\left(\omega_4^2 - \omega_2^2\right) + \tau_{\text{wind},x} \\ I_y\dot{\beta} + I_x\delta\gamma - I_z\delta\gamma &= kl\left(\omega_3^2 - \omega_1^2\right) + \tau_{\text{wind,y}} \\ I_z\dot{\gamma} + I_y\delta\beta - I_x\delta\gamma &= b\left(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2\right) + \tau_{\text{wind},z} \end{aligned} \tag{3.33}$$

3.4 Mathematical Model of Quadrotor

The three coordinate angles (ϕ, θ, ψ) and physical positions in the inertial frame (x, y, z) are then taken as state variables (total of 12) and their corresponding velocities $(\dot{\phi}, \dot{\theta}, \dot{\psi})$ and $(\dot{x}, \dot{y}, \dot{z})$ are then used to create the state vector. The acceleration variables will obtain the velocity variables from equations 3.10 and the state vector's rotational accelerations were obtained by removing the gyroscopic moments from the Newton-Euler equation.

$$\tau_{\text{actuating}} = J\dot{\omega} + \omega \times J\omega \tag{3.34}$$

Where: ω angular speed in inertial frame and J moment of inertia matrix from 3.17

$$\tau_{\text{actuating}} = \begin{bmatrix} \tau_{ac,x} \\ \tau_{ac,y} \\ \tau_{ac,z} \end{bmatrix} = \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{3.35}$$

Solving for $(\ddot{\phi}, \ddot{\theta}, \ddot{\psi})$

$$\begin{aligned} \ddot{\phi} &= \frac{\tau_{ac,x} + \left[J_y - J_z\right]\dot{\theta}\dot{\psi}}{J_x} \\ \ddot{\theta} &= \frac{\tau_{ac,y} + \left[J_z - J_x\right]\dot{\phi}\dot{\psi}}{J_y} \\ \ddot{\psi} &= \frac{\tau_{ac,z} + \left[J_x - J_y\right]\dot{\phi}\dot{\theta}}{J_z} \end{aligned} \tag{3.36}$$

The accelerations for the state vector obtained from the equation 3.20 using Newton's second law of motion in an inertial reference frame will be as follows:

$$ma_t = F_{mg} + R_{(\phi,\theta,\psi)}{}^T F_{\text{thrust}} \tag{3.37}$$

Where: F_{mg} force due to gravity in inertial frame and F_{thrust} actuating force

$$m\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R_{(\phi,\theta,\psi)}{}^T \begin{bmatrix} 0 \\ 0 \\ F_{\text{thrust}} \end{bmatrix} \tag{3.38}$$

Solving for $\ddot{x}, \ddot{y}, \ddot{z}$

$$\begin{aligned} \ddot{x} &= \frac{F_{\text{thrust}}\left[c(\psi)s(\theta)c(\phi)+s(\psi)s(\phi)\right]}{m} \\ \ddot{y} &= \frac{F_{\text{thrust}}\left[s(\psi)s(\theta)c(\phi)-c(\psi)s(\phi)\right]}{m} \\ \ddot{z} &= \frac{F_{\text{thrust}}\left[c(\theta)c(\phi)\right]}{m} - g \end{aligned} \tag{3.39}$$

As an underactuated system, the quadcopter has six degrees of freedom (DOF) represented by x, y, z, ϕ, θ, and ψ. It is controlled by four input variables, U_1, U_2, U_3, and U_4. The underactuated system must be converted to a fully actuated one in order to attain autonomous control. Only the propulsion and gravity forces are taken into account, assuming that the quadcopter is not affected by any other factors. The thrust force F_{thrust} is represented by U_1 in the body frame. A rotating matrix is used to move from the body frame to the inertial frame.

Consider the virtual control inputs in the translational motions, denoted as U_x, U_y, and U_z.

$$\begin{bmatrix} U_x \\ U_y \\ U_z \end{bmatrix} = \begin{bmatrix} c(\theta)c(\psi) & s(\phi)s(\theta)c(\psi)-c(\phi)s(\psi) & s(\phi)s(\psi)+c(\phi)s(\theta)c(\psi) \\ c(\theta)s(\psi) & c(\theta)c(\psi)+s(\phi)s(\theta)s(\psi) & -s(\phi)c(\psi)+c(\phi)s(\theta)s(\psi) \\ -s(\theta) & s(\phi)c(\theta) & c(\phi)c(\theta) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{3.40}$$

$$\begin{bmatrix} U_x \\ U_y \\ U_z \end{bmatrix} = \begin{bmatrix} (s(\phi)s(\psi)+c(\phi)s(\theta)c(\psi))U_1 \\ (-s(\phi)c(\psi)+c(\phi)s(\theta)s(\psi))U_1 \\ (c(\phi)c(\theta))U_1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{3.41}$$

Finding the magnitude of the vector:

$$U_1 = \sqrt{U_x^2 + U_y^2 + (U_z + mg)^2} \tag{3.42}$$

Since:

$$\begin{aligned} \mathrm{U}_x &= (\sin(\phi)\sin(\psi)+\cos(\phi)\sin(\theta)\cos(\psi))U_1 \\ \mathrm{U}_y &= (-\sin(\phi)\cos(\psi)+\cos(\phi)\sin(\theta)\sin(\psi))U_1 \\ \mathrm{U}_z &= (\cos(\phi)\cos(\theta))U_1 - mg \end{aligned} \tag{3.43}$$

So,

$$\ddot{X} = \frac{U_x}{m}, \tag{3.44}$$

$$\ddot{Y} = \frac{U_y}{m}, \tag{3.45}$$

$$\ddot{Z} = \frac{U_z}{m} \tag{3.46}$$

Four control inputs, represented by $\mathbf{U} = [U_1, U_2, U_3, U_4]^T$, affect the quadcopter. In particular, U_1 regulates altitude, which in turn affects lift force. The quadcopter's U_2 affects its roll angle, U_3 affects its pitch angle, and U_4 determines its yaw angle. The equations of motion are taken into consideration. Thus, we obtain:

$$\begin{aligned}
\ddot{\phi} &= \dot{\psi}\dot{\theta}\left(\frac{J_y - J_z}{J_x}\right) + \frac{l}{J_x}U_2 \\
\ddot{\theta} &= \dot{\psi}\dot{\phi}\left(\frac{J_z - J_x}{J_y}\right) + \frac{l}{J_y}U_3 \\
\ddot{\psi} &= \dot{\theta}\dot{\phi}\left(\frac{J_x - J_y}{J_z}\right) + \frac{1}{J_z}U_4 \\
\ddot{z} &= (\cos\phi\cos\theta)\frac{1}{m}U_1 - g \\
\ddot{x} &= (\cos\phi\cos\psi\sin\theta + \sin\phi\sin\psi)\frac{1}{m}U_1 \\
\ddot{y} &= (\cos\phi\sin\psi\sin\theta - \sin\phi\cos\psi)\frac{1}{m}U_1
\end{aligned} \tag{3.47}$$

Lets define the control signals be the actuating force and torques;

$$U = [U_1, U_2, U_3, U_4]^T = \left[F_{thrust}, \tau_{ac,x}, \tau_{ac,y}, \tau_{ac,z}\right]^T \tag{3.48}$$

The system inputs are U_1, U_2, U_3, U_4 , obtaining:

$$\begin{aligned}
U_1 &= k\left({\omega_1}^2 + {\omega_2}^2 + {\omega_3}^2 + {\omega_4}^2\right) \\
U_2 &= kl\left({\omega_4}^2 - {\omega_2}^2\right) \\
U_3 &= kl\left(\omega_3^2 - \omega_1^2\right) \\
U_4 &= b\left(\omega_1^2 + \omega_3^2 - \omega_2^2 - \omega_4^2\right)
\end{aligned} \tag{3.49}$$

The quadcopter can reach a specified altitude and attitude with just four fundamental movements. The propellers' angular velocity ω_i, which is naturally controlled by the propellers' rotations per minute, affects the movement vector. When considering aerody-

namic effects, it suggests that forces and moments are equivalent to the squared angular velocities of the propellers. The movement vector can be expressed as the product of the propellers' squared angular velocities vector ω_i^2 from equation 3.49 and the movement matrix.

The rotor's angular velocity, can be described inversely as:

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_2 \end{bmatrix} = \begin{bmatrix} k & k & k & k \\ 0 & -kl & 0 & kl \\ -kl & 0 & kl & 0 \\ b & -b & b & -b \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{3.50}$$

The squared angular velocities of the propellers will be;

$$\begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} = \begin{bmatrix} k & k & k & k \\ 0 & -kl & 0 & kl \\ -kl & 0 & kl & 0 \\ b & -b & b & -b \end{bmatrix}^{-1} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \tag{3.51}$$

3.4.1 State Space Representation

A state-space representation is a technique that uses state variables and first-order differential equations to describe the dynamics of a physical system. A quadcopter's state is a set of characteristics, including position, velocity, direction, and angular velocity, that characterize its condition at a given moment. The quadcopter's state is made up of these state variables, which include things like the position's X, Y, and Z coordinates, as well as the orientation's roll, pitch, and yaw angles and their corresponding derivatives [21]. Let the state variables be: $X, \dot{X}, Y, \dot{Y}, Z, \dot{Z}, \phi, \dot{\phi}, \theta, \dot{\theta}, \psi$, and $\dot{\psi}$, denoted as X_1, X_2, X_3, X_4, $X_5, X_6, X_7, X_8, X_9, X_{10}, X_{11}, X_{12}$

$$a_1 = \frac{J_y - J_z}{J_x}, \quad a_2 = \frac{W_r}{J_x}, \quad b_1 = \frac{1}{J_x}$$
$$a_3 = \frac{J_z - J_x}{J_y}, \quad a_4 = \frac{W_r}{J_y}, \quad b_2 = \frac{1}{J_y}$$
$$a_5 = \frac{J_x - J_y}{J_z}, \quad b_3 = \frac{1}{J_z}$$

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

$$\begin{aligned}
\dot{X}_1 &= X_2 \\
\dot{X}_2 &= U_x/m \\
\dot{X}_3 &= X_4 \\
\dot{X}_4 &= U_y/m \\
\dot{X}_5 &= X_6 \\
\dot{X}_6 &= U_z/m \\
\dot{X}_7 &= X_8 \\
\dot{X}_8 &= b_1 U_2 + a_1 X_{10} X_{12} + a_2 X_{10} \\
\dot{X}_9 &= X_{10} \\
\dot{X}_{10} &= b_2 U_3 + a_3 X_8 X_{12} - a_4 X_8 \\
\dot{X}_{11} &= X_{12} \\
\dot{X}_{12} &= b_3 U_4 + a_5 X_8 X_{10}
\end{aligned}$$

3.5 Model Verification

Model verification for a quadrotor involves systematically ensuring that its mathematical representation accurately reflects real-world dynamics and behavior. This process includes defining the quadrotor motion equations, simulating its performance by simply applying control signals of altitude and attitude control inputs and comparing simulation results. Measure the propeller speeds based on the control inputs and analyse real dynamics of quadcopter with those speed results. AND validated through MATLAB Software which is essential for demonstrating the model's reliability and limitations.

The designed state space model must be tested with zero initial condition and external input before trying to see the system performance with the designed control strategy [22]. The results shows that the model is valid and now it's able to go for the designed ANFIS-SMC performance evaluation.

The image illustrated in Figure 3.7 presents a block diagram representing the dynamic model of a UAV (Unmanned Aerial Vehicle) system, broken down into several interconnected subsystems. It begins with the Control Input, as described in equation 3.50, which influences the Rotational System in equation 3.36. This subsystem outputs the thrust force and torques, which affect the rotational rates. These rates are integrated

to obtain the Euler angles in the Kinematic Equation block, defined in equation 3.8. The angles and rates of the motion are then passed to the Translational System, described in equation 3.39, which computes the linear accelerations and, through double integration, yields the position coordinates.

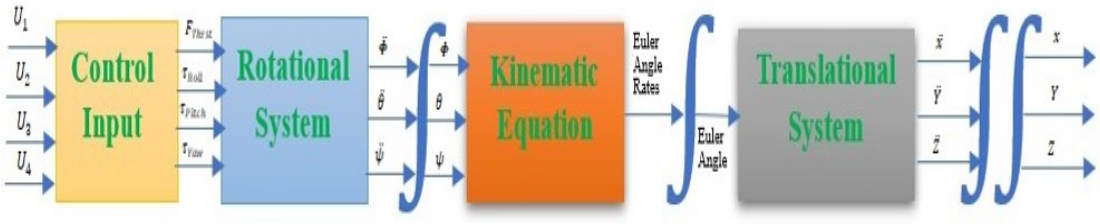


Figure 3.7: Mathematical Model

For simulation of the system a set of parameters and constants needed [23], they are taken from literature survey.

Parameter	Symbol	Value	SI Unit
Mass	m	0.486	Kg
Earth Gravity	g	9.81	$m.s^2$
Arm length	l	0.25	m
Lift Constant	k	$2.9842^{*}10^{-5}$	$N.s^2$
Drag Constant	b	$3.232^{*}10^{-7}$	$N.m.s^2$
Inertia in x	J_x	$3.8278^{*}10^{-3}$	$Kg.m^2$
Inertia in y	J_y	$3.8288^{*}10^{-3}$	$Kg.m^2$
Inertia in z	J_z	$7.6566^{*}10^{-3}$	$Kg.m^2$

Table 3.1: Parameters and Constants

Thrust (Hovering state): movement along z- axis, which is provide by applying altitude control signal U_1, which leads rotating all the motors at the same speed.If the value of trust T = mg , then the quadcopter stays on same height above the ground. when U_1=mg =4.76766 U_2=U_3=U_4=0 and initial states of the quadcopter is at 6m above the ground on hovering state. Then the measured speed is w_1 = w_2 = w_3 = w_4 = 210rad/s. All angles stay on zero state.

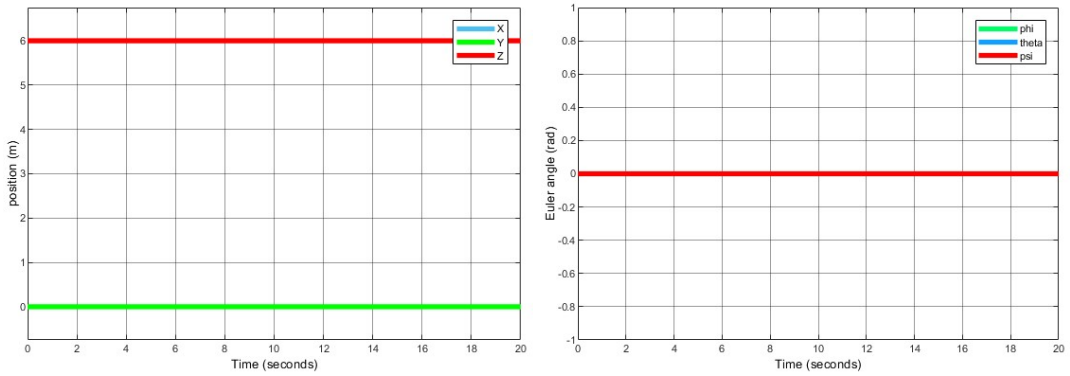


Figure 3.8: Model verification for hovering state output

Upward movement:- where the value of thrust T > mg , the vehicle goes up towards positive z- axis. when U_1= 6N,U_2=U_3=U_4=0 and initial states of the quadcopter is at 0m. Then the measured speed is w_1 = w_2 = w_3 = w_4 =225.3rad/s. All angles stay on zero state.

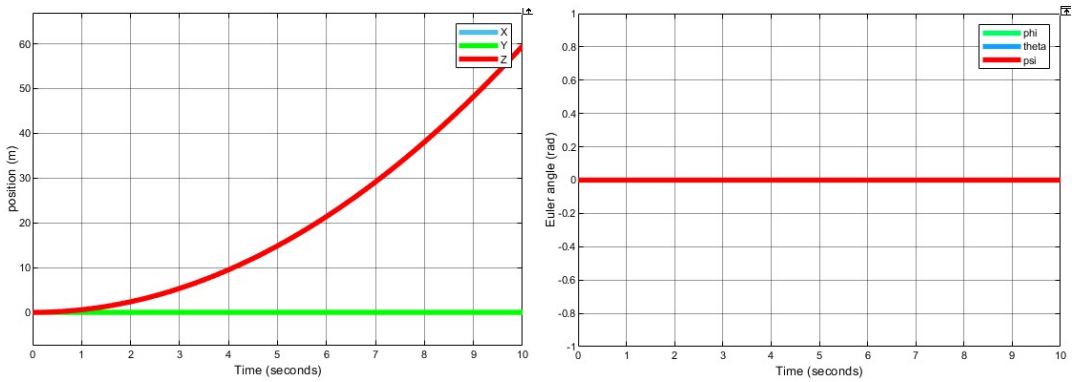


Figure 3.9: Model verification for Upward movement

Downward state: - The value of thrust T < mg , the vehicle fails down towards negative z- axis. when U_1= 4N,U_2=U_3=U_4=0 and initial states of the quadcopter is at 6m above the ground. Then the measured speed is w_1 = w_2 = w_3 = w_4 = 203.7rad/s. All angles stay on zero state.

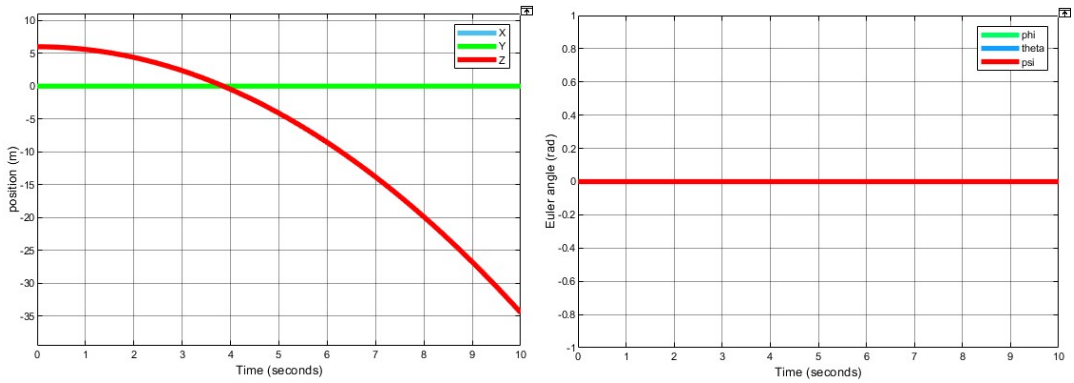


Figure 3.10: Model verification for Downward movement

Roll:-movement of the quadcopter towards Y-axis and Z-axis, When a control signal U_2 = 0.0001, U_1 = 4.76766N and U_3=U_4=0 and initial states of the quadcopter is at 6m above the ground is applied Which leads the speed w_1 = w_2= 205.781rad/s and w_3 = w_4 = 205.768 rad/s will cause roll angle change.The quadcopter moves down ward along negative z-axis and towards negative y-axis and x-axis remains zero.

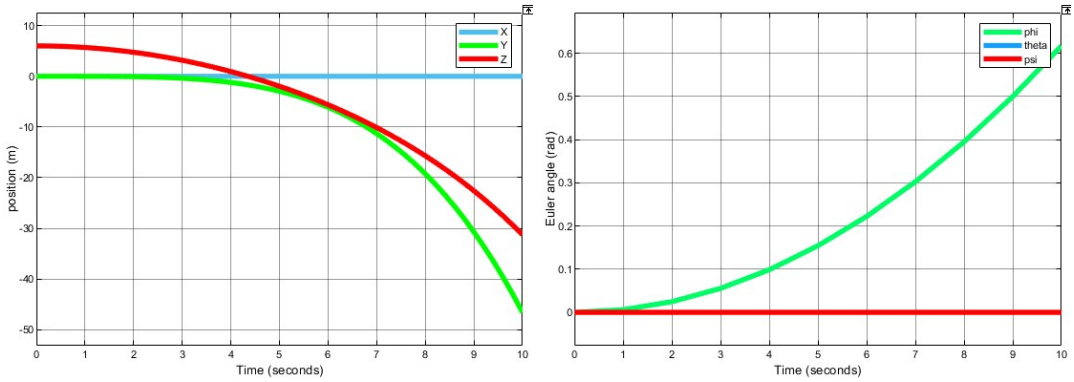


Figure 3.11: Model verification for Roll movement

Pitch:-movement of the quadcopter towards Y-axis and Z-axis, When we apply U_3= 0.0001, U_1 = 4.76766N and U_2=U_4=0 and initial states of the quadcopter is at 6m above the ground Which leads an increasing the speed w_1 = w_4 =205.781rad/s and decreasing w_2 = w_3 = 205.768rad/s speeds at the same time will cause pitch angle change.The quadcopter moves down ward along negative z-axis and towards positive x-axis and y-axis remains zero and will cause the change of Pitch angle

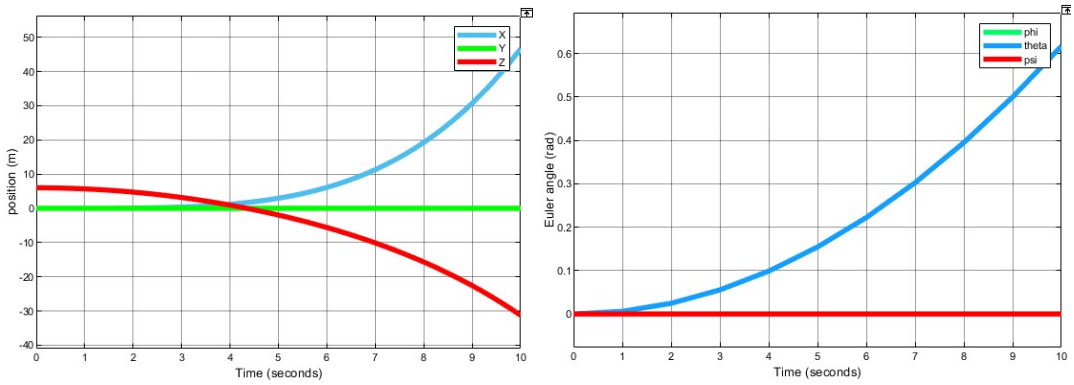


Figure 3.12: Model verification for Pitch movement

Yaw:- When we apply U_4 = 0.0001, U_2=U_3=0 U_1 = 4.76766NN and initial states of the quadcopter is at 6m above the ground Which leads an increasing the speed of w_1 = w_3 = 205.8rad/s and decreasing the speed of w_2 = w_4 = 205.7rad/s.The positions are not affected and The quadcopter rotates CW.

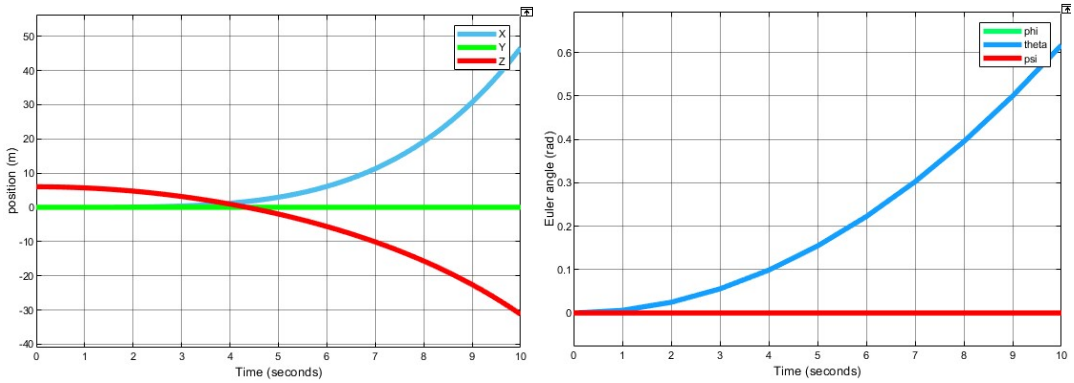


Figure 3.13: Model verification for Yaw movement

Chapter 4

Controller Design

4.1 Introduction

This chapter presents the design of neuro fuzzy system tuning sliding mode control for quad copter UAV. The proposed controller will able to handle uncertainty and imprecision which may appear in the system. A non-linear system's model imprecision may result from a conscious choice to simplify the system's dynamics or from actual system uncertainty. There are two primary categories of modeling imprecision: organized and unstructured uncertainty. Model error will give adverse influence on overall control system performance and this error can be removed by robust control approaches which can deal with non-linearity, uncertainty, and disturbance.

Sliding Mode Control (SMC) is a robust nonlinear control technique designed to ensure stability and consistent performance [24]. It works by forcing the system to stay on the sliding surface through the use of a switching control input. Since SMC a combination of discontinuous control mode transitions with continuous state-space evolution, it can be viewed as a specific case of a hybrid dynamical system. The control law is formulated as a discontinuous function of time, allowing the system to transition between multiple continuous control modes based on its state at any given moment. As a result, SMC is classified as a variable structure control method within modern control theory.

In recent years intelligent controllers are becoming the part of quadrotor control for stability to the system , which might yield excellent results. Fuzzy logic and Neural networks are getting wide acceptance in quadcoptor control due to their inherent advantages. An main advantage of neural networks is their capacity for self-adaptation and learning. In a similar way, fuzzy logic is the unique capacity to use fuzzy if-then rules

to account for the inherent imprecision and uncertainty of real systems. This hybrid system, known as the Adaptive network based fuzzy inference system (ANFIS), is an integrated control strategy that combines fuzzy logic and neural networks. The adaptive neural fuzzy tuning Sliding Mode Controller concept for quadcopter trajectory tracking was presented in this chapter. After learning its behavior, the Adaptive Neuro-Fuzzy Inference System is meant to adjust the conventional Sliding Mode Controller in the suggested control architecture. In particular, the system creates the input/output training data for ANFIS using the tracking error as input and the control signal generated by the SMC as output. After training, the control duty is maximized by ANFIS, which efficiently executes the SMC control law's response while greatly minimizing the chattering effect that is usually connected to SMC. The quadcopter's position, altitude, and attitude may be controlled more smoothly thanks to this substitution, which also enables control input adaption in response to shifting environmental conditions. The control scheme is illustrated in Figure 4.1

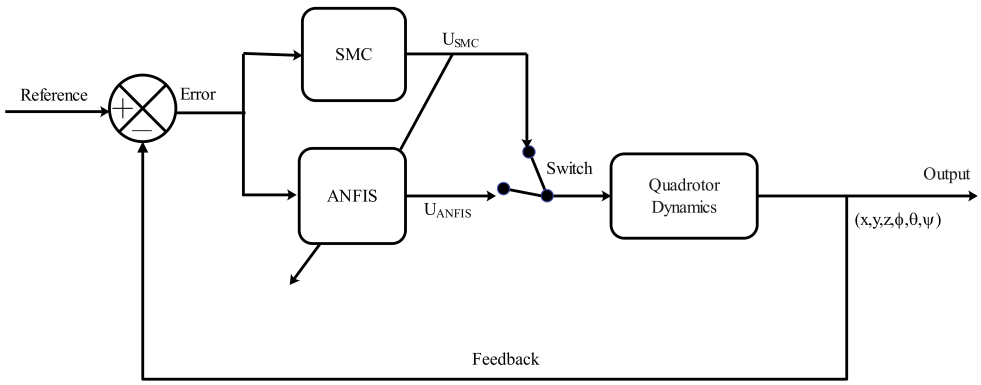


Figure 4.1: Control scheme of the quadcopter system

Jang proposed a complementary use of adaptive neuro-fuzzy inference system (ANFIS) controller in 1995 [25] to improve the conventional SMC. This strategy is simple to implement with ANFIS Toolbox, and ANFIS control does not require a mathematical model of the system [26]. Combining SMC with ANFIS control is one method to enhance its performance and the design of a sliding mode controller with ANFIS control results in a simpler control scheme, less chattering, less harmonic distortion of voltage, and increased resilience to non-linearities and disturbances.

4.2 Sliding Mode Controller Design

SMC systems are made to force the system states onto a specific sliding surface which a surface in the state space. Once the sliding surface is reached, sliding mode control keeps the states on the close system of the sliding surface. Selecting a control rule that will make the switching surface attractive to the system state comes after designing a sliding surface to guarantee that the motion conforms with design specifications.

The benefits of sliding mode control includes, the sliding function selection may be used to modify the system's dynamic behavior and in response to some specific uncertainty, the closed loop reaction becomes completely insensitive. This theory also applies to bounded non-linearity, disturbance, and uncertainties in model parameters. Only when the system state fulfills the dynamic equation that controls the sliding mode indefinitely does there exist an ideal sliding mode. One of the most effective methods for creating reliable controllers for high-order, nonlinear dynamic system is operating in uncertain environments with the sliding mode control approach. The issue of stability and reliable performance can be approached methodically with sliding controller design [27].

Sliding mode controllers have the advantage of not requiring precise modeling since they are resistant to disturbances and parameter changes once they enter the sliding mode. Two steps in the SMC design process are to design a surface that is created to order and when a plant is on a sliding surface, its dynamics are constrained by the surface equations and are resilient to both outside disturbances and plant uncertainty and to develop a feedback control rule that will allow the system's trajectory to converge at the sliding surface, resulting in the sliding surface being acquired within a certain time frame. The sliding mode refers to the way the system moves on the sliding surface and stay on the sliding surface, S(t), which represents as;

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

where s is the sliding variable, λ is a gain defined by the designer and e is the error.

The control law in which the process model and sliding condition are combined to create the continuous portion of the controller and the nonlinear discontinuous portion symbolizes the control law's switching component. The control gain determines the level of aggression needed to reach the sliding surface; nevertheless, excess force on the part of the controller may contribute to chattering as shown on the figure below.

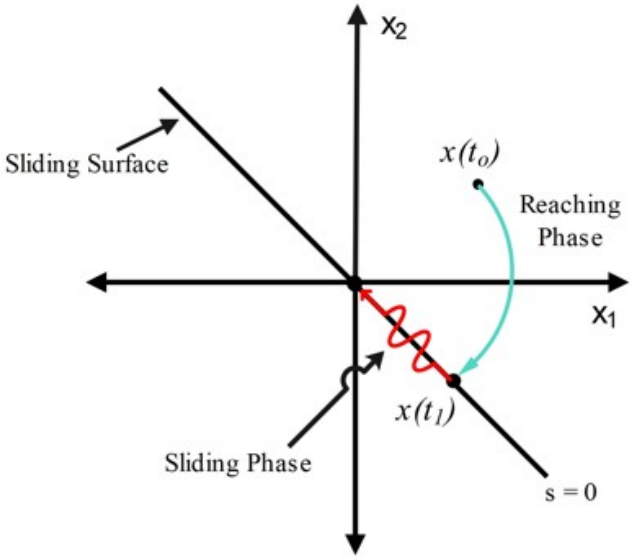


Figure 4.2: Sliding Mode Controller

A trajectory begins its reaching phase at a nonzero initial condition and moves toward a sliding surface. The sliding surface where the trajectory, once it reaches it, stays there forever and changes in accordance with the dynamics that the sliding surface specifies. The trajectories go along the switching function in theory. Because of high frequency switching in which states regularly traverse the surface rather than remain on it—there is no practical ideal sliding mode. Techniques to reduce chattering and maintain the trajectories inside a limited region of the surface have been developed.

Here are some advantages and disadvantages:

Advantages

- SMC is highly robust against uncertainties and disturbances. It can handle variations in system dynamics and external disturbances effectively.
- The control law can maintain performance despite variations in system parameters, making it suitable for systems with unknown or time-varying parameters.
- SMC can provide quick response times due to the sliding mode, which drives the system states to the desired trajectory rapidly.

Disadvantages

- A significant drawback of SMC is high-frequency oscillations around the sliding

surface. This can lead to wear and tear in mechanical systems and undesirable behavior.

- Designing an appropriate sliding surface can be complex and may require expert knowledge of the system dynamics.
- While SMC can achieve finite-time convergence, the speed of convergence can be influenced by the choice of control parameters, which may require fine-tuning.

The sliding mode controller is designed due to there tolerant to external disturbances, parameter variation and hence robust and handles all system non-linearity. In order to mitigate the chattering issue caused by the interaction between the sliding manifold and discontinuous version controller, the continuous version of SMC is constructed in this study using the Lyapunov and reaching law approaches. The state space must be transformed into error dynamics to drive the SMC error dynamics toward zero with great responsiveness and resilience, hence enforcing quick convergence.

The position error and its derivative as:-

$$\begin{bmatrix} \mathbf{e}_x \\ \mathbf{e}_y \\ \mathbf{e}_z \end{bmatrix} = \begin{bmatrix} \mathbf{x}_d - \mathbf{x} \\ \mathbf{y}_d - \mathbf{y} \\ \mathbf{z}_d - \mathbf{z} \end{bmatrix} \tag{4.2}$$

To implement SMC to quadcopter dynamics, the sliding surfaces x, y, z directions are defined respectively as:

$$\begin{cases} s_x = a_1 e_x + \dot{e_x} \\ s_y = a_2 e_y + \dot{e}_y \\ s_z = a_3 e_z + \dot{e}_z \end{cases} \tag{4.3}$$

After choosing the sliding surface, the control law can be calculated. This control law requires that the system converge to the sliding surface. We can obtain from $\dot{s}_i = -ksgn(s_i)$ the discontinuous control law, where $i = x, y, z$.

$$\begin{aligned} \dot{s}_x &= a_1 \dot{e}_x + \ddot{e}_x = -k_1 \operatorname{sgn}(s_x), \\ \dot{s}_y &= a_2 \dot{e}_y + \ddot{e}_y = -k_2 \operatorname{sgn}(s_y), \\ \dot{s}_z &= a_3 \dot{e}_z + \ddot{e}_z = -k_3 \operatorname{sgn}(s_z). \end{aligned} \tag{4.4}$$

The virtual contol U_x, U_y, and U_z:

$$\begin{aligned}
\frac{U_x}{m} &= k_1 \operatorname{sgn}(s_x) + a_1 \dot{e}_x + \ddot{x}_d \\
\frac{U_y}{m} &= k_2 \operatorname{sgn}(s_y) + a_2 \dot{e}_y + \ddot{y}_d \\
\frac{U_z}{m} &= k_3 \operatorname{sgn}(s_z) + a_3 \dot{e}_z + \ddot{z}_d
\end{aligned} \tag{4.5}$$

4.2.1 Attitude Control

With the Euler angle representation, we will develop the equations governing the angular rates in order to obtain the sliding mode control equations for a quadcopter attitude, and position dynamics. All of the controllers will be shown, including those for pitch (θ), yaw (ψ), roll (ϕ), and x, y, z, and roll (ϕ). Let us define the attitude error and its derivative as:-

$$\begin{bmatrix} \mathbf{e}_\phi \\ \mathbf{e}_\theta \\ \mathbf{e}_\psi \end{bmatrix} = \begin{bmatrix} \phi_d - \phi \\ \theta_d - \theta \\ \psi_d - \psi \end{bmatrix} \tag{4.6}$$

To implement SMC to quadcopter dynamics, the sliding surfaces ϕ, θ, ψ directions are defined respectively as.

$$\begin{cases} s_\phi = a_4 e_\phi + \dot{e_\phi} \\ s_\theta = a_5 e_\theta + \dot{e}_\theta \\ s_\psi = a_6 e_\psi + \dot{e}_\psi \end{cases} \tag{4.7}$$

The control law will be implemented by selecting the sliding surface. The system must converge to the sliding surface under this control law. We may obtain $\dot{s}_i = -k_1 sgn(s_i)$ from the corresponding control law, where $i = \phi, \theta, \psi$.

$$\begin{aligned}
\dot{s}_\phi &= a_4 \dot{e}_\phi + \ddot{e}_\phi = -k_4 \operatorname{sgn}(s_\phi), \\
\dot{s}_\theta &= a_5 \dot{e}_\theta + \ddot{e}_\theta = -k_5 \operatorname{sgn}(s_\theta), \\
\dot{s}_\psi &= a_6 \dot{e}_\psi + \ddot{e}_\psi = -k_6 \operatorname{sgn}(s_\psi).
\end{aligned} \tag{4.8}$$

We can get the control effort U_2, U_3, and U_4 from equation 3.47, **??** and 4.8

4.2.1.1 Roll Controller

From 3.47, the roll angular acceleration is defined as:

$$\ddot{\phi} = \frac{U_2 + \left[J_y - J_z\right]\dot{\theta}\dot{\psi}}{J_x}$$

Let the tracking error be given by $e_\phi = \phi_d - \phi$, and define the state variables $\phi_1 = e_\phi$, $\phi_2 = \dot{e}_\phi$. The error dynamics in state-space form become:

$$\begin{aligned} \dot{\phi}_1 &= \phi_2 \\ \dot{\phi}_2 &= \ddot{\phi}_d - \frac{U_2 + \left[J_y - J_z\right]\dot{\theta}\dot{\psi}}{J_x} \end{aligned} \tag{4.9}$$

Define the sliding surface for some positive constant $a_4 > 0$ as:

$$S = a_4\phi_1 + \phi_2$$

Reaching Law: Imposing $\dot{S} = 0$, we derive:

$$\begin{aligned} \dot{S} &= a_4\dot{\phi}_1 + \dot{\phi}_2 = a_4\phi_2 + \ddot{\phi}_d - \frac{U_2 + \left[J_y - J_z\right]\dot{\theta}\dot{\psi}}{J_x} = 0 \\ U_2 &= \left[J_z - J_y\right]\dot{\theta}\dot{\psi} + J_x\left[a_4\phi_2 + \ddot{\phi}_d\right] \end{aligned} \tag{4.10}$$

Lyapunov Analysis: Choosing the control input as $U_2 = -K_\phi\,\mathrm{sgn}(S)$ and candidate Lyapunov function $V = \frac{S^2}{2}$, we have:

$$\dot{V} = S\dot{S} = S\left(a_4\phi_2 + \ddot{\phi}_d - \frac{-K_\phi\,\mathrm{sgn}(S) + \left[J_y - J_z\right]\dot{\theta}\dot{\psi}}{J_x}\right) < 0 \tag{4.11}$$

Asymptotic stability is ensured when:

$$K_\phi < \frac{\left[J_y - J_z\right]\dot{\theta}\dot{\psi} - J_x\ddot{\phi}_d}{\mathrm{sgn}(S)}$$

provided $\mathrm{sgn}(S) > 0$.

$$U_2 = -K_\phi\,\mathrm{sgn}(S) + \left[J_z - J_y\right]\dot{\theta}\dot{\psi} + J_x\left[a_4\dot{e}_\phi + \ddot{\phi}_d\right] \tag{4.12}$$

4.2.1.2 Pitch Controller

Referencing 3.47, the pitch dynamics are:

$$\ddot{\theta} = \frac{U_3 + [J_z - J_x]\dot{\phi}\dot{\psi}}{J_y}$$

With error $e_\theta = \theta_d - \theta$, define the state variables as $\theta_1 = e_\theta$, $\theta_2 = \dot{e}_\theta$. The system in state-space becomes:

$$\begin{aligned} \dot{\theta}_1 &= \theta_2 \\ \dot{\theta}_2 &= \ddot{\theta}_d - \frac{U_3 + [J_z - J_x]\dot{\phi}\dot{\psi}}{J_y} \end{aligned} \tag{4.13}$$

Let the sliding surface be $S = a_5\theta_1 + \theta_2$, for $a_5 > 0$.

Reaching Law:

$$\begin{aligned} \dot{S} &= a_5\dot{\theta}_1 + \dot{\theta}_2 = a_5\theta_2 + \ddot{\theta}_d - \frac{U_3 + [J_z - J_x]\dot{\phi}\dot{\psi}}{J_y} = 0 \\ U_3 &= [J_x - J_z]\dot{\phi}\dot{\psi} + J_y\left[a_5\theta_2 + \ddot{\theta}_d\right] \end{aligned} \tag{4.14}$$

Lyapunov Function: With $U_3 = -K_\theta \operatorname{sgn}(S)$ and $V = \frac{S^2}{2}$, the derivative becomes:

$$\dot{V} = S\dot{S} = S\left(a_5\theta_2 + \ddot{\theta}_d - \frac{-K_\theta \operatorname{sgn}(S) + [J_z - J_x]\dot{\phi}\dot{\psi}}{J_y}\right) < 0 \tag{4.15}$$

The stability condition is:

$$K_\theta < \frac{[J_z - J_x]\dot{\phi}\dot{\psi} - J_y\ddot{\theta}_d}{\operatorname{sgn}(S)}, \quad \text{when } \operatorname{sgn}(S) > 0$$

$$U_3 = -K_\theta \operatorname{sgn}(S) + [J_x - J_z]\dot{\phi}\dot{\psi} + J_y\left[a_5\dot{e}_\theta + \ddot{\theta}_d\right] \tag{4.16}$$

4.2.1.3 Yaw Controller

According to 3.47, the yaw acceleration is:

$$\ddot{\psi} = \frac{U_4 + \left[J_x - J_y\right]\dot{\phi}\dot{\theta}}{J_z}$$

Given the desired yaw ψ_d, and error $e_\psi = \psi_d - \psi$, define states as $\psi_1 = e_\psi$, $\psi_2 = \dot{e}_\psi$:

$$\begin{aligned} \dot{\psi}_1 &= \psi_2 \\ \dot{\psi}_2 &= \ddot{\psi}_d - \frac{U_4 + \left[J_x - J_y\right]\dot{\phi}\dot{\theta}}{J_z} \end{aligned} \tag{4.17}$$

Define sliding surface $S = a_6\psi_1 + \psi_2$, where $a_6 > 0$.

Reaching Law:

$$\begin{aligned} \dot{S} &= a_6\psi_2 + \ddot{\psi}_d - \frac{U_4 + \left[J_x - J_y\right]\dot{\phi}\dot{\theta}}{J_z} = 0 \\ U_4 &= \left[J_y - J_x\right]\dot{\phi}\dot{\theta} + J_z\left[a_6\psi_2 + \ddot{\psi}_d\right] \end{aligned} \tag{4.18}$$

Lyapunov Stability: With control input $U_4 = -K_\psi \operatorname{sgn}(S)$, the Lyapunov derivative becomes:

$$\dot{V} = S\dot{S} = S\left(a_6\psi_2 + \ddot{\psi}_d - \frac{-K_\psi \operatorname{sgn}(S) + \left[J_x - J_y\right]\dot{\phi}\dot{\theta}}{J_z}\right) < 0 \tag{4.19}$$

Stability is ensured if:

$$K_\psi < \frac{\left[J_x - J_y\right]\dot{\phi}\dot{\theta} - J_z\ddot{\psi}_d}{\operatorname{sgn}(S)}, \quad \text{provided } \operatorname{sgn}(S) > 0$$

$$U_4 = -K_\psi \operatorname{sgn}(S) + \left[J_y - J_x\right]\dot{\phi}\dot{\theta} + J_z\left[a_6\dot{e}_\psi + \ddot{\psi}_d\right] \tag{4.20}$$

4.2.2 Altitude Control

According to 3.47, the vertical acceleration is given by:

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

With desired height Z_d, define the error $e_z = Z_d - Z$, and set $Z_1 = e_z$, $Z_2 = \dot{e}_z$. The system dynamics become:

$$\begin{aligned} \dot{Z}_1 &= Z_2 \\ \dot{Z}_2 &= \ddot{Z}_d - g - \frac{U_1\cos(\theta)\cos(\phi)}{m} \end{aligned} \tag{4.21}$$

Define the sliding surface for $a_3 > 0$ as $S = a_3Z_1 + Z_2$.

Reaching Law:

$$\begin{aligned} \dot{S} &= a_3 Z_2 + \ddot{Z}_d - g + \frac{U_1 \cos(\theta)\cos(\phi)}{m} = 0 \\ U_1 &= m(g - a_3 Z_2 - \ddot{Z}_d) \end{aligned} \tag{4.22}$$

Lyapunov Analysis:

$$\dot{V} = S\dot{S} = S\left(a_3 Z_2 + \ddot{Z}_d - g + \frac{U_1 \cos(\theta)\cos(\phi)}{m}\right) < 0 \tag{4.23}$$

For stability:

$$K_Z > \frac{m(\ddot{Z}_d - g)}{\text{sgn}(S)\cos(\theta)\cos(\phi)}$$

$$U_1 = -K_Z \,\text{sgn}(S_z) + m(g - a_3 \dot{e}_z - \ddot{Z}_d) \tag{4.24}$$

4.2.3 X and Y Motion Control

Unlike altitude and rotational motion, the horizontal positions x and y are not directly manipulated by control inputs U_1 to U_4. Instead, they are indirectly controlled via desired angles ϕ_d and θ_d, derived from the translational dynamics in **??** and **??**.

Under small-angle approximations (around hover), $\sin\phi_d \approx \phi_d$, $\sin\theta_d \approx \theta_d$, $\cos\phi_d \approx \cos\theta_d \approx 1$, we obtain:

$$\ddot{x} = \left(\cos\phi\cos\psi\sin\theta + \sin\phi\sin\psi\right)\frac{U_1}{m} \tag{4.25}$$

$$\ddot{y} = \left(\cos\phi\sin\psi\sin\theta - \sin\phi\cos\psi\right)\frac{U_1}{m} \tag{4.26}$$

Rewriting:

$$(\ddot{x})\frac{m}{U_1} = \phi_d \sin\psi + \theta_d \cos\psi \tag{4.27}$$

$$(\ddot{y})\frac{m}{U_1} = \theta_d \sin\psi - \phi_d \cos\psi \tag{4.28}$$

In matrix form:

$$\begin{bmatrix} \sin\psi & \cos\psi \\ -\cos\psi & \sin\psi \end{bmatrix} \begin{bmatrix} \phi_d \\ \theta_d \end{bmatrix} = \frac{m}{U_1} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} \tag{4.29}$$

Solving:

$$\begin{bmatrix} \phi_d \\ \theta_d \end{bmatrix} = \frac{m}{U_1} \begin{bmatrix} \sin\psi & -\cos\psi \\ \cos\psi & \sin\psi \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} \tag{4.30}$$

Thus:

$$\phi_d = \frac{m}{U_1}[\sin\psi \cdot \ddot{x} - \cos\psi \cdot \ddot{y}] \tag{4.31}$$

$$\theta_d = \frac{m}{U_1}[\cos\psi \cdot \ddot{x} + \sin\psi \cdot \ddot{y}] \tag{4.32}$$

4.3 Artificial Neural Network Control

An artificial neural network (ANN) is a computer model that draws inspiration from the neural architecture of the human brain. It is made up of layers of interconnected nodes, or neurons. These nodes allow information to flow, and during training, the network modifies the connection strengths (weights) to learn from the input. This allows the network to identify patterns, anticipate outcomes, and perform a variety of machine learning and artificial intelligence tasks [29]. The human brain is not only the source of inherent intelligence, but it is also capable of processing incomplete information received through perception relatively quickly. The neural network evolved as a result of attempts by researchers to simulate the human brain. Here, the brain has been simulated using a continuous time non-linear dynamic system with a link architecture. The weighted connections between the neurons, or processing units, in this architecture are meant to resemble the structure of the human brain.As a result, by modifying the connections between the layers, the neural network can learn and adapt. The most important characteristics of the neural network are:

- Model non-linearities and provide better control.
- Learn to compensate disturbances by adjusting control outputs dynamically based on real-time sensor data.
- Can be trained on flight data to improve their decision-making.
- It learns from experience and does not need to be reprogrammed.

4.3.1 Architecture of Artificial Neural Network

The ANN is a simple model of an human's brain and a neuron is a type of cell in the brain that interprets electrical or chemical information. Tens of billions of interconnected cells make up the human brain, which is made up of neurons that are coupled to one another

to form networks [30]. A vast network of interconnected neurons in the biological brain enables it to carry out sophisticated intellectual functions.

Artificial Neural Networks are constructed using neuron models and typically comprise layers of neurons connected by weighted connections. The topology (or architecture) of a neural network is the configuration of its neurons, connections, and patterns. The neuronal frameworks seen in biology serve as inspiration for artificial neural networks. An unpredictable chemical process in which particular transmitter chemicals are produced from the transmitting side of the neural link, are used by biological neurons for signal transmission. The electrical potential inside the receiving cell is altered as a result. In the odd chance that this potential wins an advantage, the neuron fires [31]. With a few modifications, the neuron model seen in the figure below is frequently utilized in artificial neural networks.

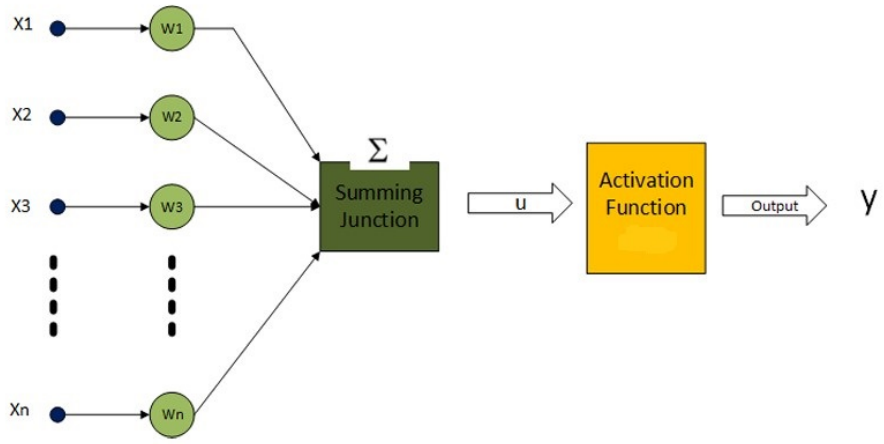


Figure 4.3: The Artificial Neural Network Architecture

Multiple layers of feature-detecting neurons make up the network. where a large number of neurons in each layer react to various input signal combinations from the layers above.Although artificial neural networks have many different kinds of layers, the input layer, hidden layer, and output layer are the three main layers [31].

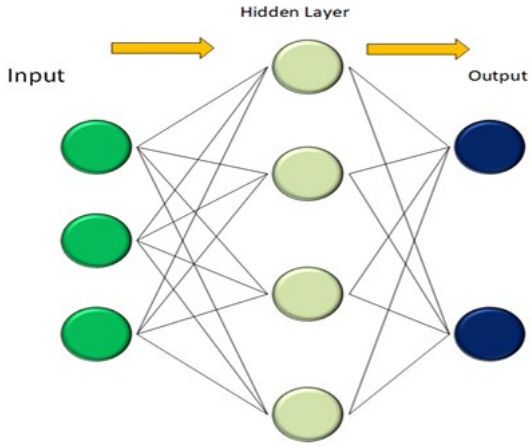


Figure 4.4: Artificial Neural Network

Input layer: The input layer provides the neural network with a pattern and interacts with the outside world. Every input neuron ought to stand for an independent variable that influences the output of the neural network. The neural network's input layer is its initial layer. Its main purpose is to take in input data, which can take many different forms. A feature or variable of the input data is represented by each neuron in the input layer.

Hidden layer: The real processing and transformation of the input data takes place in hidden layers. They make it possible for the network to pick up intricate representations and patterns. One or more hidden layers may be present in a neural network. Neurons that apply weights and biases to the inputs from the preceding layer make up each hidden layer. Based on the discrepancy between the expected and actual outputs during training, the network modifies the weights and biases of the neurons in hidden layers.

Output layer: An artificial neural network's output layer gathers and sends data in the manner intended by the network's architecture. The final predictions or classifications are produced by the output layer using the input data that has been processed by the hidden layers. The input layer can be directly linked to the pattern that the output layer displays. The particular job determines how many neurons are in the output layer.

4.4 Fuzzy Logic Control

A fuzzy logic controller is one way to express, alter, and implement a human's knowledge of system control. Lotfi Zadeh came up with the idea after seeing that, unlike computers,

humans make decisions by selecting between YES and NO. The fuzzy logic controller consists of four key parts: fuzzy logic, rules, defuzzification, and inference system.

Fuzzy logic control, or FLC, has shown effective for complex, nonlinear, and imprecisely described systems when conventional model-based control strategies are impractical or unachievable. Fuzzy Logic solves ambiguity and uncertainty problems by using membership functions with values between 0 and 1. If the system is too complex to extract the required decision rules, or if the provided data is untrustworthy, developing a fuzzy logic controller becomes extremely difficult. In this case, the expert knowledge can be used to create suitable rules that can be used to adjust the controller to achieve a better result.

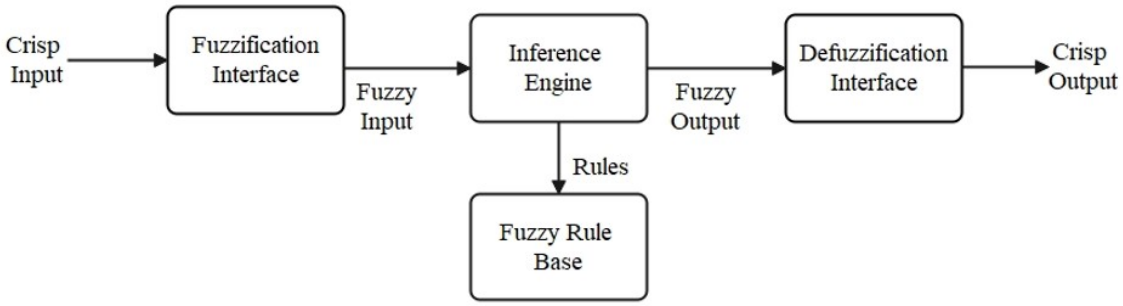


Figure 4.5: Fuzzy Controller Structure

Fuzzification: The initial stage of fuzzy logic control, known as "fuzzification," converts the system inputs—which are discrete numbers—into fuzzy set membership values in the fuzzy inputs. The linguistic variables of the fuzzy rules are represented by the fuzzy sets, which are defined based on the degree of the associated membership functions. The technique of figuring out how much a crisp input belongs to the fuzzy set is called "fuzzification." The membership functions may be triangular, trapezoidal, gaussian, or bell-shaped. Fuzzification, then, is the process of turning clear input facts into fuzzy values using appropriate linguistic values and predefined membership functions [32].

Fuzzy Rule Base: It includes a rule base with the control objectives and control policy established by the experts using language control rules, as well as a database with the definitions required to characterize linguistic variables and fuzzy data manipulation. The decision-making unit, which employs fuzzy logic inference rules to infer fuzzy control actions, is the brains or core of a fuzzy logic controller. A set of IF-THEN rules is used to infer a control action based on expert knowledge or design. Often referred to

as fuzzy conditional statements or fuzzy controlled rules, IF (antecedents are satisfied) THEN (consequences are inferred) rules are supplemented with linguistic variables [32].

Inference Engine: Based on the input field, it determines which rules should be applied and compares how closely the current fuzzy input meets each rule. The fired rules are then combined to generate the control actions. Every rule in the rule base uses the operators AND, OR, and NOT as premises words. These operators work between the membership degrees of the input values to produce a single result, which is referred to as the rule's certainty. The second type of operation is an implication, which is an operation between the fuzzy set of the associated rule's output and its certainty after implication. For each rule in the rule base, the implied output is represented by a fuzzy set.

Defuzzification: TAn inferred fuzzy control value from the decision-making unit is converted into a crisp or non-fuzzy control value by the defuzzification unit, the last part of the fuzzy logic controller, and then fed into the regulating action process. Defuzzification can be done in a variety of ways. The most often used method for determining the ideal control value is the Center of Gravity approach. However, because the integral calculation in the center of gravity approach is typically erratic, it is not computationally simple. Since a good approximation of the center of gravity defuzzification method is the center of average defuzzification method, which is computationally very simple, it is the most widely used defuzzification method in practical fuzzy control applications which expressed as:

$$\text{crisp } = \frac{\sum_{n=1}^{m} w_n u_n}{\sum_{n=1}^{m} w_n} \tag{4.33}$$

Where: $\mu_{\text{out}}(u)$, u, u_n, w_n and m are the resultant output fuzzy sets from the inference mechanism, crisp input, the center of the n^{th} implied output fuzzy set, height of n^{th} implied output fuzzy set and, the total number of rules in the rule base respectively.

4.4.1 Types of Fuzzy System

Mamdani fuzzy system: is a well-known framework for fuzzy inference that relates how humans handle ambiguity and imprecision in thinking. Fuzzy sets, which serve as the system's central component and represent input and output variables. A series of if-then rules drive the system, connecting fuzzy inputs to fuzzy outputs. The system assesses the rules based on fuzzy logic operators to determine how strongly each rule is activated after performing fuzzification to convert crisp input values into fuzzy words.

Following rule evaluation, the Mamdani system combines the fuzzy outputs from every rule that has been triggered into a single fuzzy set that symbolizes the system's overall reaction. The last stage, called defuzzification, uses techniques like the centroid approach to transform this combined fuzzy output into a crisp number. Because of this procedure, the system is guaranteed to produce useful results, which makes it especially useful for control applications.

$$Crisp = \frac{\int_{\mu} \mu(x) x dx}{\int_{\mu} \mu(x) dx} \tag{4.34}$$

Takagi-Sugeno fuzzy system: is a kind of fuzzy inference system that uses an alternative method for rule formulation to improve the capabilities of conventional fuzzy systems. An antecedent (the "if" component) plus a consequent (the "then") make up a fuzzy rule in a TS fuzzy system. The consequent is usually written as a crisp function of the input variables. Because the rules may now be stated as constant or linear functions rather than fuzzy sets, the system can generate outputs that are more accurate.

Takagi-Sugeno systems are especially useful in applications like system identification, control, and optimization that call for increased precision and adaptability because of their structure. Fuzzifying the inputs, assessing the degree to which each rule is fulfilled, and then aggregating the outputs by taking a weighted average of the subsequent functions are the steps involved in the inference process.Because of this, TS fuzzy systems produce a clear output that accurately captures the aggregate effect of all active rules. This makes them appropriate for complicated, nonlinear systems that require accurate modeling of the precise relationships between inputs and outputs. Calculation of the crisp output as a weighted average of the total clipped singletons [33].

$$Crisp = \frac{\sum_{i=1}^{n} \mu(z_i) \cdot z_i}{\sum_{i=1}^{n} \mu(z_i)} \tag{4.35}$$

4.5 Adaptive Neuro Fuzzy Inference System

Before fuzzy systems were widespread in industrial applications, it was challenging to create a fuzzy system with acceptable performance. Finding the ideal rules and membership features frequently requires a great deal of trial and error. Neural networks, which have an efficient learning capacity, were one way to automate the production of fuzzy system tuning. To combine their advantages and treat their separate ailments, neural networks and fuzzy logic systems can be combined [17]. Fuzzy systems provide

computational learning properties to neural networks, which in turn understand and provide clear system representations. Here, the learning capabilities of neural networks compensate for the shortcomings of fuzzy systems. The combination of neural network and fuzzy logic controller is considered the most efficient approach for different functions' approximation and indicates their ability to control nonlinear dynamical systems. We use Adaptive Neuro Fuzzy Inference System (ANFIS) architectures for this thesis work.

One kind of fuzzy inference system (FIS) that creates the mapping of inputs to output is the adaptive neural fuzzy inference system (ANFIS). Fuzzy logic (FL) and artificial neural networks (ANN) are both used in the input-to-output mapping process. FL is renowned for its structured knowledge representation and adept handling of ambiguity. Obtaining membership functions, allocating membership functions, and establishing fuzzy rules are the most challenging aspects of FIS. The trial-and-error method is used to determine these parameters, and neural networks are used to modify them. In ANFIS, the ANN component aids in parameter optimization and error reduction. Since ANFIS combines the benefits of FL and ANN, it has emerged as a significant research step in domains where FIS has been effectively applied, including computer vision, data categorization, decision analysis, expert systems, and automatic control [34].

Adaptive network is a network structure made up of nodes and directed links. The values of a set of adjustable parameters that connect the nodes dictate the overall input-output behavior of the network [35]. The adaptive system determines the parameters of Sugeno-type fuzzy inference systems using a hybrid learning technique. In order to simulate a given training data set, it uses a mix of the back-propagation gradient descent approach and the least-squares method for training FIS membership function parameters [36].The learning or training phase of the neural network involves determining parameter values that suit the training data. To reduce training error in ANFIS, alternative algorithms can be employed. The gradient descent algorithm and the least squares method are coupled for an effective search for the optimal parameters. The main benefit of such a hybrid technique is that it converges significantly faster because it reduces the search space dimensions of the backpropagation method used in neural networks [37]. The fuzzy Sugeno model, known as ANFIS, is incorporated into the adaptive system framework and is used for model construction and validation in order to support training and adaptation [38].

4.5.1 ANFIS Architecture

Adaptive networks are multilayer feed-forward networks with nodes connected by directed links. To generate a single node output, each node in the network applies a particular function to the signals it receives. There are no weights associated with the links in an adaptive network; instead, each connection chooses the direction in which signals flow from one node to another. More specifically, the setup of an adaptive network generates a single node output by applying a static node function to its incoming signals. Every node function is a parameterized function with movable parameters; changing these parameters changes the behavior of the adaptive network as a whole as well as the node functions.

The fuzzification layer, product layer, normalized layer, defuzzified layer, and total output layer make up the whole system design. The ANFIS method uses input/output data for a given set of parameters to model the fuzzy inference system (FIS) whose membership function parameters are tuned (adjusted) using either a back-propagation algorithm alone or in combination with a least squares kind of approach. The main objective of the ANFIS is to determine the optimal values for the corresponding fuzzy inference system parameters through a learning approach.

To reduce the erroe between the anticipated and actual output, the parameters are changed throughout the training phase. A hybrid approach for optimization that blends the gradient descent methodology with the least squares approximation. The form of the membership functions is dictated by the premise parameters in ANFIS, which are the ones that require optimization.Citation: Patel (2014) Flood. Any of several optimization techniques can be applied to reduce the error measure after MFs have been created. The parameter design of the adaptive network allows fuzzy systems to learn from the data they are representing. It is assumed that the adaptive neural fuzzy inference system being studied in this work has one output as training data and one error input.

Let's examine a Takagi, Sugeno, and Kang (TSK) fuzzy inference system of the first order with two rules [40]:

Rule 1: **IF** (x is A_1) **AND** (y is B_1) **THEN** $f_1 = p_1x + q_1y + r_1$

Rule 2: **IF** (x is A_2) **AND** (d is B_2) **THEN** $f_2 = p_2x + q_2y + r_2$

Where: x and y are the inputs, A_i and B_i are the fuzzy sets, f_i are the outputs within the fuzzy region specified by the fuzzy rule, and p_i, q_i, r_i are the design parameters that are determined during the training process.

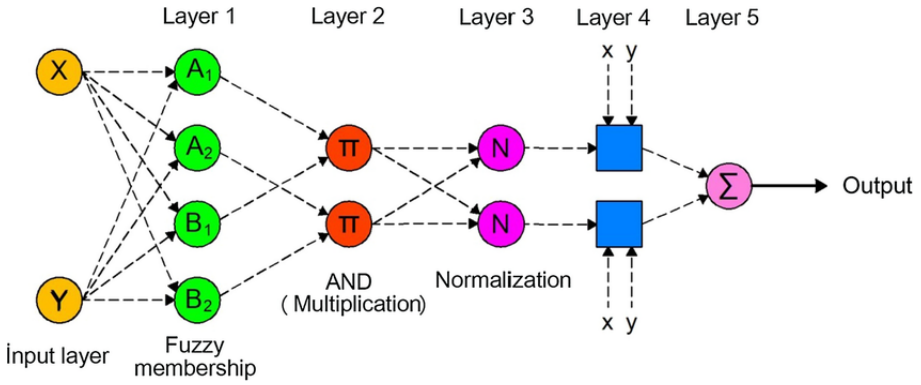


Figure 4.6: Basic architecture of ANFIS

Since fuzzy logic and neural networks are integrated to generate ANFIS, it is crucial to precisely adjust the number of training epochs, membership functions, and fuzzy rules while creating the ANFIS model. Because it could cause the system to overfit the data or fail to fit the data at all, mapping those parameters is extremely important. A hybrid approach that combines the gradient descent and least-squares methods with a mean square error method can be used to achieve this adjustment. We tend to lower the training error in the training process since a better (more accurate) ANFIS system is indicated by a smaller difference between the ANFIS output and the desired aim [41].

4.5.2 Layers of ANFIS

Below is an overview of the ANFIS algorithm's five layers. Fuzzy Sugeno-type system components make up the five-layer architecture of the suggested neuro-fuzzy network [15][42]:

Layer 1 (Fuzzification): his layer's input nodes are all adaptive nodes, and outputs are the inputs' fuzzy membership grades, which are determined by the following formulas:

$$O_{1,i} = \mu_{Ai}(x), i = 1,2 \tag{4.36}$$

$$O_{1,j} = \mu_{Bj}(y), j = 1,2 \tag{4.37}$$

where x and y are the inputs to node i, and A_i and B_j are the linguistic labels associated with this node function. $\mu_{Ai}(x)$ and $\mu_{Bj}(y)$ can adopt any fuzzy membership function. For example, if the bell-shaped membership function is employed, $\mu_{Ai}(x)$ is given by:

$$\mu_{Ai}(x) = \frac{1}{1 + \left[\left(\frac{x-c_i}{a_i}\right)^2\right] b_i}, i = 1,2, \tag{4.38}$$

For the Gaussian membership function by:

$$\mu_{Ai}(x) = \exp\left[-\left(\frac{x-c_i}{a_i}\right)^2\right] \tag{4.39}$$

where: a_i, b_i, and c_i are the parameters of the membership function.

Layer 2 (weighting of fuzzy rules): The nodes in the layer are fixed nodes. This layer fuzzifies the inputs using fuzzy operators and the AND operator. The firing strength w_i in this layer is calculated using the membership values found in the fuzzification layer. The results are known as the rules' firing strengths and are calculated as follows:

$$O_{2,i} = \mu_{Ai}(x) * \mu_{Bj}(y), i, j = 1,2 \tag{4.40}$$

Layer 3 (normalization): Every node in this layer is fixed, and in order to show that they have a normalizing effect on the firing strengths from the previous layer, each node is normalized by calculating the ratio of the rule's firing strength (true values) to the firing strength of all rules. This layer's output can be shown as follows:

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2}, i = 1,2. \tag{4.41}$$

Layer 4 (de-fuzzification): The output of each adaptive node in this layer is just the product of a first order polynomial (for a first order Sugeno model) and the normalized firing strength. The weighted consequent values of the rules are determined by each node in this layer as follows:

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i), i = 1,2, \tag{4.42}$$

where: w is the output of Layer (3), and p_i, q_i, and r_i are the consequent parameters.

Layer 5 (Summation): In order to get the overall ANFIS output, the output of this layer—which has a single fixed node labeled with $\sum$—is calculated by adding the outputs of all incoming signals from the defuzzification layer. The model's total output is provided by:

$$O_{5,i} = \sum_i \bar{w} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}. \tag{4.43}$$

4.5.3 Hybrid Learning Algorithm

The popular ANFIS learning algorithm is a hybrid algorithm that blends gradient descent with least squares methods. In the forward pass of the hybrid learning process, node outputs continue until the outputs of all incoming signals are added up, and the resulting parameters are then determined by the least squares method. In the backward pass, the error signals propagate backward while the premise parameters are modified using gradient descendant. The hybrid learning strategy converges significantly faster than the original back propagation method by reducing the search space dimensions [43]. The overall output can be given by:

$$f = \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2, \tag{4.44}$$

$$f = \bar{w}(p_1 x + q_1 y + r_1) + \bar{w}(p_2 x + q_2 y + r_2), \tag{4.45}$$

$$f = (\bar{w}_1 x) p_1 + (\bar{w}_1 y) q_1 + (\bar{w}_1) r_1 + (\bar{w}_2 x) p_2 + (\bar{w}_2 y) q_2 + (\bar{w}_2) r_2, \tag{4.46}$$

where: p_1, q_1, r_1, p_2, q_2, and r_2 are the linear consequent parameters.

The resultant parameters are optimized and their ideal values are determined using the least squares approach (forward pass). The search space grows and the training's convergence slows down when the premise parameters are left unfixed. In each iteration, the learning algorithm back-propagation is used to adjust the parameters of the premises while keeping the consequent constant. The gradient descent method (backward pass) is used to optimize the premise parameters, and the input patterns are propagated once again.To accomplish a desired input-output mapping, each node in layers one and four has an adaptive node with updated parameters.

The premise parameters (layer one) and consequent parameters (layer four) in the ANFIS learning process should be adjusted to provide the best possible mathematical relationship between inputs and outputs. ANFIS employs a two-pass learning cycle. In the forward pass, the algorithm determines the consequent parameters using the least squares approach, and in the backward pass, the premise parameters are updated using gradient descent and the mistakes are propagated backward.

4.6 Adaptive Neuro-Fuzzy based Sliding Mode Control

In order to provide quadcopters with reliable and intelligent control despite nonlinear dynamics and uncertainties in aerial dynamics, the Adaptive Neuro-Fuzzy infrence system adjusting Sliding Mode Controller (ANFIS-SMC) was designed. The dynamics of a quadcopter are described by parameters such as orientation, altitude, and position that need tight control to maintain stability and responsiveness.

Sliding mode control appears in a dynamic system restricted by an ordinary differential equation with discontinuous sides. This sudden discontinuity happens if the control of the system state switches at a high frequency as sliding mode control disadvantage, that is the chattering phenomenon. This phenomenon can damage the system; chattering originates from the interaction between uncontrolled dynamics and high frequency switching control [44]. In order to overcome these problems, a Adaptive Neuro-Fuzzy controller dependent on a sliding mode controller is designed in this thesis work.

Adaptive Neuro-Fuzzy Inference System (ANFIS) presents an innovative approach that gives the strengths of fuzzy logic and neural networks to enhance quadcopter performance. ANFIS offers a more fluid control signal by applying fuzzy rules that adapt to different situations. This adaptability is especially useful in dynamic environments where the quadcopter might face changes in payload, wind disturbances, or battery performance. By continuously learning from operational data, ANFIS can modify its control strategies in real-time, ensuring stable flight even in unpredictable conditions.

Various researchers employ various techniques to minimize chattering. These techniques, which include substituting the signum function with a saturation function and a fuzzy logic controller, are documented in the literature; however, it is challenging to determine the ideal value to satisfy robustness and high frequency chattering in the quadrotor control design. Because of the Neuro-Fuzzy system's strong capacity for self-learning, we employ it to lessen high frequency chattering, torque ripple, and system robustness.

The most effective method for modeling various functions is the combination of a neural network and a fuzzy logic controller, which shows that they can manage nonlinear dynamical systems. The designed ANFIS-SMC combines sliding mode control methodology with neural network and Fuzzy logic controller. In this thesis work, the quadcopter virtual controller signals and the control effort signals of the designed sliding mode con-

troller is tuned by Neuro-Fuzzy controller (ANFIS) to minimize chattering and torque ripple.

A fuzzy inference system is created using SMC data, defining membership functions and fuzzy rules to illustrate the connection between inputs and the desired outputs. The training process employs a neural network approach to enhance these rules and fine-tune the membership functions, leading to a model that accurately captures the dynamics of the quadcopter. After designing the ANFIS controller, it is incorporated into the quadcopter's control loop. Simulation studies are performed to assess the ANFIS controller's performance compared to the conventional SMC approach. Important performance metrics, including stability, response time, and robustness to disturbances, are examined. Initial findings typically indicate that ANFIS-SMC not only minimizes chattering but also enhances the quadcopter's overall responsiveness, facilitating smoother transitions and more accurate maneuvers. This is especially crucial for applications that demand stable hovering and agile movement, as reliable control can greatly improve user experience and operational efficiency.

4.6.1 Training of ANFIS controller using SMC data

An organized method that combines control theory and machine learning is required to train an Adaptive Neuro-Fuzzy Inference System (ANFIS) controller for a quadcopter using data from a sliding mode controller. A fundamental idea in sliding mode control, which is especially useful for controlling the dynamics and uncertainties involved with quadcopter flight, is the sliding surface.

To start the ANFIS learning; first, from figure 4.1 the switch will select the appropriate control signal to be applied to the quadrotor dynamics. SMC training data set that contains the desired input / output data pairs of target systems to be modeled is designed. The training process starts with gathering input-output data related to the error input and controlled output from the designed SMC as shown figure 4.7 . The design parameters required for any ANFIS controller are Number of data pairs, Training data set and Fuzzy inference systems for training, Number of epochs to be chosen to start the training, learning results to be verified after mentioning the step size.

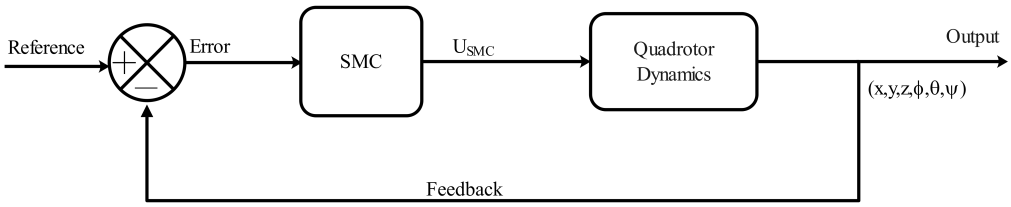


Figure 4.7: Sliding Mode Controller block diagram

After collecting the data, using Neuro-Fuzzy Designer (ANFIS Toolbox) will define the fuzzy rules and membership functions that will serve as the foundation of the ANFIS model. These rules are often based on expert knowledge or can be derived from the training data using clustering techniques. The membership functions must be thoughtfully designed to reflect the system's dynamics and address the uncertainties found in real-world situations. This step is essential because the success of the ANFIS controller heavily relies on how accurately these fuzzy rules and membership functions depend on the underlying system behavior.

Finally optimization algorithms and hybrid learning techniques are used to train the ANFIS controller with the goal of minimizing the discrepancy between expected and actual outputs. The settings of the fuzzy rules and membership functions are changed during this training phase to enhance the system's tracking capabilities. Once the training is complete, the ANFIS controller can adapt to changes in system dynamics and external disturbances, by tuning the SMC and making it a reliable solution for a range of control applications. By learning from data while integrating fuzzy logic principles, ANFIS can deliver effective control even in complex and uncertain environments.

A conventional SMC controller is developed and simulated using MATLAB/2024a to gather training data for 30 seconds in order to create an ANFIS system. Following input/output data collection, the data was converted into training data sets. The total data were designated as training data, because ANFIS combines the capabilities of fuzzy logic and neural networks, it offers a way for the fuzzy modeling process to learn about a data set and calculate the membership function parameters that enable the related fuzzy inference system (FIS) to monitor the provided input/output data.

These parameters are determined in this thesis work through the use of a hybrid learning approach and the virtual controller signals U_x, U_y, and U_z, as well as the control effort signals U_2, U_3, and U_4, are trained ANFIS form the SMC control signals.

ANFIS was trained using 10 membership functions and 100 epochs, using Gaussian functions ($gaussmf$), ($gauss2mf$), and sigmoidal ($psigmf$) as input and output membership functions. Ten rules are generated from the selected number of membership functions using the linear and constant membership function type as the output. Appendix B displays the plot of membership functions for inputs, surface, and rule views. To evaluate how successfully the model is learning to operate the quadcopter, the training data sets are chosen based on RMSE result. In order to obtain the ideal ANFIS system, we trained the system with different settings for items such as data set sample, type and number of membership functions, and number of epochs to achieve the best performance. Using the obtained input/output data set from SMC, the MATLAB toolbox ($anfisedit$) editor constructs a fuzzy inference system with Fuzzy logic Tool box as shown on the flowchart 4.8

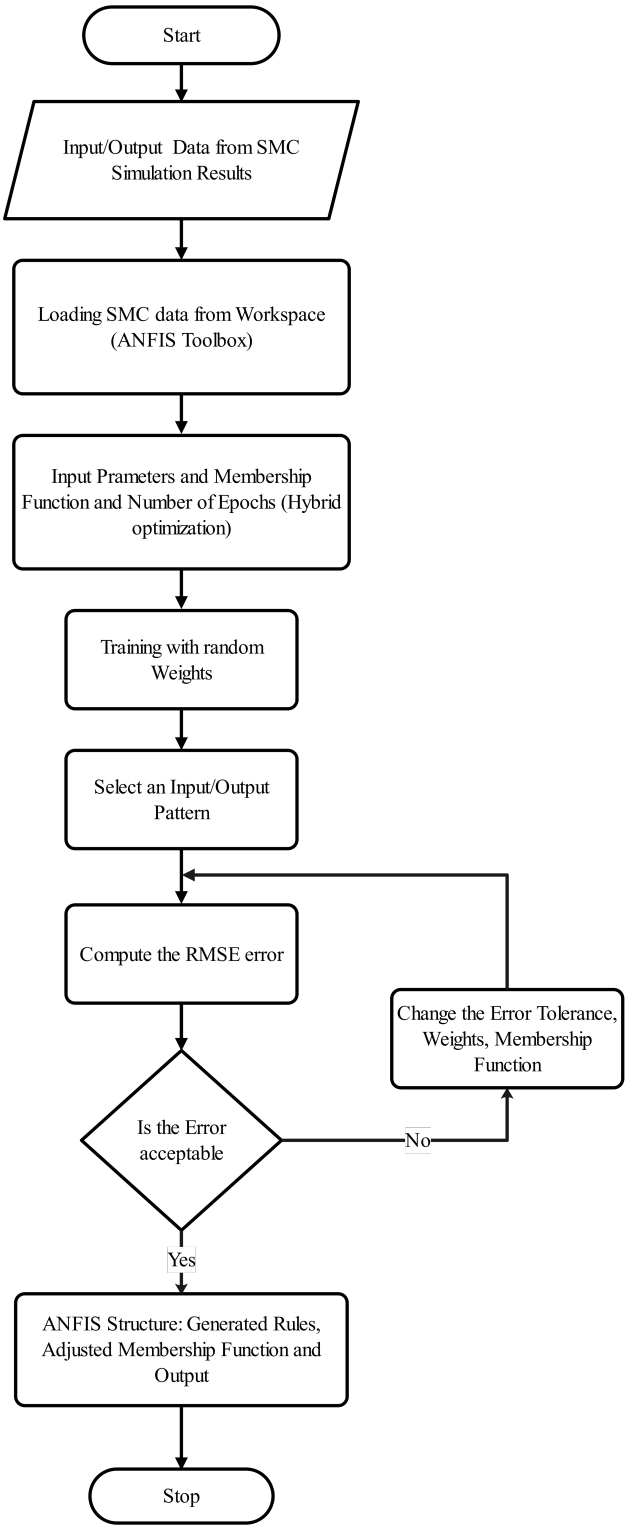


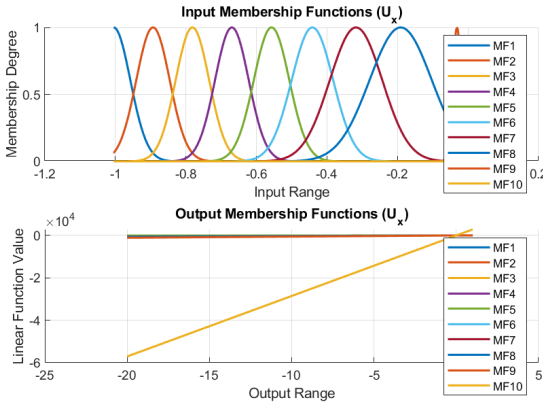
Figure 4.8: ANFIS Training System

The summary of the tested membership function numbers and types with their respective epoch numbers is tabulated in Appendix A.1. Trials and error values are used to calculate the necessary number of membership functions. It suggests that the kind and quantity of membership functions have a significant impact on each ANFIS model. The RMSE for the training data at each epoch is recorded by checking the RMSE error as the system is trained, the checking error versus epochs curve is plotted using the ANFIS Editor GUI. The minimal RMSE for training is 0.0014424.

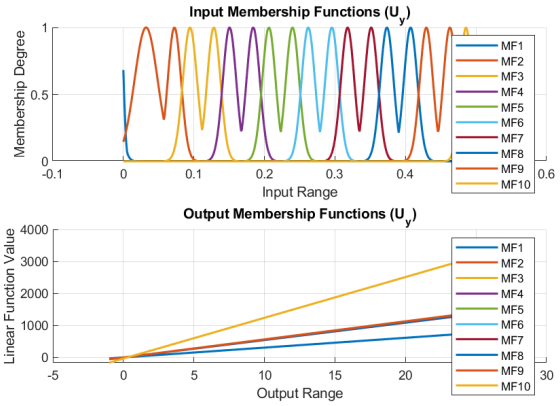
The plots in figure 4.9 (a) illustrating membership functions associated with ANFIS which has Gaussian-shaped membership functions (MF1 to MF10), each representing a degree of membership for a distinct region of the input data, are shown in the figure, "Input Membership Functions (Ux)," which is defined over specified input range. The overlapping structure is critical to ensure that the input values activate more than one membership function, facilitating smooth interpolation in the fuzzy system. And the "Output Membership Functions (Ux)," the figure at the bottom, shows the matching output membership functions. Each function are linear and with the membership functions (MF1 to MF10) to their corresponding colors is included in both plots. Here, the MFs appear to represent linear functions, with some functions (e.g., MF3) showing a steep slope while others remain flat, likely corresponding to constant output values. Overall the figure show the system's fuzzification and defuzzification of the input and output.The smooth transitions in the Gaussian input MFs and the linear output mapping demonstrate a carefully designed system capable of handling complex, nonlinear input-output relationships.

Figure 4.9 (b) showing membership functions for the Input Membership Functions (Uy) with two Gaussian membership functions (MF1 to MF10) and with input range as displayed in the top plot. Because these MFs overlap, it is possible to activate many membership functions at once, which facilitates robust input coverage and smooth interpolation. These input MFs' range and spacing show that the input variable covers a comparatively small range, indicating fine-tuned system with accurate fuzzy input representation. The associated output membership functions are shown in the bottom plot, "Output Membership Functions (Uy)," which is linear and the input fuzzification and the output subsequent defuzzification are shown in the figures. In conclusion, the figure shows a well-structured fuzzy inference system that can effectively represent both small and major changes in output values since input MFs allow for fine-tuned input responses while output MFs offer a variety of linear responses.

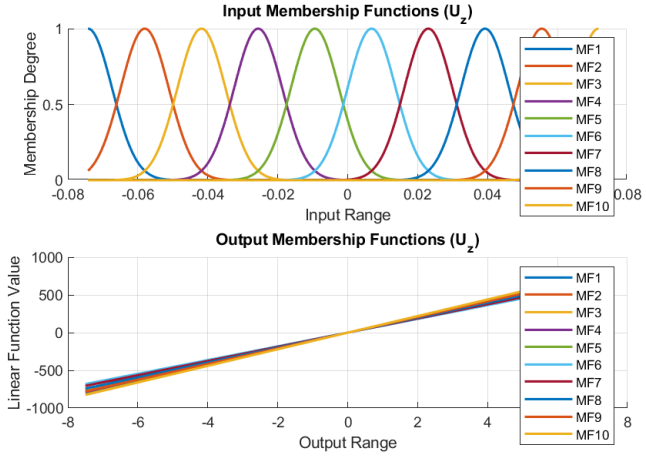
Figure 4.9 (c) illustrating membership functions for a Neuro fuzzy designer with Gaussian-shaped membership functions (MF1 to MF10), distributed symmetrically over a given input range which are displayed in the top plot, "Input Membership Functions (Uz)," signifying varying degrees of membership for discrete intervals in the input space. The matching output membership functions, which are linear and cover an output range as shown in the bottom plot, "Output Membership Functions (Uz)," which illustrates how fuzzified inputs are mapped to their outputs. Every function has a distinct color, and a legend links these colors to the corresponding membership functions (MF1 through MF10). The figure shows a well-tuned fuzzy system for U_z, which is appropriate for accurate small-range control applications because linear output functions maintain proportionality and narrow input membership functions guarantee high sensitivity.



(a) Ux



(b) Uy



(c) Uz

Figure 4.9: Membership function of Virtual controllers

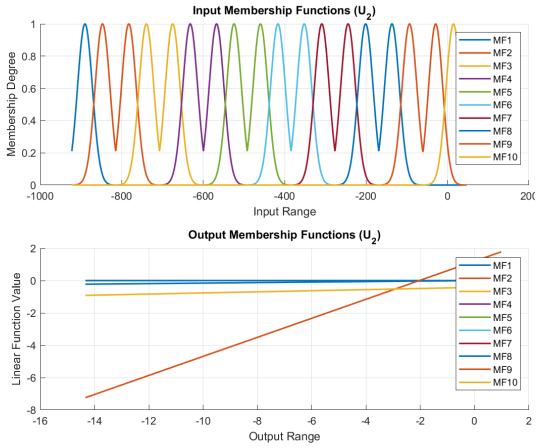
Two graphs on figure 4.10 (a) shows the neuro fuzzy logic membership functions are

used different membership functions (MF1 to MF10) are displayed in the top graph, which depicts the input membership functions U_2. This variation illustrates varying degrees of membership to different input values, facilitating a more advanced interpretation of data points. The overlapping characteristics of these functions imply that several membership functions may be applicable to a single input value, which is a core aspect of fuzzy logic that supports a more adaptable and human-like reasoning approach. The output membership functions for U_2 are shown in the bottom graph, which exhibits a linear function in the output range and The lines shows that these outputs remain constant across a wide range of inputs and the graph illustrates the membership functions' output values, some of which stay constant while others rise linearly, indicating varying degrees of effect on the output depending on the input membership degrees.

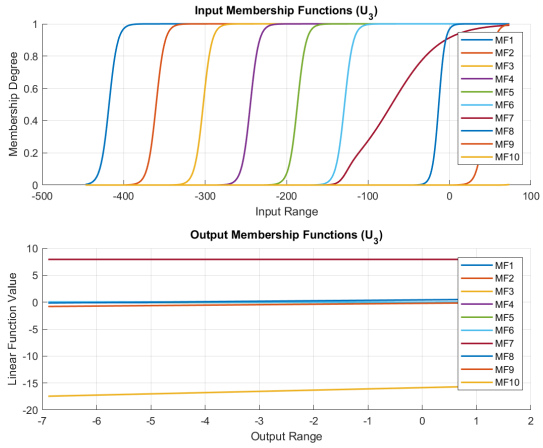
Two graphs from figure 4.10 (b) shows the ANFIS membership functions with different functions (MF1 to MF10) spanning an input range, the top graph displays the input membership functions U_3. These functions' sigmoidal forms, which show different changes in membership degrees as the input varies. This graph illustrates the inherent uncertainty present in the data, facilitating a more advanced decision-making process, as various inputs may simultaneously fit into multiple categories. The output membership functions for U_3 are depicted in the bottom graph, which shows a flat line spanning the output range, with all membership functions (MF1 to MF10) staying constant at zero. This implies that, within this range, changes in the input membership degrees do not substantially alter the output, suggesting a potentially non-influential or stable output response. The analysis of results emphasizes the generation of clear and consistent outputs derived from different inputs. In summary, the integration of precisely defined input membership functions with stable output functions underscores the efficacy of fuzzy logic in addressing uncertainty while guaranteeing dependable results and better decision-making.

Two plots figure 4.10 (c) showing Neuro fuzzy logic membership functions for U_4. The input membership functions are depicted in the top graph, which has different functions (MF1 to MF10) over an input range. These functions are distinguished by oscillatory patterns that show different levels of membership throughout this range. Each membership function intersects with others at specific points, illustrating the principle of fuzzy logic that allows an input to be associated with multiple membership functions. The output membership functions forU_4, on the other hand, are displayed in the bottom graph. They show a linear decline over the output range, with values falling off

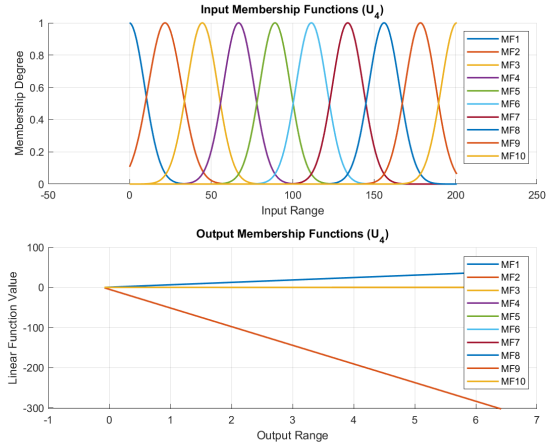
significantly, especially for MF1 and MF2. This pattern implies that the influence of the membership functions decreases with increasing output, suggesting a strong correlation between input and output behaviors. The integration of complex input functions with and output functions illustrates the membership data relationships and generating clear, actionable results that makes the designed fuzzy logic framework is well-suited for both adaptability and dependability in decision-making.



(a) U_2



(b) U_3



(c) U_4

Figure 4.10: Membership function of control effort

Chapter 5

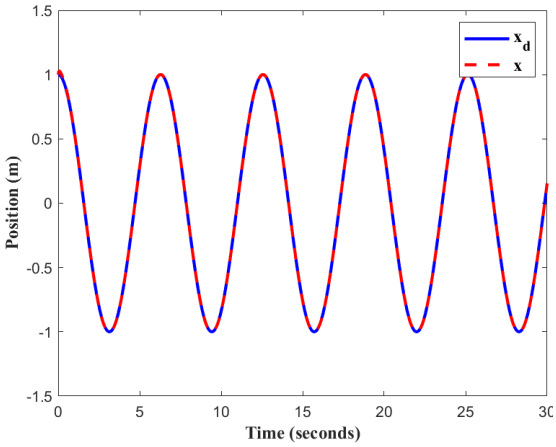
Simulation Result and Discussion

In this chapter, the dynamics of a quadrotor were simulated to evaluate the performance of the proposed controller in trajectory tracking. The simulation of the system is done using MATLAB Simulink software to test and verify the performance of the designed ANFIS controller and SMC contoller. Also design the simulation for Adaptive Neuro-Fuzzy inference system controller based sliding mode controller. The simulations were all done using sampling time of 30 seconds and the auto(ode45) solver of the fixed step.

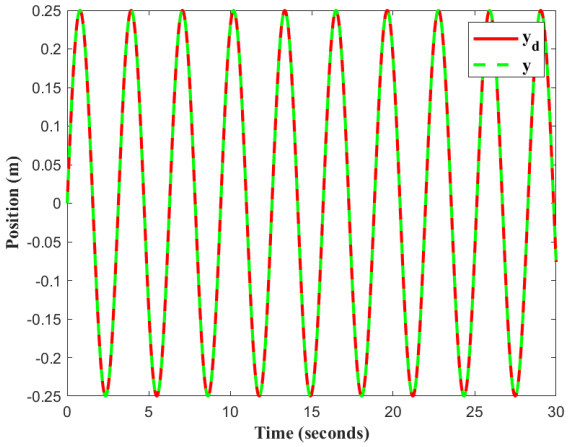
This section assessed how well the suggested controller performed in trajectory tracking by simulating using the proposed controller of a quadrotor. The conventional SMC controller and the ANFIS architecture were contrasted. Two case situations were used for the simulations. MATLAB and Simulink software (version R2024a) were used to calculate all simulation results. Lastly, Simulation results are presented and discussed to indicate the effectiveness of the proposed quadcopter based on position and orientation controlling of the flight system at different trajectories. Finally, to verify the validity and efficiency of the proposed controller, Appling different trajectories with disturbance like helical and infinity trajectories.

5.1 Infinity Trajectory Tracking

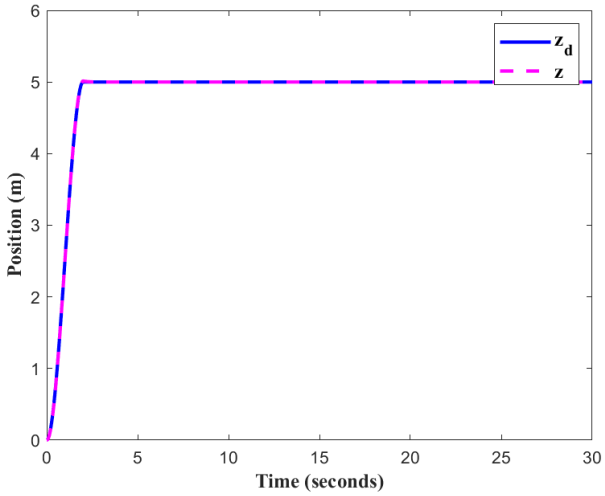
Infinity trajectory was utilized as the tracking reference, with sinusoidal input trajectories applied to both the X and Y axes, and Z axes. Positions along x,y and z-axis are presented in 5.1 and 5.2 displays the reference signal that was produced by the outer loop together with its trajectory output, which includes the roll (ϕ), pitch (θ), and yaw (ψ) angles. Figure 5.3 displays the resultant 3D trajectory tracking response. It is evident from the responses that the suggested ANFIS-SMC can ensure precise reference tracking and quick convergence while preserving stability.



(a) Infinity trajectory position along x-axis.

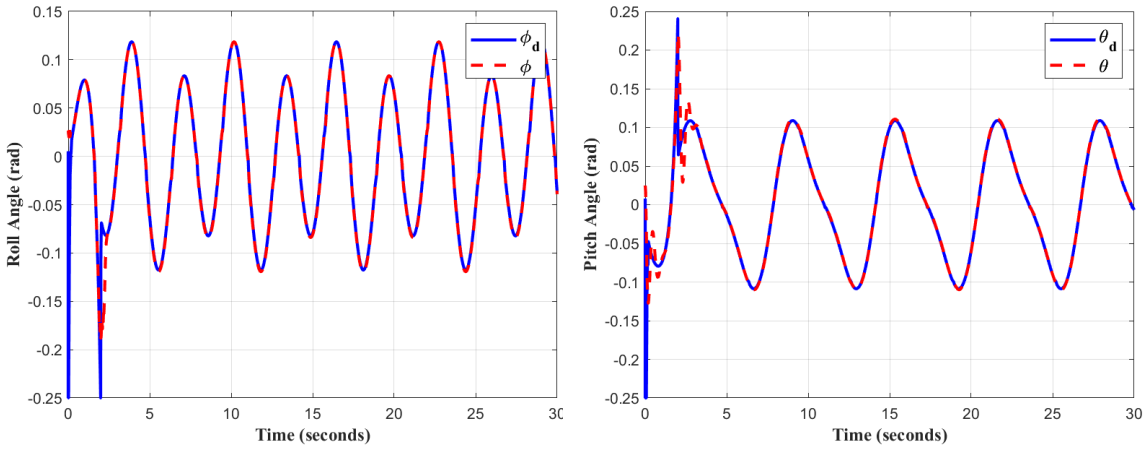


(b) Infinity trajectory position along y-axis.

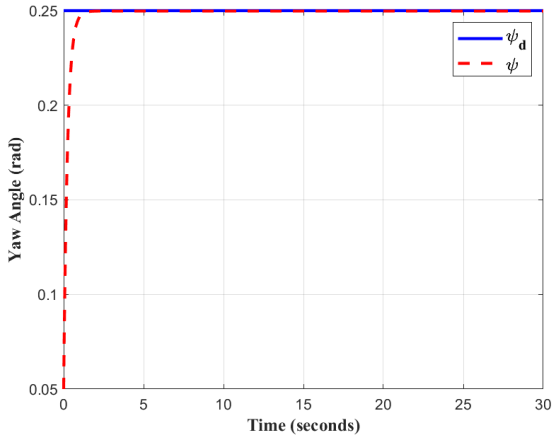


(c) Infinity trajectory position along z-axis.

Figure 5.1: Positions in Infinity Trajectory Tracking



(a) **(b)**



(c)

Figure 5.2: Attitudes in Infinity Trajectory Tracking

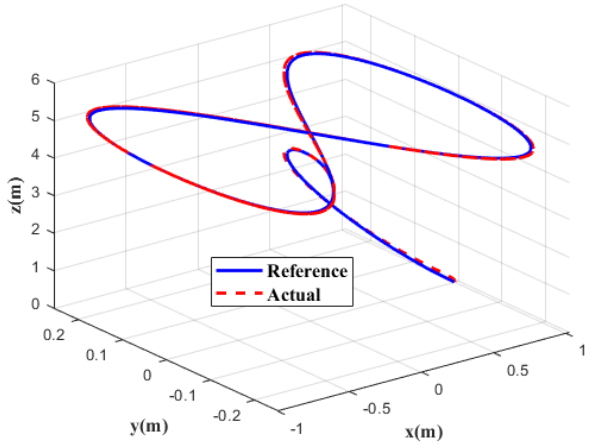


Figure 5.3: 3D plot of Infinity trajectory tracking

Figure5.4 displays the outer loop or virtual controller signals Ux, Uy, and Uz, which are utilized to compute the intended roll (ϕ_d), pitch (θ_d), and attitude controller U1. In order to calculate the force needed to maintain the intended trajectory, the controller signal's amplitude varies in reaction to the input trajectory.

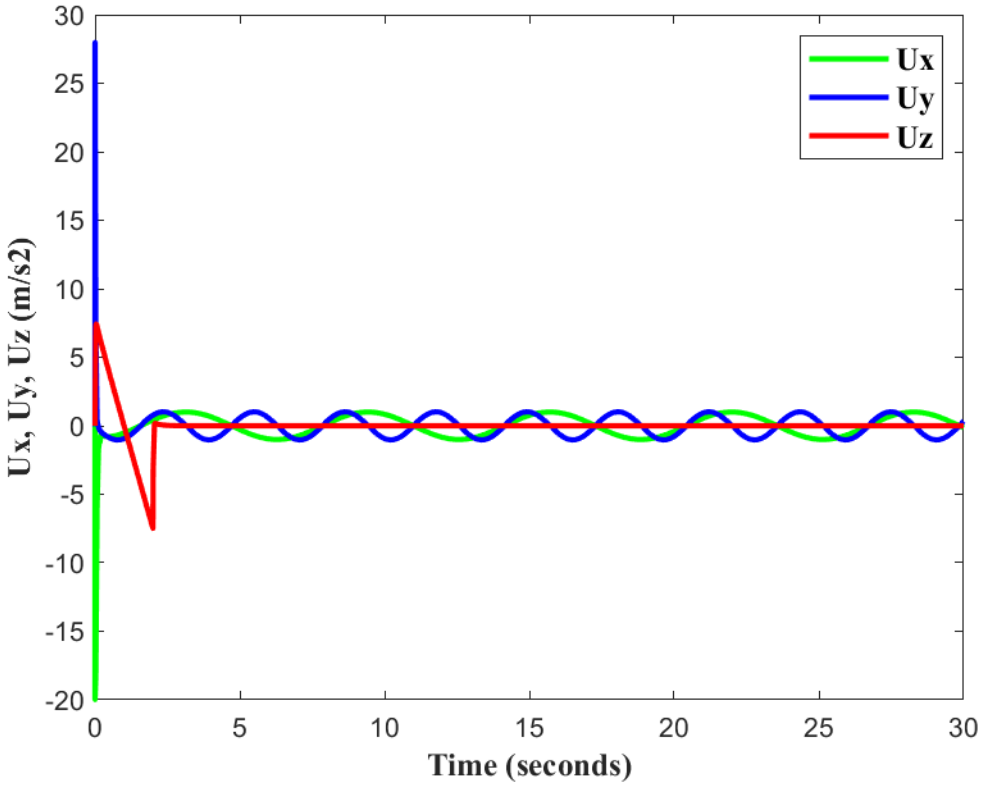
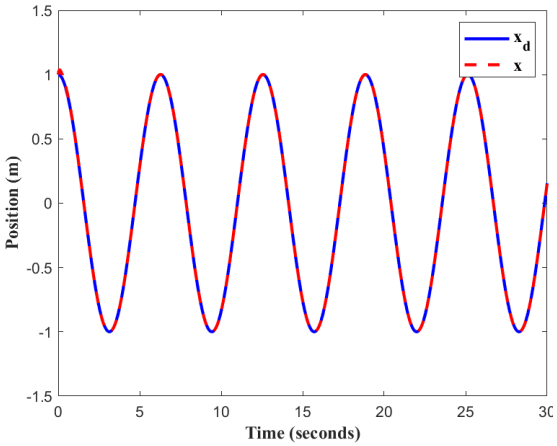


Figure 5.4: Infinity Virtual controller response.

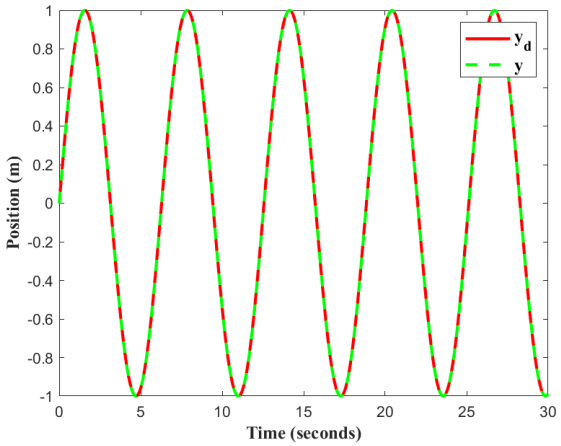
5.2 Helical Trajectory Tracking

A quadcopter's helical trajectory is a three-dimensional path that simultaneously ascends or descends vertically and moves in a circle around a central axis. Figures 5.5 illustrate the helical trajectory of the x, y, and z motion. Additionally, the roll (ϕ), pitch (θ), and yaw (ψ) angles followed the desired trajectory that was generated from the x, y, and z motion, allowing them to reach their desired trajectories. Figures 5.6 also show the roll and angle trajectory, and they had no issue maintaining the desired trajectory. Figure 5.7 show the helix 3D trajectory tracking demonstrates the controller's ability to follow the trajectory satisfactorily.

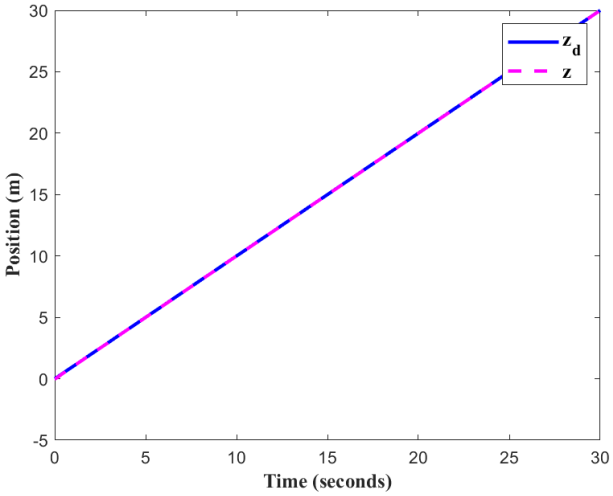
Figure 5.8 displays the helical virtual controller signals Ux, Uy, and Uz, which are utilized to compute the desired roll (ϕ_d), pitch (θ_d), and attitude controller U1. In order to calculate the force needed to maintain the intended trajectory, the controller signal's amplitude varies in reaction to the input trajectory.



(a) Helical trajectory position along x-axis.

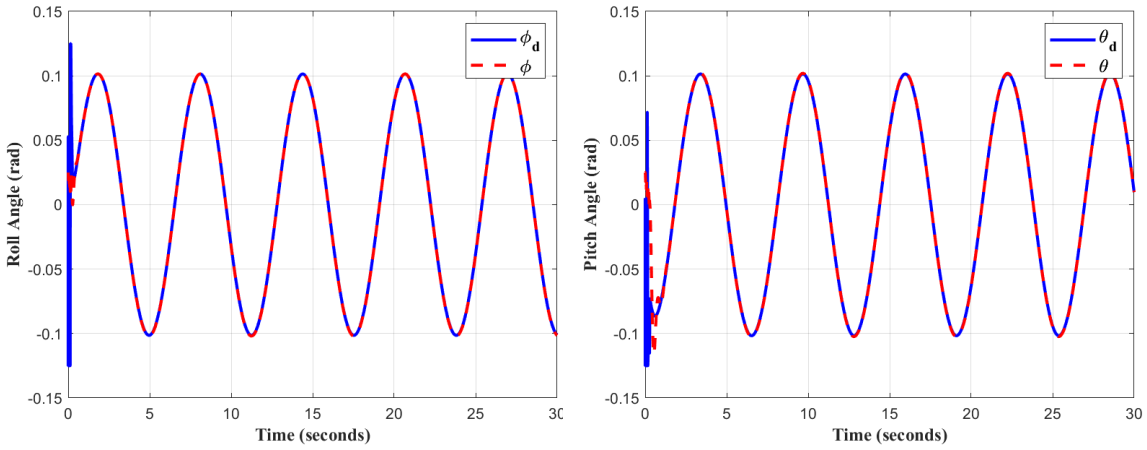


(b) Helical trajectory position along y-axis.

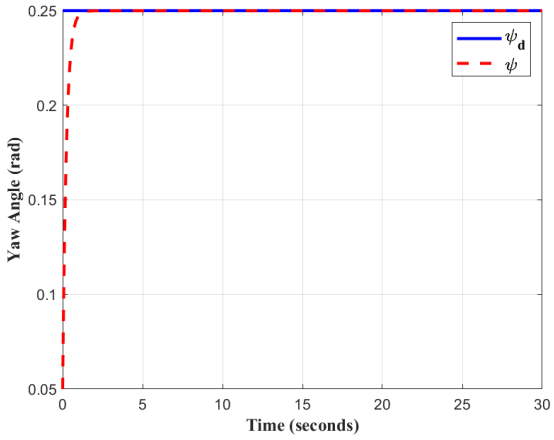


(c) Helical trajectory position along z-axis.

Figure 5.5: Positions in Helical Trajectory Tracking



(a) **(b)**



(c)

Figure 5.6: Attitudes in Helical Trajectory Tracking

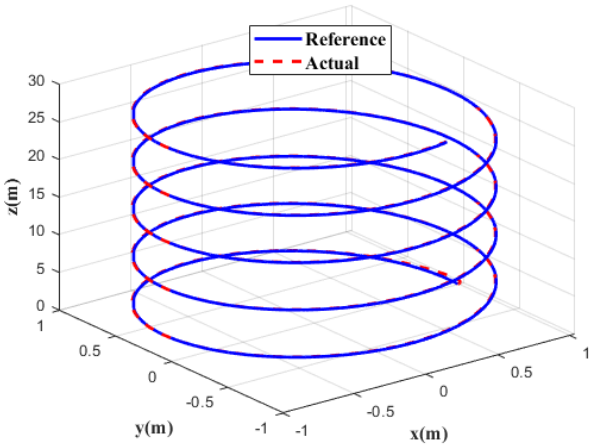


Figure 5.7: 3D plot of Helical trajectory tracking

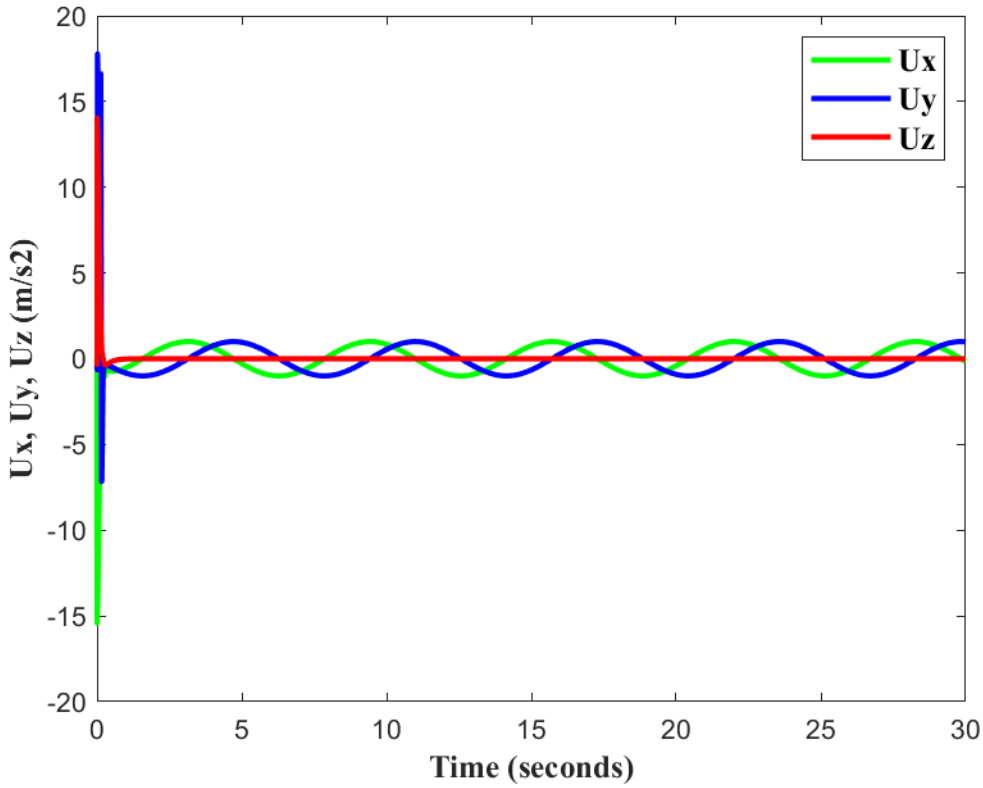
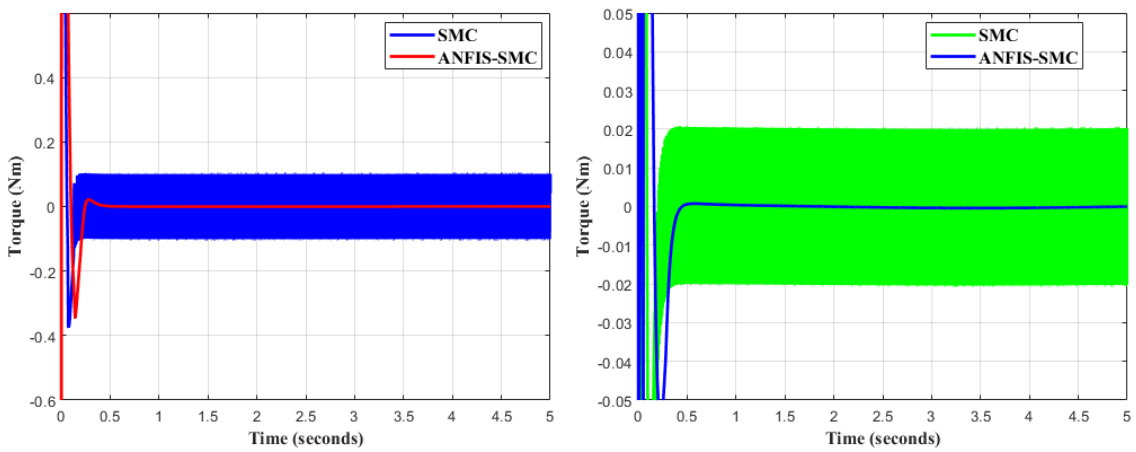


Figure 5.8: Helical Virtual controller response.

5.3 Comparison of SMC based ANFIS with SMC

5.3.1 Response for Helical Cylindrical Trajectory

The attitude roll angle controller U_2 and pitch angle controller U_3 control the reference trajectory and yaw angle controller U_4 are responsible for controlling the specified altitude provided by the task manager generated from outer loop controller. To demonstrate the performance of ANFIS-SMC compared to SMC, with a Helical trajectory is employed. This trajectory is complex; therefore, achieving the best controller performance is essential. The simulation result for the Adaptive Neuro-Fuzzy inference system controller based sliding mode controller shows good tracking performance with fast convergence of the states to the desired reference signals. Where as the sliding mode controller gives chattering and low performance as shown in Figure 5.9 (a),(c),(e). The suggested ANFIS-based SMC controller minimizes chattering in the SMC's control inputs. as shown in Figure 5.9 (b),(d),(f).



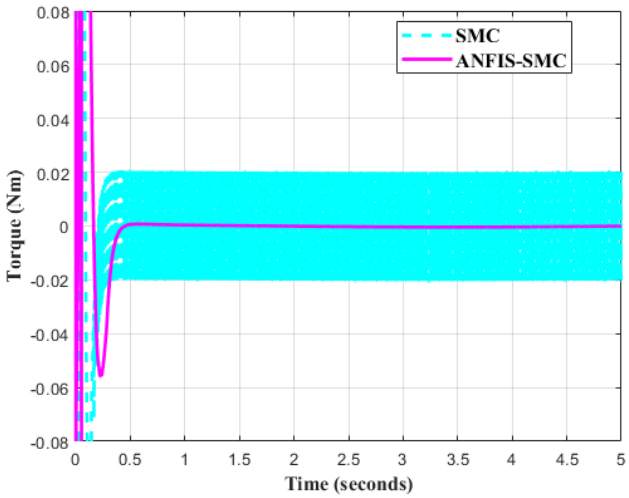


Figure 5.9: Controllers Effort in Helical Trajectory Tracking

5.3.2 Response for Infinity Trajectory

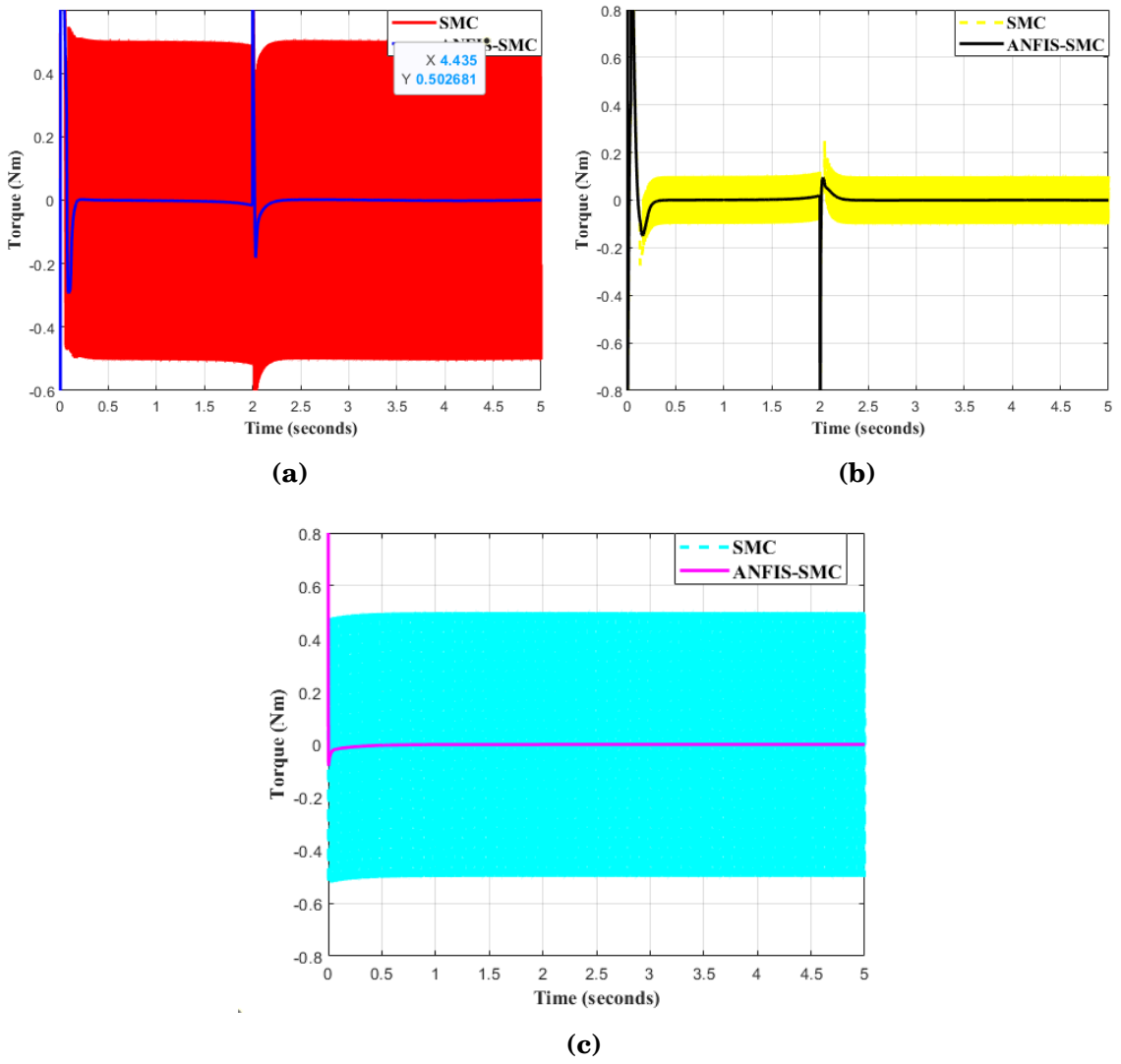


Figure 5.10: Controllers Effort in Infinity Trajectory Tracking

The ANFIS controller achieves good tracking performance of the states to the intended reference signals based on the simulation result for the above infinite trajectory tracking control effort. On the other hand, as Figure 5.10 (a), (c), and (e) demonstrate, the sliding mode controller produces chattering and poor performance. As seen in Figure 5.9 (b), (d), and (f), the suggested adaptive neuro-fuzzy inference system controller-based sliding mode controller controller reduce chattering that arises in the control effort of SMC.

For each of the cases the propellers revolutions per minute (RPM) are set in a way that results in the occurrence of the controllable variables which causes one of four basic

quadcopter movements in space with the Quadcopter frame is assumed to be rigid and the only thing that has direct influence on the quadcopter movement are each motor's RPM [45]. To verify that the system equations of the quadcopter are correct and ensure proper functionality within the recommended range, some tests are conducted. The first test presents the results for altitude, roll, pitch, and yaw when all four rotors are set to the calculated speed. The calculated rotor speeds (in RPM) based on the control inputs U_1, U_2, U_3, and U_4 from the simulation result and using equation 3.49 we get:

$$\begin{aligned} \text{Rotor 1:} &\quad 1931.04 \ \text{RPM}, \\ \text{Rotor 2:} &\quad 1902.52 \ \text{RPM}, \\ \text{Rotor 3:} &\quad 1930.84 \ \text{RPM}, \\ \text{Rotor 4:} &\quad 1896.57 \ \text{RPM} \end{aligned}$$

The system equations that link the control inputs to the angular velocities of each rotor are solved to obtain the above values. The comparatively near readings for the rotor speeds show that the motor speeds of the quadcopter are balanced and produce the necessary torques to keep the correct orientation with the recommended range. The fact that the calculated rotor speeds in this instance fall within the recommended range of the control inputs which is essential for stable flying. Based on the given control inputs, the computed values indicate that the system is well-tuned to achieve the necessary altitude, roll, pitch, and yaw control.

Also from the simulation result, the thrust force U_1 (which is the total upward force generated by the rotors) is calculated to be 4.821N. This thrust force should be at least equal to or greater than the gravitational force acting on the quadcopter. The gravitational force is determined by multiplying the mass of the quadcopter ($m = 0.486$) by the gravitational acceleration ($g = 9.81\,\text{m/s}^2$). In this case, the calculated thrust force U_1 is sufficient to counteract the gravitational force, ensuring better flight.

5.4 Matrix Of Performance

In this section ITSE (Integral Time Square Error) is used for this thesis as a performance index because of by measuring the error magnitude over time in a particular manner, ITSE is a helpful statistic for comparing various control strategies and identifying which control arrangement produces the best error reduction and tracking performance. The ITSE are given for altitude and position trajectories.

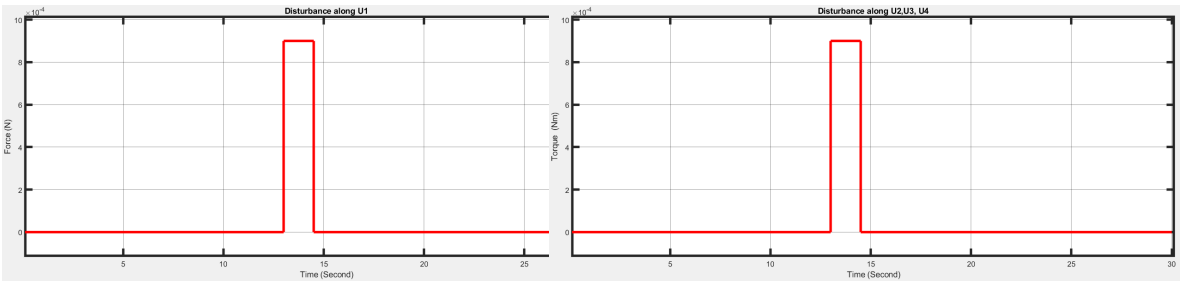
Table 5.1: Integral Time Square Error (ITSE) performance index for Infinity

Input/State	SMC	ANFIS-SMC
ITSE(x)	0.9617	0.0784
ITSE(y)	0.3678	0.03978
ITSE(z)	0.06323	0.01143
ITSE(ϕ)	0.5699	0.02826
ITSE(θ)	0.07307	0.006132
ITSE(ψ)	5.622	0.5672

As shown in the table 5.1 shows that the ITSE values for transnational x, y, and z states remain unaffected, showing zero percentage difference across all conditions. However, the rotational states (roll (ϕ), pitch (θ) and yaw (ψ) angles trajectory) exhibit with percentage differences of 91.83%, 81.15%, 81.88%, 95.06%, 91.82%, and 89.83% respectively compared to SMC.

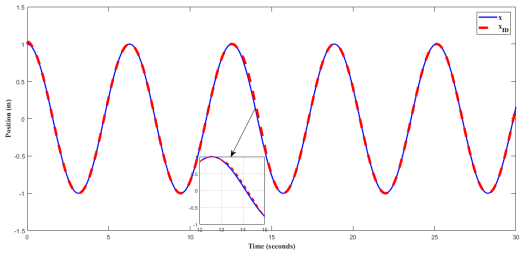
5.4.1 Infinity Trajectory with Input Disturbance

The capacity of the suggested Adaptive Neuro-Fuzzy inference system based sliding mode controller to reject input disturbances and get over parametric uncertainty is evaluated in this section. Bounded disturbance is used in real-world applications to track response for an infinity reference trajectory in order to demonstrate the effectiveness and resilience of the ANFIS-SMC technique in tracking trajectory.

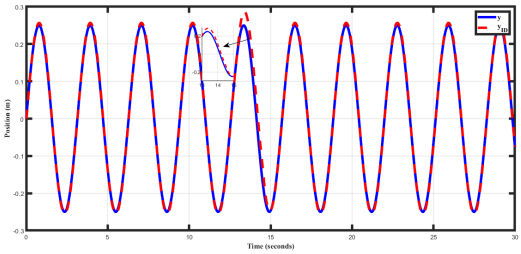


(a) **(b)**

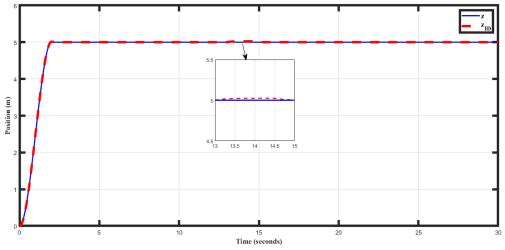
Figure 5.11: Applied Disturbance along U1, U2 and U3.



(a) Position along x-axis.



(b) Position along y-axis.



(c) Position along z-axis.

Figure 5.12: Infinity trajectory tracking with input disturbance for position control.

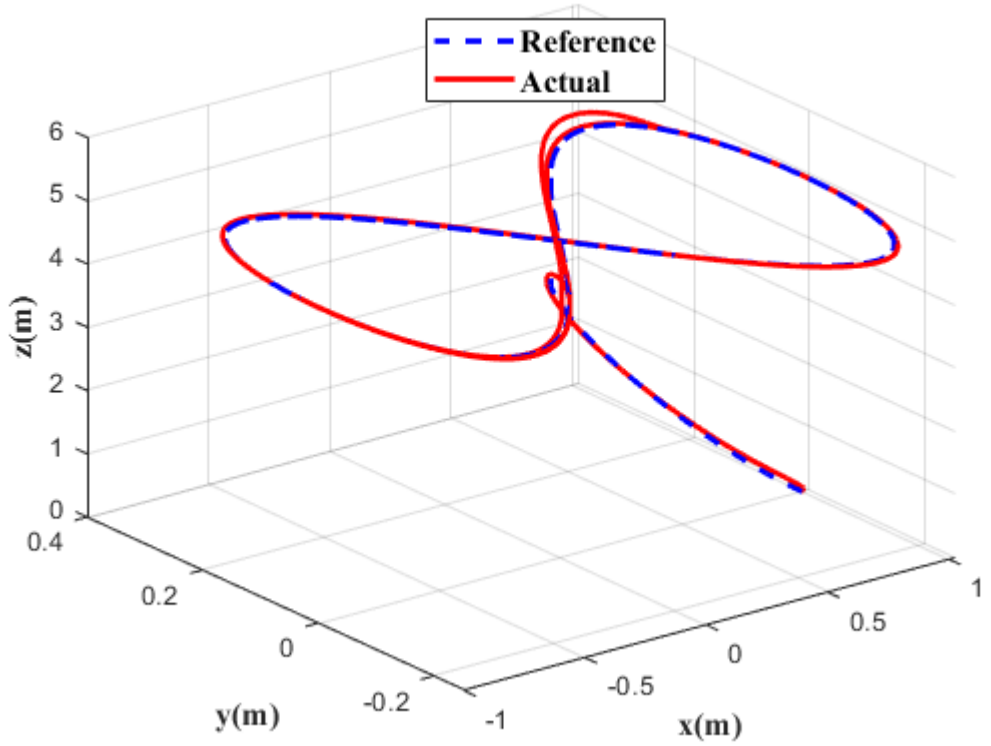
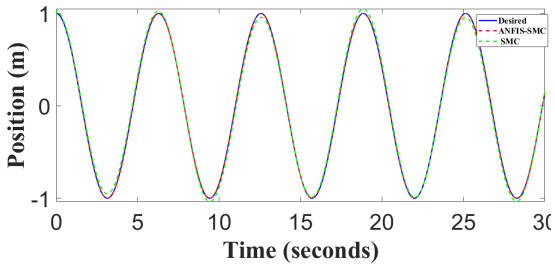


Figure 5.13: Infinity Response with Disturbance.

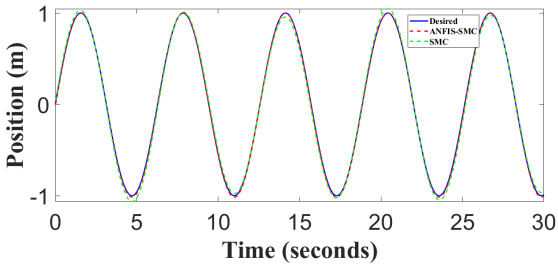
The external disturbance is represented by a random signal exhibiting differing force amplitudes along U1, U2, and U3 at different time intervals shown in Figure 5.11 illustrates the quadcopter's trajectory during the experiment. It commences its ascent from ground level, reaching an altitude of 6 meters within a mere 2.5 seconds. This trajectory persists for the next 13 seconds, during which the quadcopter maintains its position at the 6 meter. However, at the 13th second, it encounters a 8 N force in the positive Z direction, causing a temporary ascent of 0.2 meters from its initial position. It travels with this height from 13th to 15th second. Remarkably, within a mere 2.5 seconds, it ascends back to its original altitude of 6 meters, displaying the controller's remarkable disturbance rejection capability. These remarkable control outcomes underscore the system's ability to efficiently reject disturbances and maintain the desired flight trajectory, emphasizing the robustness and reliability of the SMC based ANFIS control strategy. The system response of the proposed controllers under disturbances in 3D is presented in Figure 5.13. From the simulation result, it is observed that the suggested controller by automatically tuning its parameters and give the amount of appropriate force and torque to the quadcopter, it achieves better performance of the desired trajectory tracking in the presence of disturbances.

5.4.2 Helical Trajectory with Parameter Variation

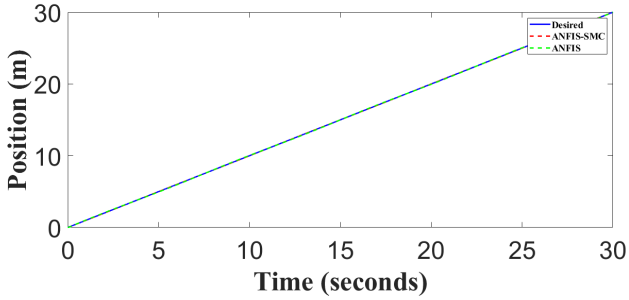
As demonstrated by the results in 5.14 **??** (a), (b), and (c), the SMC-based ANFIS scheme is efficient and robust in tracking trajectory when there is a 51.2% parameter variation in the quadcopter mass and moment of inertia. Helical trajectory is generated to incorporate both translational and rotational components, and testing the robustness and performance of the proposed controller in disturbance rejection capability in both movement types. Figure 5.14 - 5.15 clearly shows that through modeling, simulation, and comparison verified that the proposed intellegent When it came to trajectory tracking, ANFIS-SMC outperformed the conventional SMC controller in terms of stability and convergence resilience.



(a) Helical trajectory position along x-axis.

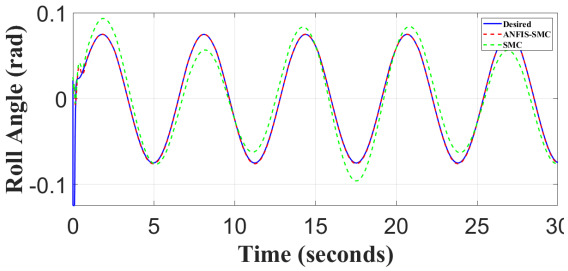


(b) Helical trajectory position along y-axis.

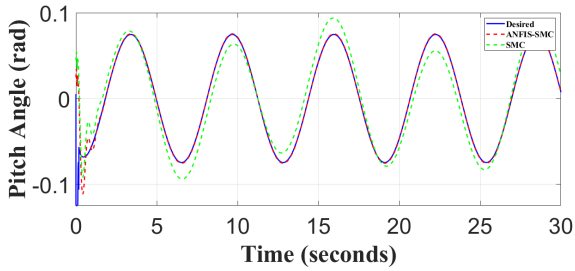


(c) Helical trajectory position along z-axis.

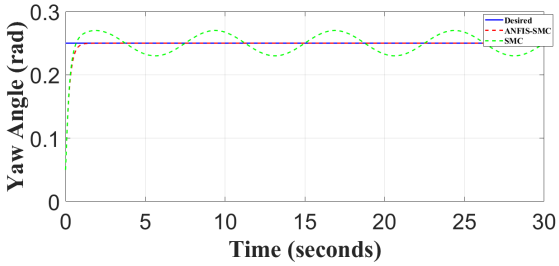
Figure 5.14: Helical Trajectory Tracking with parameter variation for position control



(a)



(b)



(c)

Figure 5.15: Helical Trajectory Tracking with parameter variation for attitude control

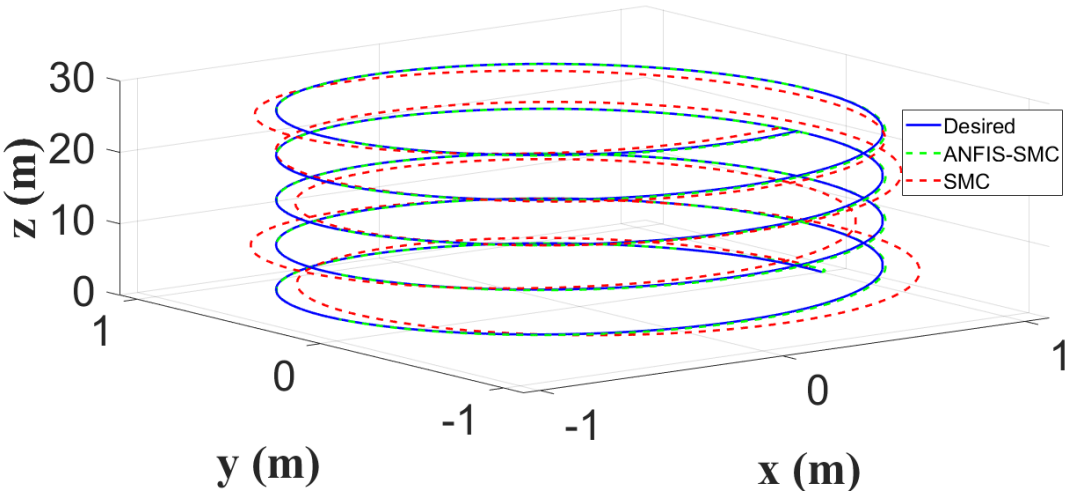


Figure 5.16: Helical response with parameter variation.

5.4.3 Quantitative Analysis for SMC based ANFIS and SMC in Control Efforts

As illustrated in figure 5.9, 5.10, the amount of chattering in SMC and ANFIS-SMC may be statistically measured for various control signals. The root mean square (RMS) value can be used as an index to express it. The square root of the arithmetic mean of the squares of the values is the RMS value of a continuous time signal or a collection of values. It shows the signal's average power over a specified period of time. Therefore, the energy content of a signal's chattering profile has a direct correlation with RMS. In regard to this, a signal with a high RMS value is anticipated to have a high chattering rate.

Table 5.2: Chattering performance Index in Helical Trajectory

Control Effort	SMC	ANFIS-SMC
U_2	0.5154	0.07477
U_3	0.2234	0.07402
U_4	0.08543	0.09438

Table 5.3: Chattering performance Index in Infinity Trajectory

Control Effort	SMC	ANFIS-SMC
U_2	0.6498	0.1234
U_3	0.3155	0.0104
U_4	0.5126	0.0828

The comparison shows that SMC-based ANFIS greatly lowers chattering in control efforts U_2 and U_3, reaching improvements of roughly 85.49% and 66.87%, respectively, when compared to SMC, as tables 5.2 demonstrate. This demonstrates ANFIS based SMC superior capability in minimizing chattering and enhancing system smoothness for those control actions.But for U_4, are nearly same. This is because the helical trajectory contains very little variation of real data along the yaw angle, meaning that the system requires comparatively little mathematical computation. And 5.3 analysis for the infinity trajectory demonstrates that ANFIS-SMC markedly outperforms SMC in reducing chattering across all control efforts. Specifically, ANFIS-SMC achieves reductions of approximately 81.02% in U_2, 96.71% in U_3, and 83.84% in U_4 compared to SMC. These findings demonstrate the strong ability of ANFIS-based SMC to suppress high-frequency oscillations, resulting in smoother control signals.

Chapter 6

Conclusion and Recommendation

6.1 Conclusion

The paper proposes an sliding mode controller based Adaptive Neuro-Fuzzy inference system controller to improve quadcopter trajectory tracking performance. In order to increase output in compliance with a set of guidelines, the proposed ANFIS based SMC is employed. The position control and attitude control were tuned by a neuro-fuzzy adaptive inference method, which enabled the system to respond to environmental changes. In order to verify the effectiveness of the proposed controller, the study then compared the performance of the standard SMC controller with that of the suggested Adaptive Neuro-Fuzzy inference system based sliding mode controller.

The results of the analysis showed that the suggested controller could maintain by adapting to changes in the system and converge to the desired trajectory with remarkable stability and accuracy. Compared to the SMC controller, it also had a faster rate of convergence and less chattering. The study also demonstrated the effectiveness of the ANFIS-based SMC controller by evaluating its performance under different conditions. It was discovered that the proposed SMC-based ANFIS controller was more robust and outperformed the SMC controller in terms of trajectory tracking in many scenarios where external disturbances might affect the system's dynamics. By adapting control parameters to changes, the ANFIS control technique might keep the control system stable.

On the other hand, the SMC approach's predetermined control parameters might not be the best in these circumstances, leading to less reliable control performance with the chattering phenomena. The research results presented in this work demonstrate

that the suggested controller outperformed the conventional SMC controller in terms of stability, high precision, robustness, and fast convergence. Because of its adaptive mechanism, the controller can retain high precision and stability while reacting softly to changes in the dynamics of the system. These results offer important insights into the design of efficient control systems, which may aid in the creation of responsive and reliable control strategies for UAV trajectory tracking.

6.2 Recommendation

There is still a lot of work required in this field of control and in the field of the quadcopter, The need for a more realistic model to cover more aspects from the real life need more research and development.

- The parameters for ANFIS training would be much better in the future to use an optimization technique or integrating the SMC with other controllers to have a perfect training data.
- Experimental testing and validation of the proposed controllers.
- Finally to start testing on a real system for the purpose of creating more robust quadcopter and achieve complicated tasks.

References

[1] S. A. H. Mohsan, N. Q. H. Othman, Y. Li, M. H. Alsharif, and M. A. Khan, "Unmanned aerial vehicles (uavs): Practical aspects, applications, open challenges, security issues, and future trends," *Intelligent Service Robotics*, vol. 16, no. 1, pp. 109–137, 2023.

[2] G. Hoffmann, H. Huang, S. Waslander, and C. Tomlin, "Quadrotor helicopter flight dynamics and control: Theory and experiment," in *AIAA guidance, navigation and control conference and exhibit*, 2007, p. 6461.

[3] T. Chevet, M. Makarov, C. S. Maniu, I. Hinostroza, and P. Tarascon, "State estimation of an octorotor with unknown inputs. application to radar imaging," in *2017 21st International Conference on System Theory, Control and Computing (ICSTCC)*. IEEE, 2017, pp. 723–728.

[4] F. Santoso, M. A. Garratt, and S. G. Anavatti, "Adaptive neuro-fuzzy inference system identification for the dynamics of the ar. drone quadcopter," in *2016 International Conference on Sustainable Energy Engineering and Application (ICSEEA)*. IEEE, 2016, pp. 55–60.

[5] V. I. Utkin, *Sliding modes in control and optimization*. Springer Science & Business Media, 2013.

[6] D. Shiferaw and R. Mitra, "Neuro-fuzzy sliding mode controller: design and stability analysis," *International Journal of Computational Intelligence Studies*, vol. 1, no. 3, pp. 242–255, 2010.

[7] B. Mrinal, "Adaptive network based fuzzy inference system (anfis) as a tool for system identification with special emphasis on training data minimization," in *Department of Electronics and Communication Engineering*. Indian Institute of Technology Guwahati, 2008, vol. 141.

[8] H. Benbouhenni, "Anfis-sliding mode control of a dfig supplied by a two-level svpwm technique for wind energy conversion system," *International Journal of Applied Power Engineering*, vol. 9, no. 1, pp. 36–47, 2020.

[9] A. Dorzhigulov, B. Bissengaliuly, B. F. Spencer, J. Kim, and A. P. James, "Anfis based quadrotor drone altitude control implementation on raspberry pi platform," *Analog Integrated Circuits and Signal Processing*, vol. 95, pp. 435–445, 2018.

[10] A. K. Ravandi, E. Khanmirza, and K. Daneshjou, "Hybrid force/position control of robotic arms manipulating in uncertain environments based on adaptive fuzzy sliding mode control," *Applied Soft Computing*, vol. 70, pp. 864–874, 2018.

[11] K. Runcharoon and V. Srichatrapimuk, "Sliding mode control of quadrotor," in *2013 The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE)*. IEEE, 2013, pp. 552–557.

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

[13] M. Öztürk, "A modified anfis system for aerial vehicles control," 2022.

[14] B. Selma, S. Chouraqui, and H. Abouaïssa, "Optimization of anfis controllers using improved ant colony to control an uav trajectory tracking task," *SN Applied Sciences*, vol. 2, no. 5, p. 878, 2020.

[15] P. Darwito and N. Indayu, "Adaptive neuro-fuzzy inference system based on sliding mode control for quadcopter trajectory tracking with the presence of external disturbance," *J. Intell. Syst. Control*, vol. 2, no. 1, pp. 33–46, 2023.

[16] S. Rezazadeh, M. A. Ardestani, and P. S. Sadeghi, "Optimal attitude control of a quadrotor uav using adaptive neuro-fuzzy inference system (anfis)," in *The 3rd International Conference on Control, Instrumentation, and Automation*. IEEE, 2013, pp. 219–223.

[17] M. YALEW, "Design neuro-fuzzy sliding mode controller for switched reluctance motor speed control applicable to electric vehicle," Ph.D. dissertation, 2022.

[18] M. Manimaraboopathy, H. V. Christopher, S. Vignesh *et al.*, "Unmanned fire extinguisher using quadcopter," *International Journal on Smart Sensing and Intelligent Systems*, vol. 10, no. 5, pp. 471–481, 2017.

[19] R. Beard, "Quadrotor dynamics and control rev 0.1," 2008.

[20] M. Islam, M. Okasha, and M. Idres, "Dynamics and control of quadcopter using linear model predictive control approach," in *IOP conference series: materials science and engineering*, vol. 270, no. 1. IOP Publishing, 2017, p. 012007.

[21] Z. Tahir, W. Tahir, and S. A. Liaqat, "State space system modelling of a quad copter uav," *arXiv preprint arXiv:1908.07401*, 2019.

[22] B. Derseh, L. Negash, and C. M. Abdissa, "Robust pso tuned fosmc for altitude stabilization and trajectory tracking of agricultural monitoring uav," *Authorea Preprints*, 2023.

[23] H. Bouadi, M. Bouchoucha, and M. Tadjine, "Sliding mode control based on backstepping approach for an uav type-quadrotor," *International Journal of Mechanical and Mechatronics Engineering*, vol. 1, no. 2, pp. 39–44, 2007.

[24] A. R. Firdaus, "Design of sliding mode-based nonlinear control systems with nonlinear full-order state observers for underactuated coupled systems," Ph.D. dissertation, University of Sheffield, 2018.

[25] T. Wang, C. Sabourin, and K. Madani, "Anfis controller for non-holonomic robots," 2011.

[26] A. K. Pouya, "Design of adaptive neural fuzzy controller for speed control of bldc motors," *Majlesi Journal of Electrical Engineering*, vol. 11, no. 1, p. 37, 2017.

[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] H. talla Mohamed, "Dynamic modeling and control of a quadrotor using linear and nonlinear approaches," *School of Sciences and Engineering, Spring*, 2014.

[29] K. Worden, G. Tsialiamanis, E. Cross, and T. Rogers, "Artificial neural networks," in *Machine Learning in Modeling and Simulation: Methods and Applications*. Springer, 2023, pp. 85–119.

[30] W. BRUCE and E. VON OTTER, "Artificial neural network autonomous vehicle: Artificial neural network controlled vehicle," 2016.

[31] R. E. Akubo, "Design of a neural network architecture for traffic light detection in autonomous vehicles," Ph.D. dissertation, 2019.

[32] S. N. Rajguru, P. Shah, N. Rayavarapu, and P. Tupe, "Application of fuzzy controllers for flow control processes in chemical industries," *Journal of Chemical and Pharmaceutical Research*, vol. 6, no. 12, pp. 403–410, 2014.

[33] Y. Wang and Y. Chen, "A comparison of mamdani and sugeno fuzzy inference systems for traffic flow prediction." *J. Comput.*, vol. 9, no. 1, pp. 12–21, 2014.

[34] N. Walia, H. Singh, and A. Sharma, "Anfis: Adaptive neuro-fuzzy inference system-a survey," *International Journal of Computer Applications*, vol. 123, no. 13, 2015.

[35] J.-S. Jang and C.-T. Sun, "Neuro-fuzzy modeling and control," *Proceedings of the IEEE*, vol. 83, no. 3, pp. 378–406, 1995.

[36] K. Rezaei, R. Hosseini, M. Mazinani *et al.*, "A fuzzy inference system for assessment of the severity of the peptic ulcers," *Computer Science & Information Technology*, pp. 263–271, 2014.

[37] H. Hamdan and J. M. Garibaldi, "Adaptive neuro-fuzzy inference system (anfis) in modelling breast cancer survival," in *International Conference on Fuzzy Systems*. IEEE, 2010, pp. 1–8.

[38] S. S. Roy, "Design of adaptive neuro-fuzzy inference system for predicting surface roughness in turning operation," 2005.

[39] D. Patel and F. Parekh, "Flood forecasting using adaptive neuro-fuzzy inference system (anfis)," *International Journal of Engineering Trends and Technology (IJETT)*, vol. 12, no. 10, pp. 510–514, 2014.

[40] M. Pratama, S. Rajab, and E. M. Joo, "Extended approach of anfis in cascade control," *International Journal of Computer and Electrical Engineering*, vol. 3, no. 4, pp. 572–576, 2011.

[41] T. Uçar, A. Karahoca, and D. Karahoca, "Tuberculosis disease diagnosis by using adaptive neuro fuzzy inference system and rough sets," *Neural Computing and Applications*, vol. 23, pp. 471–483, 2013.

[42] B. Selma, S. Chouraqui, B. Selma, and H. Abouaïssa, "Anfis controller design based on pigeon-inspired optimization to control an uav trajectory tracking task," *Iran Journal of Computer Science*, vol. 4, no. 1, pp. 1–16, 2021.

[43] J.-S. Jang, "Anfis: adaptive-network-based fuzzy inference system," *IEEE transactions on systems, man, and cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.

[44] I. F. Bouguenna, A. Azaiz, A. Tahour, and A. Larbaoui, "Electronic differential and neuro-fuzzy sliding mode control with extended state observer for an electric vehicle system," in *E3S Web of Conferences*, vol. 61. EDP Sciences, 2018, p. 00007.

[45] Z. Benić, P. Piljek, and D. Kotarski, "Mathematical modelling of unmanned aerial vehicles with four rotors," *Interdisciplinary Description of Complex Systems: INDECS*, vol. 14, no. 1, pp. 88–100, 2016.

Appendix A

ANFIS Toolbox Parameters

Table A.1: Summary of Fuzzy Inference System Parameters

Parameter	**Description/Value**
Opt. method	Hybrid learning
Or method	Probor
And method	Product
Implication	Min
Training pairs	351
Agg. method	Max
Checking pairs	150
Defuzzification	Wtaver
Min. training RMSE	0.0014424
Inputs	1
Input MFs	10
Outputs	1
Input MF type	Gaussian (gaussmf, guass2mf, psigmf)
Output MF type	Linear,Constant
Epochs	100
Fuzzy rules	10
FIS structure	Sugeno 1st order
No. of Nodes	44
No. of Parameters	40
No. of trainig data pairs	3005

Appendix B

Control signals (ANFIS Toolbox)

For the virtual controller signals U_x, U_y, and U_z and the control effort signals U_2, U_3, and U_4 using Adaptive Neuro Fuzzy Inference System using MATLAB procedure. We have defined ten membership functions (MF1 to MF10) for both inputs and output datas. Their respective plots are shown below.

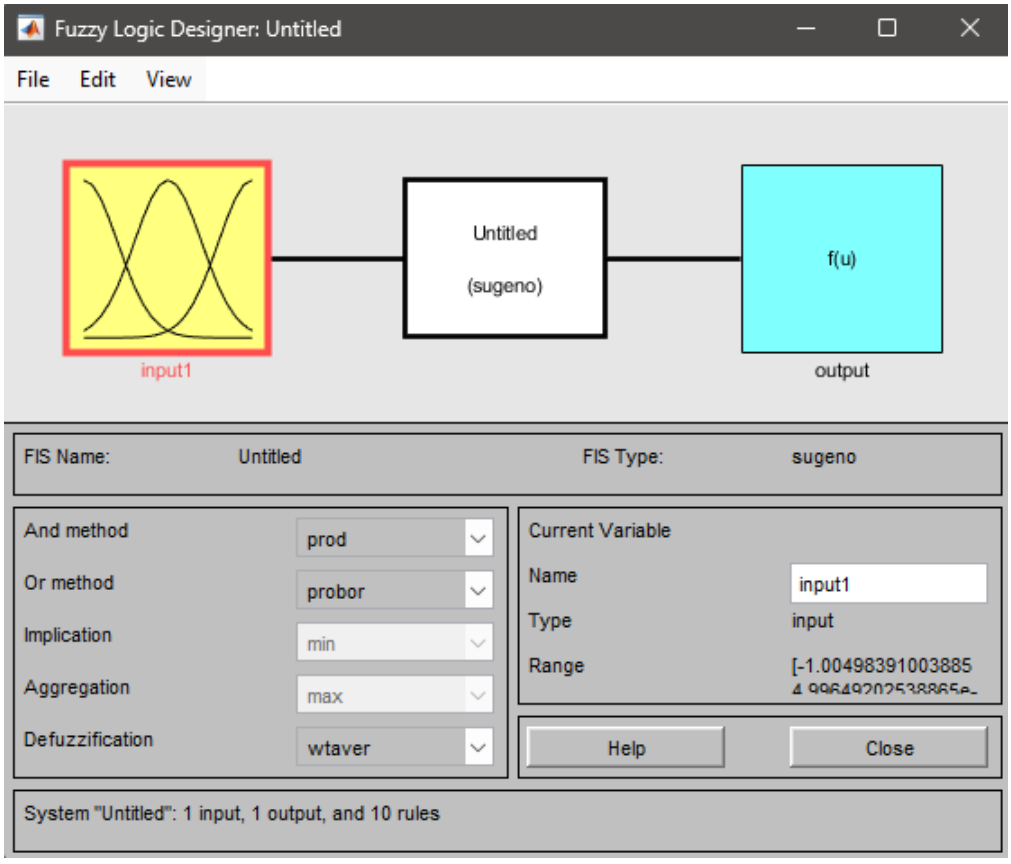


Figure B.1: FIS Property Editor for Sugeno type Fuzzy Model.

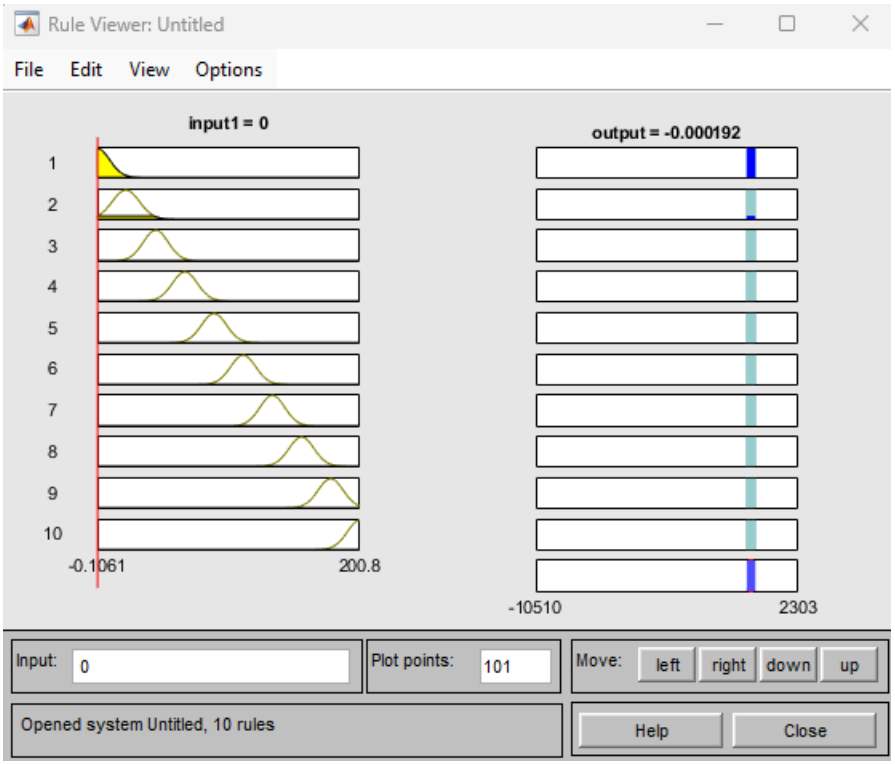


Figure B.2: ANFIS Rule Viewer.

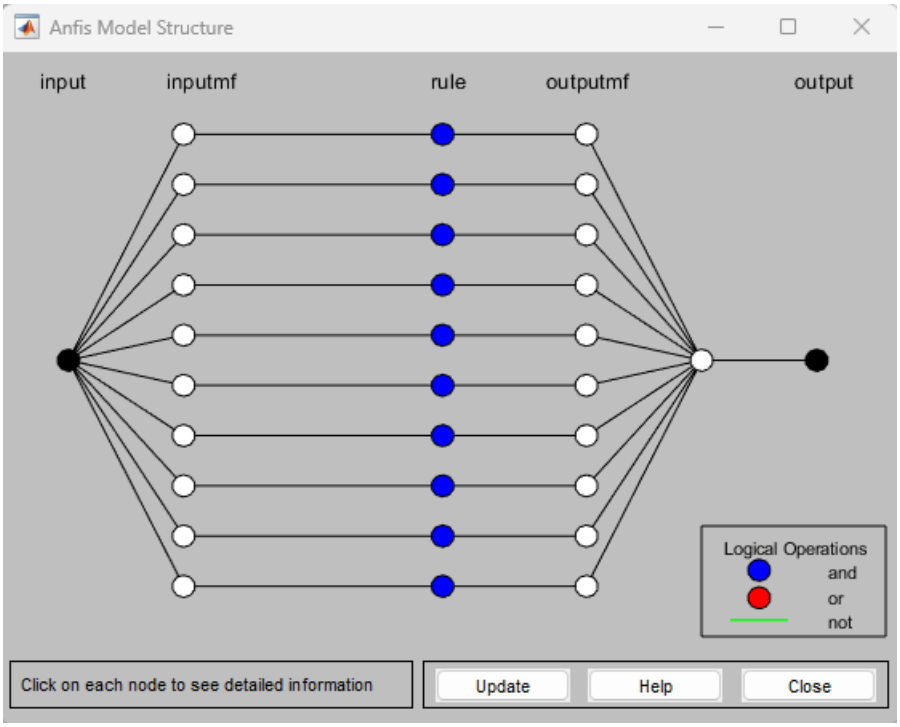


Figure B.3: ANFIS Model structure.

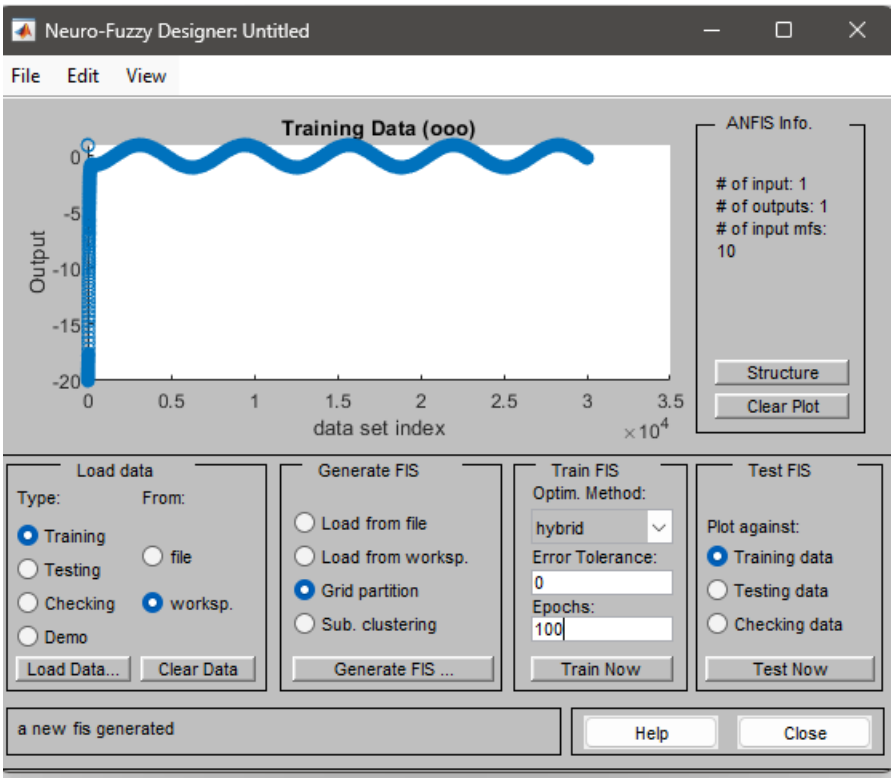


Figure B.4: Surface viewer after ANFIS Training.

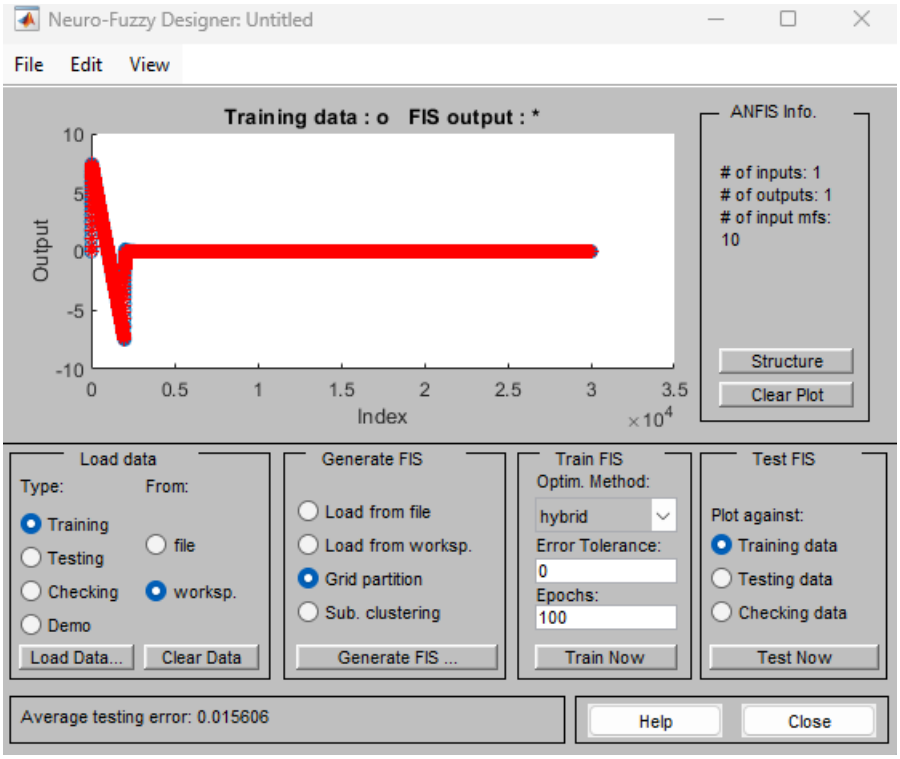


Figure B.5: ANFIS Model training and testing.

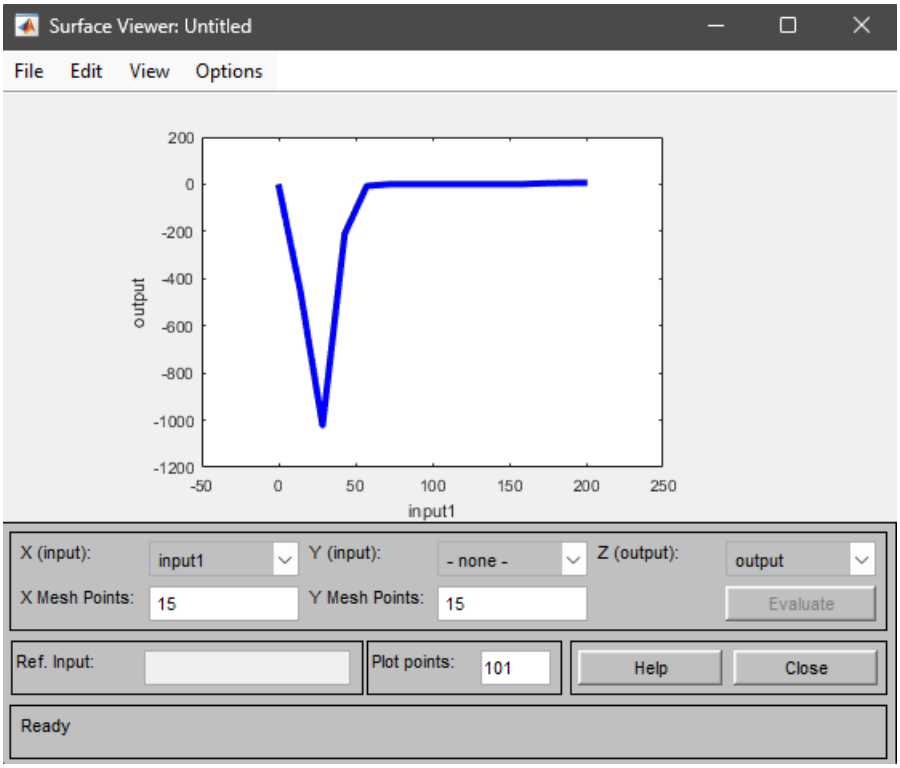


Figure B.6: ANFIS Fuzzy Input/Output relation.

Appendix C

The weights of the fuzzy rules code(ANFIS Toolbox)

The weights of the fuzzy rules system maps one input to one output using Gaussian membership functions for input and linear membership functions for output. Each rule corresponds to a direct mapping between specific membership functions. The weighted average defuzzification ensures smooth transitions across outputs based on the degree of activation for each rule.

[System]
Name='ANz'
Type='sugeno'
Version=2.0
NumInputs=1
NumOutputs=1
NumRules=10
AndMethod='prod'
OrMethod='probor'
ImpMethod='prod'
AggMethod='sum'
DefuzzMethod='wtaver'

[Input1] Name='input1'
Range=[-0.0742490784552352 0.0717093669498595]

NumMFs=10
MF1='in1mf1':'gaussmf',[0.0068869827565942 -0.0742490784552352]
MF2='in1mf2':'gaussmf',[0.0068869827565942 -0.0580314734102247]
MF3='in1mf3':'gaussmf',[0.0068869827565942 -0.0418138683652142]
MF4='in1mf4':'gaussmf',[0.0068869827565942 -0.0255962633202036]
MF5='in1mf5':'gaussmf',[0.0068869827565942 -0.0093786582751931]
MF6='in1mf6':'gaussmf',[0.0068869827565942 0.00683894676981742]
MF7='in1mf7':'gaussmf',[0.0068869827565942 0.0230565518148279]
MF8='in1mf8':'gaussmf',[0.0068869827565942 0.0392741568598385]
MF9='in1mf9':'gaussmf',[0.0068869827565942 0.055491761904849]
MF10='in1mf10':'gaussmf',[0.0068869827565942 0.0717093669498595]

[Output1]
Name='output'
Range=[-7.49987288749015 7.42155094012359]
NumMFs=10
MF1='out1mf1':'linear',[100.607968922973 -0.0286293582835246]
MF2='out1mf2':'linear',[101.307819244468 0.00501009570637556]
MF3='out1mf3':'linear',[102.978305814863 0.0589562079218655]
MF4='out1mf4':'linear',[106.156889625714 0.100256557281319]
MF5='out1mf5':'linear',[109.337812781324 0.0274345469163314]
MF6='out1mf6':'linear',[90.9095613490933 -0.00947667024574773]
MF7='out1mf7':'linear',[94.0799836657898 0.100380531414663]
MF8='out1mf8':'linear',[98.9667371425532 0.0350505985423636]
MF9='out1mf9':'linear',[103.828040183954 -0.142282794243302]
MF10='out1mf10':'linear',[109.980839492509 -0.464361031321927]

[Rules]

1, 1 (1) : 1

2, 2 (1) : 1

3, 3 (1) : 1

4, 4 (1) : 1

5, 5 (1) : 1

6, 6 (1) : 1

7, 7 (1) : 1

8, 8 (1) : 1

9, 9 (1) : 1

10, 10 (1) : 1

Appendix D

Lyapunov Stability Analysis for SMC

Lyapunov Stability Analysis for Sliding Mode Control (SMC) of a Quadcopter

Stability Analysis for X Position Define the sliding surface:

$$S_1 = a_1 e_x + \dot{e}_x$$

Lyapunov candidate function:

$$V(S_1) = \frac{1}{2} S_1^2$$

Taking the derivative:

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

Using control law:

$$U_x = -k_x \text{sign}(S_1) + \text{compensation terms}$$

Then,

$$\dot{S}_1 = a_1 \dot{e}_x + \ddot{e}_x = -\frac{k_x}{m} \text{sign}(S_1) + \text{bounded terms} \Rightarrow \dot{V}(S_1) = -\frac{k_x}{m} |S_1| < 0$$

Conclusion: $k_x > 0$ ensures asymptotic stability.

Stability Analysis for Y Position

$$S_2 = a_2 e_y + \dot{e}_y$$

$$V(S_2) = \frac{1}{2} S_2^2, \quad \dot{V}(S_2) = S_2 \dot{S}_2$$

$$U_y = -k_y \text{sign}(S_2) + \text{compensation terms} \Rightarrow \dot{V}(S_2) = -\frac{k_y}{m} |S_2| < 0$$

Conclusion: $k_y > 0$ ensures stability.

Stability Analysis for Z (Altitude)

$$S_3 = a_3 e_z + \dot{e}_z, \quad V(S_3) = \frac{1}{2} S_3^2$$

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

$$U_1 = -k_z \text{sign}(S_3) + m(g - a_3 \dot{e}_z - \ddot{Z}_d)$$

$$\dot{S}_3 = a_3 \dot{e}_z + \ddot{e}_z = -\frac{k_z}{mC(\theta)C(\phi)} \text{sign}(S_3) \Rightarrow \dot{V}(S_3) = -\frac{k_z}{mC(\theta)C(\phi)} |S_3| < 0$$

Conclusion: $k_z > 0$ guarantees stability.

Stability Analysis for Roll Angle ϕ

$$S_4 = a_4 e_\phi + \dot{e}_\phi, \quad V(S_4) = \frac{1}{2} S_4^2$$

$$\dot{V}(S_4) = S_4 \dot{S}_4$$

$$U_2 = -k_\phi \text{sign}(S_4) + (J_z - J_y)\dot{\theta}\dot{\psi} + J_x(a_4 \dot{e}_\phi + \ddot{\phi}_d) \Rightarrow \dot{V}(S_4) = -\frac{k_\phi}{J_x} |S_4| < 0$$

Conclusion: $k_\phi > 0$ ensures axis stability.

Stability Analysis for Pitch Angle θ

Stability Analysis for Angle θ

$$S_5 = a_5 e_\theta + \dot{e}_\theta, \quad V(S_5) = \frac{1}{2} S_5^2$$

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

$$U_3 = -k_\theta \text{sign}(S_5) + (J_x - J_z)\dot{\phi}\dot{\psi} + J_y(a_5\dot{e}_\theta + \ddot{\theta}_d) \Rightarrow \dot{V}(S_5) = -\frac{k_\theta}{J_y}|S_5| < 0$$

Conclusion: $k_\theta > 0$ ensures axis convergence.

Stability Analysis for Yaw Angle ψ

$$S_6 = a_6 e_\psi + \dot{e}_\psi, \quad V(S_6) = \frac{1}{2}S_6^2$$

$$\dot{V}(S_6) = S_6\dot{S}_6$$

$$U_4 = -k_\psi \text{sign}(S_6) + (J_y - J_x)\dot{\phi}\dot{\theta} + J_z(a_6\dot{e}_\psi + \ddot{\psi}_d) \Rightarrow \dot{V}(S_6) = -\frac{k_\psi}{J_z}|S_6| < 0$$

Conclusion: $k_\psi > 0$ ensures stability of dynamics.