الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة زيان عاشور بالجلفة
Université Ziane Achour de Djelfa

كلية العلوم والتكنولوجيا
Faculté des Sciences et de la Technologie

Département : Génie Electrique

N° d'ordre : ……./ 2026

Autorisation de soutenance N° : 097/2026

THESE DE DOCTORAT

Doctorat 3ème Cycle (D-LMD)

Présentée par

TOUDJI Khaled

En vue de l'obtention du diplôme de Docteur en 3ème Cycle D-LMD

Filière : Electronique
Spécialité : Electronique Embarquée

Sujet

Modélisation et Commande Intelligente d'un Drone Octorotor

Soutenue publiquement, le 06 /07/2026, devant le jury composé de :

Nom et Prénom	Grade	Etablissement	Désignation
Mr. AISSAOUI Djelloul	Professeur	Université de Djelfa	Président
Mr. CHERROUN Lakhmissi	Professeur	Université de Djelfa	Directeur de thèse
Mr. NADOUR Mohamed	Professeur	Université de Djelfa	Co-directeur de thèse
Mr. ACHBI Mohamed Said	MCA	Université de Ouargla	Examinateur
Mr. AISSAT Sid-ali	MCA	Université de Djelfa	Examinateur

Université de Djelfa, FST, 2026

الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research

جامعة زيان عاشور بالجلفة
Ziane Achour University of Djelfa

كلية العلوم و التكنولوجيا
Faculty of Science and Technology

Department: Electrical Engineering

Order N° : . .... / 2026

Authorization N° 097/2026

DOCTORAL THESIS

3rd Cycle Doctoral (D-LMD)

Presented by:

TOUDJI Khaled

With a view to obtaining the doctoral diploma in 3rd Cycle Doctoral (D-LMD)

At: Electronic, Speciality: Embedded Electronic

Topic

Modeling and Intelligent Control of an Octorotor Drone

Supported, on 06/07/2026, before the jury composed of:

Last and first name	Grade	Affiliation	Designation
Mr. AISSAOUI Djelloul	Professor	University of Djelfa	President
Mr. CHERROUN Lakhmissi	Professor	University of Djelfa	Supervisor
Mr. NADOUR Mohamed	Professor	University of Djelfa	Co-Supervisor
Mr. ACHBI Mohamed Said	MCA	University of Ouargla	Examiner
Mr. AISSAT Sid-ali	MCA	University of Djelfa	Examiner

Djelfa University, FST - 2026

Appreciation

First and foremost, I give thanks to Allah for granting me the strength, patience, and determination that made the completion of this work possible.

This doctoral research was carried out within the Laboratory of Applied Automation and Industrial Diagnostics (LAADI) at the University of Djelfa. I would like to express my sincere gratitude to **Professor HAFAIFA Ahmed**, Director of the laboratory, for his support, guidance, and valuable advice throughout the course of this research.

I am deeply grateful to my thesis supervisor, **Professor CHERROUN Lakhmissi,** from the University of Djelfa, for accepting to supervise this work. His constant availability, insightful guidance, and continuous encouragement at every stage of the research were instrumental to the successful completion of this thesis. I extend to him my profound appreciation and respect.

My sincere thanks are also extended to my co-supervisor **Professor NADOUR Mohamed**, for sharing his experience and for his valuable advice, which greatly contributed to the enrichment of this research.

I would also like to thank the members of the doctoral jury who honored me by accepting to evaluate this work:

- **Prof. AISSAOUI Djelloul**, University of Djelfa, for the honor of presiding over the jury.
- **Dr. ACHEBI Mohamed Said**, an associate Professor at University of Ouargla, for agreeing to serve on the jury and for reviewing this manuscript;
- **Dr. AISSAT Sid-ali**, an associate Professor at University of Djelfa, for his time, interest, and evaluation of this work.

Finally, I would like to express my heartfelt gratitude to my friends and colleagues for their support and encouragement. I also sincerely thank all those who, directly or indirectly, contributed to the successful completion of this work under favorable conditions.

Dedication

I dedicate this work:

*To my **parents**, may **Allah preserve them**;*

*To my **brothers and sisters**;*

*To my **wife** and my son **abdelmoiz**;*

*To my entire **extended family**, especially my uncles;*

To my friends.

CONTENT TABLE

FIGURES LIST

TABLES LIST

Symbols and Abbreviations :

UAV	Unmanned Ariel Viehcle
MALE	Medium Altitude Long Endurance
HALE	High Altitude Lone Endurance
6DOF	Six Degrees Of Freedom
FSMC	Fuzzy Slding Mode Controller
ANFIS	Adaptive Fuzzy Inference System
PSO	Particle Swarm Optimization
FLC	Fuzzy Logic Controller
PID	Proportionnal, Integral, Derivative
$(x_r, y_r z)$	The drone coordinate system
φ	Pitch angle
θ	Roll angle
Ψ	Yaw angle
b	Lift coefficient
d	Drag coefficient
K_1, K_2, K_3	Velocity forces
K_4, K_5, K_6	Friction moments
J_r	Rotor inertia
I_x, I_y, I_z	Axes inertia
m	UAV mass
l	UAV arm length

List of Works

The work carried out in this thesis is materialized through a set of international publications and conferences.

International Publications :

1. **K. Toudji**, L. Cherroun, M. Nadour, V. Puig and A. Hafaifa, "A Comparative Study between Different Fuzzy-based Nonlinear Controllers for UAV Coaxial Octorotor", International Journal of Fuzzy Systems (IJFS), Springer, vol. xx, no. x, pp. xx-xx, **2026** .https://doi.org/10.1007/s40815-025-02219-2. (**Classe A).**
2. **K. Toudji**, M. Nadour and L. Cherroun, "Fuzzy Logic Controllers Design for the Path Tracking of an Autonomous Coaxial Octorotor", Electrotechnica, Electronica, Automatica (EEA), **2024**, vol. 72, no. 1, pp. 39-46, ISSN 1582- 5175. https://doi.org/10.46904/eea.24.72.1.1108005**. (Classe B).**
3. **K. Toudji**, L. Cherroun, M. Nadour and A. Rabehi " Hybrid Control Structure employing Fuzzy-Sliding Mode Systems for the Guidance of UAV", Scientif Reports, Springer Nature, (2026). https://doi.org/10.1038/s41598-026-56625-z. (Classe A).

International Conferences :

1. **K. Toudji**, L. Cherroun and M. Nadour, "Modeling and Simulation of Coaxial Octorotor Unmanned Aerial Vehicle", *1st International Conf on Modern Electrical Engineering and Technology (ICMEET'22)*, February **2022**, Souk Ahras University, Algeria.
2. **K. Toudji**, L. Cherroun, M. Nadour, and A. Hafaifa"Neural Network Controller Design for the Path Tracking of a Coaxial Octorotor", *1st IEEE International Conference on Telecommunications and Intelligent Systems (ICTIS'2024*), December **2024**, Djelfa, Algeria. doi: 10.1109/ICTIS62692.2024.10894667

3. A.T. Guettaf, L. Cherroun, **K. Toudji**, M. Nadour, and A. Hafaifa"Detection of Leaves Health Status based on UAV Flight Images and Ensemble Learning Approaches", *International Conference on Applied Artificial Intelligence and Emerging Technologies 2025(AAIET'2025), Decem*ber **2025**, Djelfa, Algeria.

National Conferences :

1. **K. Toudji**, L. Cherroun, M. Nadour, "Energy Consumption Analysis of a Fuzzy Sliding Mode-Controlled Coaxial Octorotor for Solar Farm Surveillance", *The Second Algerian Conference on Mechanics and Solar Energy*, 26-27 April **2026,** University of Ouargla, Algeria.
2. **K. Toudji**, L. Cherroun, M. Nadour, "Energy Efficiency Comparison Between FLC and FSMC for Coaxial Octorotor UAVs in Solar Farm Surveillance Applications", *The Second Algerian Conference on Mechanics and Solar Energy*, 26-27 April **2026,** University of Ouargla, Algeria.

General Introduction

Autonomous mobile and aerial robotics is inherently a multidisciplinary research field, integrating control theory, computer vision, artificial intelligence, and embedded system design. In recent decades, significant attention has been devoted to the development of robotic platforms capable of operating autonomously in complex, unstructured, and dynamic environments. This growing interest is largely motivated by practical demands in application domains such as industrial inspection [1,4,7], logistics automation [2,8,11], precision agriculture [3,9,12,15], and environmental monitoring [5,6,10,13,14,16,17]. In such contexts, robots are required to perceive their surroundings, make decisions, and execute control actions in real time, often in the presence of uncertainty and external disturbances. As a result, one of the central challenges in autonomous robotics lies in the design of control architectures that guarantee safe, accurate, and energy-efficient operation while minimizing reliance on continuous human intervention [18,19].

Among autonomous robotic platforms, unmanned aerial vehicles (UAVs), particularly multirotor configurations, have attracted considerable attention due to their high maneuverability, vertical takeoff and landing capabilities, and adaptability to confined environments. Multirotor UAVs have been widely adopted in both civilian and industrial applications, owing to their mechanical simplicity and flexible control architecture [20,21]. Coaxial octorotors, characterized by eight rotors arranged in coaxial pairs, offer enhanced stability, redundancy, and payload capacity compared to conventional quadrotors. However, these advantages are accompanied by increased system complexity, as octorotors exhibit highly nonlinear, strongly coupled, and underactuated dynamics, which make their modeling and control particularly challenging [22,23].

To address these challenges, advanced control techniques capable of handling nonlinearities, uncertainties, and external disturbances are required. Classical linear controllers, such as PID regulators, often show limited performance when applied to complex aerial systems operating in uncertain environments [2]. This limitation has motivated the development of intelligent and nonlinear control approaches, including fuzzy logic control, sliding mode control, adaptive control, and learning-based methods [24,8,13]. These techniques offer improved robustness and adaptability, making them particularly suitable for autonomous UAV applications where accurate modeling is difficult and operating conditions may vary significantly [7,25].

In this context, the objective of this thesis is to develop intelligent and robust control architectures for a coaxial octorotor, enabling autonomous and reliable operation in complex and uncertain environments. The work focuses on the progressive development of control strategies, beginning with accurate nonlinear dynamic modeling and extending to advanced hybrid and learning-based control approaches. These strategies integrate fuzzy logic control, sliding mode control, adaptive neuro-fuzzy inference systems (ANFIS), and particle swarm optimization (PSO) in order to achieve high-performance trajectory tracking, robust disturbance rejection, and smooth control action. All proposed control architectures are systematically evaluated and validated through extensive simulations carried out in the MATLAB/Simulink environment.

The combination of advanced control and optimization techniques adopted in this thesis provides complementary advantages sunnerized in points that effectively address the challenges associated with the control of a coaxial octorotor operating in complex environments.

- **Fuzzy Logic Control (FLC):** Fuzzy logic controllers offer robustness against modeling uncertainties and external disturbances by emulating human reasoning. They allow effective control without requiring an exact mathematical model, making them particularly suitable for highly nonlinear aerial systems.

- **Sliding Mode Control (SMC):** Sliding mode control ensures strong robustness to parameter variations and external disturbances. Its inherent invariance properties guarantee system stability and tracking performance even under significant uncertainties.

- **Hybrid Fuzzy–Sliding Mode Control (FSMC):** The FSMC combines the robustness of sliding mode control with the smoothness of fuzzy logic. This hybridization significantly reduces chattering effects while preserving robustness, leading to improved control accuracy and actuator-friendly control signals.

- **Adaptive Neuro-Fuzzy Inference System (ANFIS):** ANFIS introduces learning and adaptation capabilities into the control loop. By tuning fuzzy rules and membership functions online or offline, it enhances adaptability to changing system dynamics and environmental conditions.

- **Particle Swarm Optimization (PSO):** PSO provides an efficient and systematic approach for optimizing controller parameters. It accelerates convergence, avoids local minima, and eliminates manual tuning, resulting in improved control performance and repeatability.
- **Hybrid and Modular Architecture:** The integration of these techniques within a unified hybrid architecture allows each method to compensate for the limitations of the others. This modularity enhances scalability, flexibility, and applicability to other multirotor UAV configurations

The control techniques investigated in this thesis are applied to the coaxial octorotor in various configurations and evaluated across diverse task scenarios. heir implementation and evaluation are structured progressively over four chapters, which are organized as follows:

The first chapter is devoted to the dynamic modeling of the coaxial octorotor. It provides an overview of UAV classifications and justifies the choice of an eight-rotor coaxial configuration. A complete nonlinear six-degree-of-freedom (6-DOF) dynamic model is derived using the Newton–Euler formalism. This model constitutes the foundation for the control design and performance evaluation presented in the subsequent chapters [26].

The second chapter addresses the design of fuzzy logic controllers for the coaxial octorotor. After introducing the theoretical background of fuzzy sets and inference systems, Mamdani-type fuzzy controllers are developed for altitude, attitude, and yaw control. The simulation results demonstrate the ability of fuzzy control to effectively manage nonlinear dynamics and uncertainties, achieving satisfactory trajectory tracking and disturbance rejection performance [7,13].

The third chapter presents hybrid control strategies that combine fuzzy logic with sliding mode control in order to exploit the advantages of both approaches. Three hybrid controllers are designed and evaluated: a Fuzzy PD Controller (FPDC), a Sliding Mode PID (SMC-PID), and a Fuzzy Sliding Mode Controller (FSMC). These controllers are tested on demanding trajectory tracking tasks, and a comparative analysis is conducted. The results highlight the superior performance of the FSMC in terms of robustness, accuracy, and smoothness of the control action [27,28].

The fourth chapter proposes an integrated and optimized control architecture that further enhances system performance. Adaptive Neuro-Fuzzy Inference Systems (ANFIS) are employed for the most dynamic attitude control loops, while the FSMC is retained for altitude and yaw regulation. In addition, the parameters of the outer-loop PID position controllers are optimized using Particle Swarm Optimization (PSO). The effectiveness of this hybrid strategy is validated through a realistic inspection scenario, demonstrating improved trajectory tracking accuracy and robustness compared to conventional fuzzy control approaches [10].

In summary, this thesis contributes to the development of intelligent, robust, and optimized control structures for coaxial octorotor UAVs. By combining rigorous dynamic modeling with advanced hybrid and learning-based control techniques, the proposed approaches address key challenges in autonomous aerial robotics and provide a solid basis for future research and real-world deployment.

Chapter I :

Dynamic Modeling of the Coaxial Octorotor

I.1 Introduction

Unmanned Aerial Vehicles (UAVs), particularly multi-rotor platforms like octorotors, have gained significant attention due to their versatility and capability to perform complex tasks across a range of industries, including military, environmental monitoring, and industrial inspections. The octorotor, in particular, offers enhanced stability, increased payload capacity, and better fault tolerance compared to other multi-rotor designs such as quadrotors or hexarotors. However, the complexity of its dynamics, including strong coupling between degrees of freedom and susceptibility to external disturbances, presents unique challenges in control and modeling.

The coaxial octorotors also named coaxial quadrotrs are one of the most used UAVS in many areas such as defense, agriculture and search tasks...Etc [16,17]. These UAVs has a better performances on the external disturbances rejection, rotor failure and heavy loads carrying than the quadrotors. The coaxial octorotors has a similar design to the quadrotor which have four rotors each in the end of the arm but the coaxial octorotors have eight rotors each tow rotors are coaxial and placed in end of arm.

The first part of this chapter will present the different types of the UAVs fixed wing and multi rotors. While the second part will present the full dynamic modeling steps for the coaxial octorotor using the Newton Euler method starting by the configuration, the applied forces, the momentum matrices and the gyroscopic effect till to a six degrees of freedom (6DOF) dynamic model. The model will be designed and evaluated in matlab simulink.

I.2 UAV Classification

I.2.1 UAV Categories:

UAVs (Unmanned Aerial Vehicles) can be broadly classified into two major categories: fixed-wing UAVs and rotary-wing UAVs, each serving different operational purposes based on the application and environment.

- **Fixed-Wing UAVs**

  Fixed-wing UAVs are designed similarly to traditional airplanes, where wings provide lift through forward motion. These UAVs are known for their high endurance and long-range capabilities, making them suitable for missions that require extensive coverage table I.1 presents a different fixed wing UAVs with their flying specifications while figure I.1 shows NASA IKHANA MALE (medium altitude long endurance) fixed wing UAV.

- **Characteristics**

  Fixed-wing UAVs excel in flight endurance and high-speed travel, making them ideal for missions requiring long-range or persistent coverage. However, their limited maneuverability restricts them to larger, open areas [29].

- **Applications**

  Common applications include aerial mapping, surveillance, military reconnaissance, and environmental monitoring. They are typically used in operations requiring long-distance flight, such as over large agricultural fields or across vast territories for border patrol[30,31].

- **Limitations**

  Despite their advantages in endurance, fixed-wing UAVs cannot hover or perform vertical takeoff and landing (VTOL), which limits their use in confined or cluttered environments [32].

Table I.1 Fixed wing UAVs type's specifications [33]

Type	Maximum weight	Maximum range
Small	150 Kg	150 Km
Tactical	600 Kg	150 Km
MALE	1000 Kg	200 Km
HALE	1000 Kg	250 Km
Heavy	2000 Kg	1000 Km
Super heavy	2500 Kg	1500 Km

Figure I.1 NASA IKHANA MALE UAV

- **Rotary-Wing UAVs**

Rotary-wing UAVs, on the other hand, use rotors for lift and thrust, allowing them to hover and perform VTOL. This makes them versatile platforms for tasks that require precision and mobility in constrained spaces.

  - **Characteristics**

    Rotary-wing UAVs, such as quadrotors and octorotors, are known for their ability to hover in place and perform agile maneuvers, especially in tight or cluttered spaces. They are ideally suited for dynamic and precise missions that require close-up observation or navigation[34,35].

  - **Applications**

    Rotary-wing UAVs are widely used for aerial photography, infrastructure inspection, and search-and-rescue operations, where the ability to hover and move slowly is crucial. Their precision and stability allow them to navigate complex environments like urban settings, forests, or disaster areas[30,34].

I.2.2 Multi-Rotor UAVs :

Multi-rotor UAVs, a subclass of rotary-wing UAVs, have become one of the most widely used UAV configurations due to their simplicity, stability, and versatility. These UAVs use multiple rotors for lift, control, and thrust, making them efficient for a range of applications, figure 1.2 shows the different multi-rotor UAVs configuration.

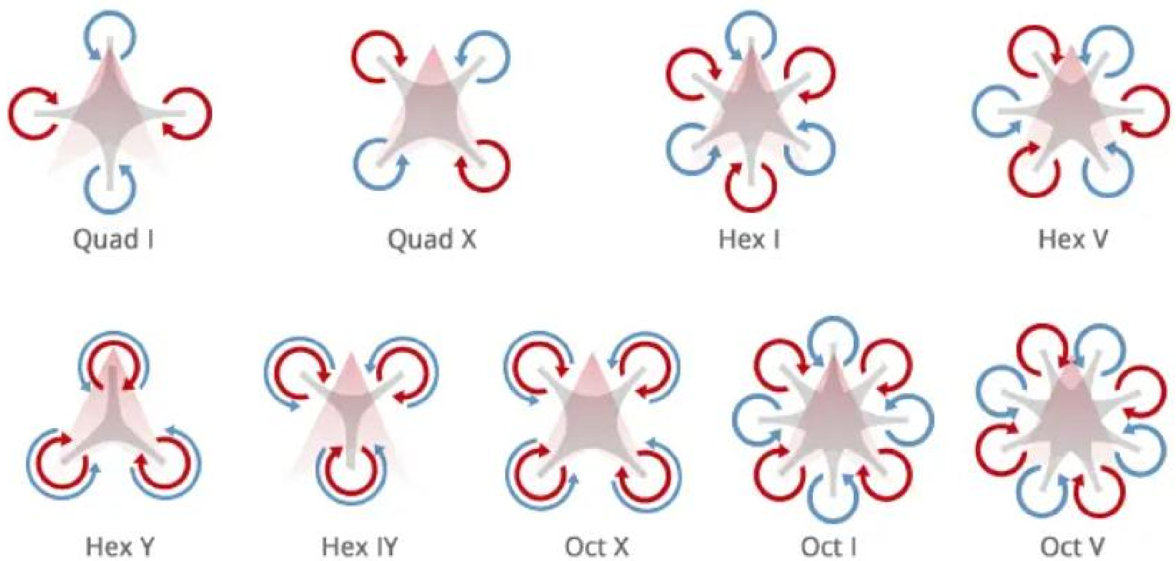


Figure I.2 Different multi-rotor UAVs configurations

- **Overview of Multi-Rotor Designs**

  Multi-rotor UAVs come in different configurations based on the number of rotors:

  - **Bi-rotors**: Simplest design but offers limited control.
  - **Tri-rotors**: Provide better control but are less common.
  - **Quadrotors**: The most popular multi-rotor UAV design, with four rotors, offering excellent stability and control.
  - **Hexarotors**: These six-rotor designs provide more thrust and better stability compared to quadrotors [36,37].
  - **Octorotors**: Feature eight rotors, providing maximum thrust, redundancy, and payload capacity [38,39].

- **Popular Platforms**

  - **Quadrotor**: Quadrotors are favored for their simplicity and ease of control, making them ideal for general applications such as aerial photography, hobbyist flying, and even delivery services [35,40].
  - **Hexarotor**: Hexarotors offer increased payload capacity and greater fault tolerance compared to quadrotors, making them suitable for more demanding tasks such as industrial inspections or carrying specialized sensors [36,41].

I.2.3 Octorotor Design :

Octorotors are the most advanced category within multi-rotor UAVs, providing superior performance in terms of payload capacity, stability, and fault tolerance. Their eight-rotor design offers enhanced control and redundancy, making them critical for applications in military and industrialsectors.

- **Star-ShapedOctorotors**

  In star-shaped octorotor designs, eight independent motors are arranged radially around a central frame.

  - **Configuration**: The rotors are spaced apart on individual arms, providing even lift distribution and robust control. This configuration simplifies maintenance and increases control precision [38,42].

- **Advantages**: Star-shaped octorotors are easy to maintain due to their simple, spread-out design. The distributed lift ensures high stability during flight, making them suitable for general-purpose missions [42].

- **Coaxial Octorotors**

  Coaxial octorotors stack pairs of rotors on the same axis, with one rotor spinning clockwise and the other counterclockwise.

  - **Configuration**: This design minimizes the overall size of the UAV by stacking the rotors vertically, which improves aerodynamic performance while maintaining thrust [43,44].

  - **Advantages**: Coaxial octorotors provide a compact design, making them more maneuverable in constrained environments. Additionally, the counter-rotating rotors cancel out torque, contributing to increased flight stability [44].

- **Advantages of Octorotors**

  Octorotors, especially those with coaxial designs, have several key advantages over other multi-rotor configurations:

  - **Increased Payload Capacity**: With eight rotors, octorotors can lift heavier payloads, making them ideal for missions requiring the transport of heavy equipment or advanced sensor packages [38,43].

  - **Fault Tolerance**: Owing to the redundancy provided by the eight rotors, an octorotor can continue flying even if one or two motors fail, ensuring safer operations in critical environments [43,45].

- **Applications**

  The unique capabilities of octorotors make them indispensable in various fields:

  - **Military Applications**: Octorotors are widely used for surveillance, reconnaissance, and combat support due to their stability and payload capacity [46].

  - **Environmental Monitoring**: These UAVs are used to carry heavy sensor arrays for monitoring environmental changes in remote or hazardous areas [43,44].

- **Industrial Inspections**: Octorotors are utilized for inspecting critical infrastructure, such as power lines, wind turbines, and pipelines, where stability and payload capacity are critical [44,45].

I.3 The coaxial octortor dynamic modelling

The configuration of the coaxial octorotor is similar to that of a quadrotor, using two counter-rotating coaxial motors mounted at the ends of each arm Figure. I.3. This configuration offers improved stability and a more efficient design in size and battery consumption compared to the traditional star-shaped octorotor that requires more than four arms, this configuration achieves the same number of rotors using only four arms, which need to be extended to ensure sufficient spacing between the rotors. This configuration gives six degrees of freedom (6dof) for the coaxial octorotor which movements and rotations in three axes X, Y and Z the rotations called respectively in the aviation standards pitch, roll and yaw [47,48]. These movements are in the body frame [X_b, Y_b, Z_b] which is coincide with the center of gravity (COG) of the coaxial octorotor and to transform these movements to the earth frame [X_e, Y_e, Z_e]. We use the transformation matrix in equation (I.1). Figure I.4 describes the configuration of the coaxial octorotor.

Figure I.3 Coaxial octorotor configuration

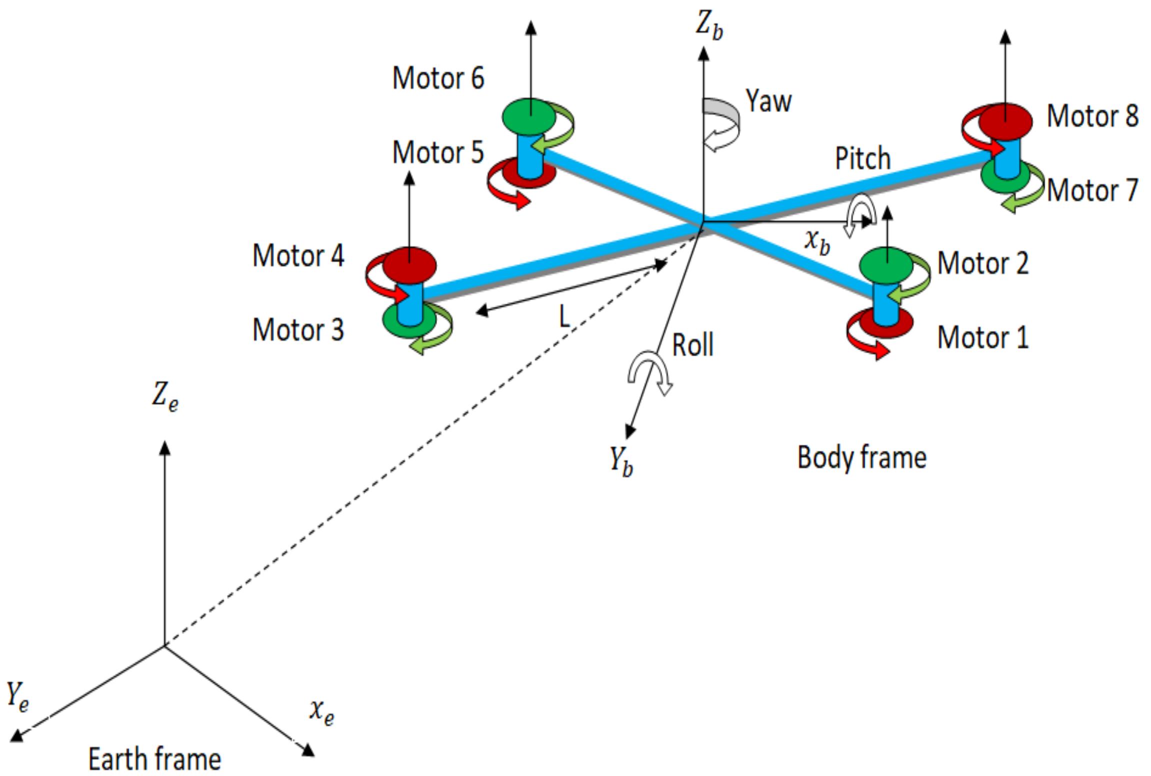


Figure I.4 The coaxial octorotor configuration

$$T = \begin{bmatrix} R & \xi \\ 0 & 1 \end{bmatrix} \tag{I.1}$$

Where **R** is the rotation matrix derived using the flight notations (3, 2, 1) in [47] and ξ is the position vector.

$$\xi = [x \quad y \quad z] \tag{I.2}$$

I.3.1 Coaxial octorotor dynamics

To analyze the motion of the coaxial octorotor, two reference frames are used (see Fig. I.4):

- The **inertial frame** E (X,Y,Z). E which is fixed reference to the ground and considered Galilean,
- And the **body frame** B (x,y,z) which is attached to the coaxial octorotor itself.

To simplify the understanding of the dynamic model presented below, several assumptions are made:

- ✓ The structure of the coaxial octorotor is considered rigid.
- ✓ The center of mass is assumed to coincide with the origin of the body frame.

- ✓ The propellers are assumed to be rigid.
- ✓ The lift and drag forces are proportional to the square of the rotor angular velocity.

- **Rotation matrix :**

We adopt the flight notation (3, 2, 1), which corresponds to rotations about the z, y, and x axes, denoted by ψ, θ, and φ respectively, following aviation standards for yaw, pitch, and roll. For a deeper understanding of Euler angles, refer to [49-51]. To formalize these rotations, we use Euler rotation matrices. Assuming the rotations occur sequentially around the z, y, and x axes transforming from Fe to F1, then F1 to F2, and finally F2 to Fb we construct a rotation matrix for each step. The final rotation matrix is obtained by multiplying these three individual rotation matrices [52,53].

$$R_b^e = R_{F1}^e * R_{F2}^{F1} * R_e^{F2} \tag{I.3}$$

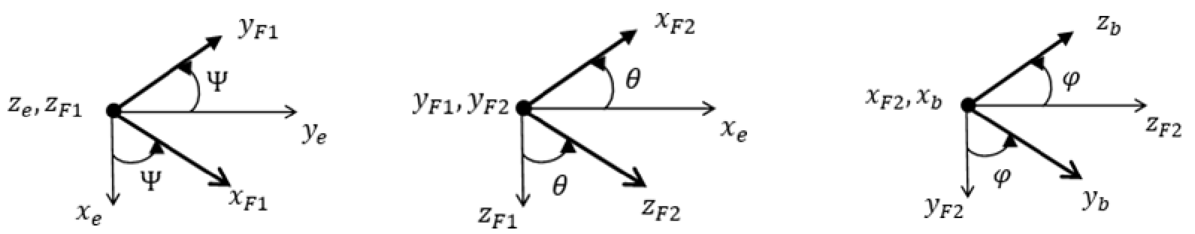


Figure I.5 otation sequences

Using the figure I.5 we can derive the following matrices

$$R_{F1}^e = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{I.4}$$

$$R_{F2}^{F1} = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \tag{I.5}$$

$$R_b^{F2} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & \sin\varphi \\ 0 & -\sin\varphi & \cos\varphi \end{pmatrix} \tag{I.6}$$

Using the formula (I.3) we get the matrices R_b^e which transfers the movements from earth to body frame, we use the inverse matrix to describe movements from body frame to earth frame R_e^b.

$$R_b^e = \begin{pmatrix} C\theta C\psi & C\theta S\psi & -S\theta \\ S\varphi S\theta - C\varphi S\psi & S\varphi S\theta S\psi + C\varphi C\psi & S\varphi C\theta \\ C\varphi S\theta C\psi + S\varphi S\psi & C\varphi S\theta S\psi - S\varphi C\psi & C\varphi C\theta \end{pmatrix} \tag{I.7}$$

$$R_e^b = \begin{pmatrix} C\theta C\psi & S\varphi S\theta S\psi - C\varphi C\psi & C\varphi S\theta C\psi + S\varphi S\psi \\ C\theta S\psi & S\varphi S\theta S\psi + C\varphi C\psi & C\varphi C\theta S\psi - S\varphi S\psi \\ -S\theta & S\varphi C\theta & C\varphi C\theta \end{pmatrix} \tag{I.8}$$

- **Angular velocity :**

The definition of the angular velocity is given by the multiplication of the Euler rates matrixes which is derived using Euler rotation matrixes and the Euler angles rates which is the time derivative of Euler angles vector. Let assume that M is the Euler rates matrix the formula as follows:

$$\Omega = M * \dot{\eta} \tag{I.9}$$

$$\eta = [\varphi \quad \theta \quad \psi]^T \tag{I.10}$$

$$\dot{\eta} = [\dot{\varphi} \quad \dot{\theta} \quad \dot{\psi}]^T \tag{I.11}$$

$$\Omega = [\Omega_x \quad \Omega_y \quad \Omega_z]^T \tag{I.12}$$

Where η, $\dot{\eta}$ and ω are the Euler angles vector, Euler angles rates and the angular velocity. Following the method that mentioned above in Euler rotation matrixes we should define angular velocity for each sequence and the result of the summation of these angular velocities is the angular velocity in F_b.

$$\Omega_b = \Omega_{F1} + \Omega_{F2} + \Omega_b^{F2} \tag{I.13}$$

By transforming ω_{F1} and ω_{F2} to F_b and applying (I.9), we get

$$\Omega_b = \begin{bmatrix} \dot{\varphi} \\ 0 \\ 0 \end{bmatrix} + R_b^{F2^{-1}} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_b^{F2^{-1}} R_{F2}^{F1} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \tag{I.14}$$

$$\Omega_b = \left(1 + R_b^{F2^{-1}} + R_b^{F2^{-1}} R_{F2}^{F1}\right) \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{I.15}$$

Now we have M:

$$M_b^e = \begin{pmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\varphi & \sin\varphi\cos\theta \\ 0 & -\sin\varphi & \cos\theta\cos\varphi \end{pmatrix} \tag{I.16}$$

The inverted version of M derived as:

$$M_e^b = \begin{pmatrix} 1 & \sin\varphi\tan\theta & \cos\varphi\tan\theta \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi\sec\theta & \cos\varphi\sec\theta \end{pmatrix} \tag{I.17}$$

- **Linear velocity :**

The linear velocity in the inertia frame represented by this formula:

$$\nu_e = R \times \nu_b \tag{I.18}$$

- **The forces :**

The coaxial octorotor is subjected to multiple interacting forces that define its motion and stability. Gravity acts constantly on the vehicle's center of mass, pulling it downward. To counter this, each rotor generates lift forces, which are perpendicular to the rotor plane and proportional to the square of the rotor's angular speed, enabling vertical takeoff and hovering. During horizontal translation along the X and Y axes, aerodynamic drag forces oppose motion, while additional rotor drag arises from air resistance on the blades.

- **Gravity forces :**

$$F_g = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \tag{I.19}$$

Where g is the gravity and m the coaxial octorotor mass.

- **Lift Forces :**

It is the perpendicular forces to the rotors which is generated by the rotation of the rotors.

$$F_i = b\omega_i^2 \tag{I.20}$$

Lift forces matrix $\boldsymbol{F_L}$ is:

$$F_L = R \times \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^{8} F_i \end{bmatrix} \tag{I.21}$$

Where $i$the rotor number and b isis the lift coefficient and ω is the angular velocity of the rotor.

- **Drag Forces :**

It is the forces generated when the octorotor is moving in both axis x and y axis which is related to the linear velocity. Also there is a drag force affect the rotor blade which is related to the air density and the blades fabrication.

$$F_r = d\omega^2 \tag{I.22}$$

$$F_v = K_d \nu \tag{I.23}$$

Where d is the drag coefficient; K_d is the translational movement drag coefficient.

- **The moments of the octorotor :**

Are the movements that the octorotor can do which is related to the forces. From figure I.4 we can see that the octorotor has 3 moments which is:

- Pitch moment which produced by the difference between rotors 3, 4 and 7, 8.

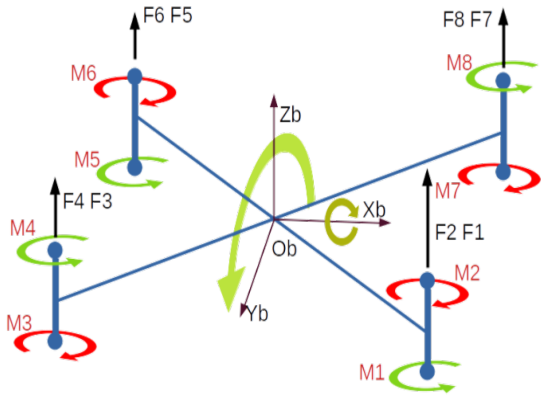


Figure I.5 Pitch moment

- Roll moment which produced by the difference between the rotors 1, 2 and 5, 6.

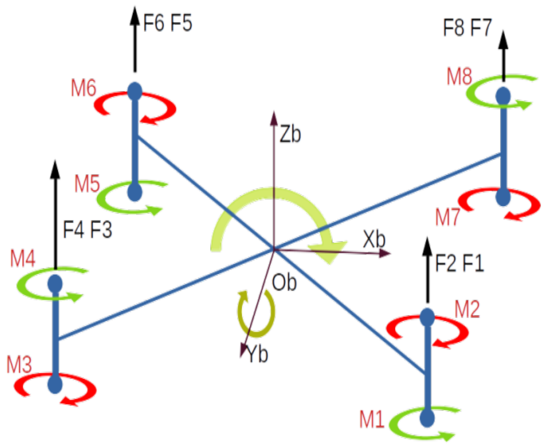


Figure I.6 Roll moment

- Yaw moment which produced by the difference between the coaxial rotors.

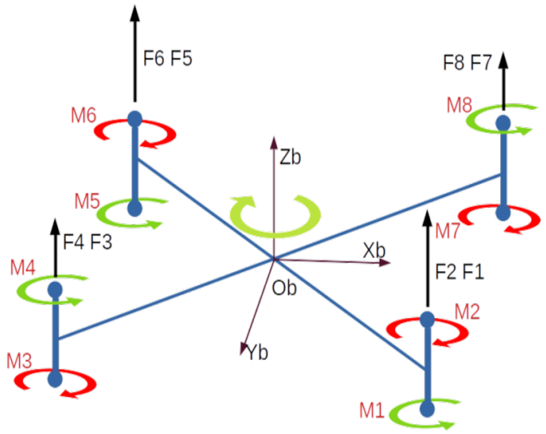


Figure I.7 yaw moment

Now the moment's vector M_L:

$$M_L = \begin{bmatrix} l\frac{\sqrt{2}}{2}(-F_1 - F_2 - F_3 - F_4 + F_5 + F_6 + F_7 + F_8) \\ l\frac{\sqrt{2}}{2}(-F_1 - F_2 + F_3 + F_4 + F_5 + F_6 - F_7 - F_8) \\ d(F2 + F3 + F6 + F7 - F1 - F4 - F5 - F8) \end{bmatrix} \quad (I.24)$$

l is the length between the centre of mass and the rotor.

- **Friction moment :**

$$M_f = \begin{bmatrix} K_4 & 0 & 0 \\ 0 & K_5 & 0 \\ 0 & 0 & K_6 \end{bmatrix} \begin{bmatrix} \varphi^2 \\ \theta^2 \\ \psi^2 \end{bmatrix} \quad (I.25)$$

- **Gyroscopic Effect :**

It's a phenomenon that affects the rotating objects, the gyroscopic effect produces when we apply a force on rotating object. There are two moments of the gyroscopic effect on the octorotor, which is applied to the rotor M_{gr} and the octorotor movement M_{gb}.

$$M_{gr} = \sum_1^8 \Omega \wedge J_r [0 \quad 0 \quad (-1)^{i+1}\omega_i] \quad (I.26)$$

$$M_{gb} = \Omega \wedge J\Omega \quad (I.27)$$

Where J_r and J is the rotor and octorotor inertia, and Ω is the angular velocity.

I.3.2 Dynamic Model Developpment

The dynamic model developed using Newton-Euler formula.The equations is writing as follow:

$$\begin{cases} \dot{\xi} = \nu \\ m\ddot{\xi} = F_g + F_L + F_v \\ J\dot{\Omega} = -M_{gb} + \dot{M}_L - M_{gr} - M_f \end{cases} \quad (I.28)$$

Where m is the octorotor mass, because we supposed that the octorotor is symmetrical the inertia matrix is diagonal.

$$J = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \quad (I.29)$$

In case of small angles, we can write:

$$\Omega = [\dot{\varphi} \quad \dot{\theta} \quad \dot{\psi}]^T \quad (I.30)$$

The drag coefficient matrix that applied to X, Y and Z.

$$F_v = \begin{bmatrix} -K_1 & 0 & 0 \\ 0 & -K_2 & 0 \\ 0 & 0 & -K_3 \end{bmatrix} * \nu \quad (I.31)$$

Now by applying (I.19), (I.21), (I.22) and (I.23), (I.24), (I.25), (I.26), (I.27), (I.29), (I.30), (I.31) in the formula (I.28) we get the final model.

$$\begin{cases} \ddot{x} = u_x - \frac{K_1}{m}\dot{x} \\ \ddot{y} = u_y - \frac{K_2}{m}\dot{y} \\ \ddot{z} = \frac{\cos\varphi\cos\theta}{m}u_1 - \frac{K_3}{m}\dot{z} - g \\ \ddot{\varphi} = \frac{(I_y - I_z)}{I_x}\dot{\theta}\dot{\psi} - \frac{K_4}{I_x}\dot{\varphi}^2 - \frac{J_r\bar{\Omega}}{I_x}\dot{\theta} + \frac{1}{I_x}u_2 \\ \ddot{\theta} = \frac{(I_z - I_x)}{I_y}\dot{\varphi}\dot{\psi} - \frac{K_5}{I_y}\dot{\theta}^2 - \frac{J_r\bar{\Omega}}{I_y}\dot{\varphi} + \frac{1}{I_y}u_3 \\ \ddot{\psi} = \frac{(I_x - I_y)}{I_z}\dot{\theta}\dot{\varphi} - \frac{K_6}{I_z}\dot{\psi}^2 + \frac{1}{I_z}u_4 \end{cases} \quad (I.32)$$

Where u_x, u_y defined as:

$$u_x = \frac{u_1}{m}(\cos\varphi\sin\theta\cos\psi + \sin\varphi\sin\psi) \quad (I.33)$$

$$u_y = \frac{u_1}{m}(\cos\varphi\sin\theta\cos\psi - \sin\varphi\sin\psi) \quad (I.34)$$

$$\bar{\Omega} = [\omega_1 - \omega_2 + \omega_3 - \omega_4 + \omega_5 - \omega_6 + \omega_7 - \omega_8] \quad (I.35)$$

And u_1, u_2, u_3, u_4 represent the moments.

$$\begin{cases} u_1 = (F_1 + F_2 + F_3 + F_4 + F_5 + F_6 + F_7 + F_8) \\ u_2 = c(-F_1 - F_2 - F_3 - F_4 + F_5 + F_6 + F_7 + F_8) \\ u_3 = c(-F_1 - F_2 + F_3 + F_4 + F_5 + F_6 - F_7 - F_8) \\ u_4 = d(-F_1 + F_2 + F_3 - F_4 - F_5 + F_6 + F_7 - F_8) \end{cases} \quad (I.36)$$

These moments are assembled in the vector u. Which considered as system input and the vector y considered as the system output.

$$u = [u_1 \quad u_2 \quad u_3 \quad u_4]^T \quad (I.37)$$

$$y = [x \quad y \quad z \quad \varphi \quad \theta \quad \psi]^T \quad (I.38)$$

The final dynamic model of the octorotor is nonlinear system with high coupling and its control is a challenge.

I.3.3 State Space Model

Now we will rewrite the previous dynamic model in state space form and we chose the state vector $\dot{x}$ as follows:

$$\dot{x} = [x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad \dot{x_6} \quad x_7 \quad \dot{x_8} \quad x_9 \quad \dot{x_{10}} \quad x_{11} \quad x_{12}] \quad (I.39)$$

$$\dot{x} = [x \quad \dot{x} \quad y \quad \dot{y} \quad z \quad \dot{z} \quad \varphi \quad \dot{\varphi} \quad \theta \quad \dot{\theta} \quad \psi \quad \dot{\psi}] \quad (I.40)$$

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = a_1 x_4 x_6 + a_2 x_2^2 + a_3 \bar{\Omega} x_4 + b_1 u_2 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = a_4 x_2 x_6 + a_5 x_4^2 + a_6 \bar{\Omega} x_2 + b_2 u_3 \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = a_7 x_2 x_4 + a_8 x_6^2 + b_3 u_4 \\ \dot{x}_7 = x_8 \\ \dot{x}_8 = a_9 x_8 + \frac{1}{m} u_x u_1 \\ \dot{x}_9 = x_{10} \\ \dot{x}_{10} = a_{10} x_{10} + \frac{1}{m} u_y u_1 \\ \dot{x}_{11} = x_{12} \\ \dot{x}_{12} = a_{11} x_{12} + \frac{\cos\theta \cos\varphi}{m} u_1 - g \end{cases} \quad (I.41)$$

With :

$$\begin{cases} a_1 = \frac{(I_y - I_z)}{I_x}, a_2 = \frac{K_4}{I_x}, a_3 = -\frac{J_r}{I_x}, b_1 = \frac{1}{I_x}, \\ a_4 = \frac{(I_z - I_x)}{I_y}, a_5 = \frac{K_5}{I_y}, a_6 = -\frac{J_r}{I_y}, b_2 = \frac{1}{I_y} \\ a_7 = \frac{(I_x - I_y)}{I_z}, a_8 = \frac{K_6}{I_z}, b_3 = \frac{1}{I_z} \\ a_9 = -\frac{K_1}{m}, a_{10} = -\frac{K_2}{m}, a_{11} = -\frac{K_3}{m} \end{cases} \tag{I.42}$$

I.3.4 Rotor Dynamics

The rotor is a blade rotated by a dc motor. The dc motor is guided using the following dynamics:

$$\begin{cases} V = r.i + L\frac{di}{dt} + K_e.\omega_i \\ K_m = J_r + C_s + K_r.\omega_i{}^2 \\ J\dot{\omega}_i = \tau_m - Q_i - C_s \end{cases} \tag{I.43}$$

From equations in formula (I.43) and by neglecting the inductance of the motor because of its small size we can derive the motor dynamic equation as follow:

$$\dot{\omega} = \frac{K_m}{r.J_r}.V - \frac{K_m.K_e}{r.J_r}.\omega_i - \frac{K_r}{J_r}.\omega_i^2 - \frac{C_s}{J_r} \tag{I.44}$$

I.4 Dynamic model evaluation

We evaluated the dynamic model of the coaxial octorotor in MATLAB/Simulink using the simulation parameters listed in Table I.2. The results are presented in Figures I.8, I.9, and I.10. Figure I.8 shows the pitch, roll, and yaw responses, while Figure I.9 displays the input control signals. Figure I.10 illustrates the full trajectory followed during the simulation.Looking at these figures, we can see that the developed dynamic model is stable and responds well to the input control signals. However, the control signals do suffer from chattering, which affects both performance and trajectory tracking to some extent.

Table I.2 Simulation parameters [55]

b	$2.9842 \times 10^{-5} N.\frac{m}{rad}/s$
d	$2.2320 \times 10^{-7} N.\frac{m}{rad}/s$
K_1, K_2, K_3	$(5.5670, 5.5670, 6.3540) \times 10^{-2} N.\frac{m}{rad}/s$
K_4, K_5, K_6	$(5.5670, 5.5670, 6.3540) \times 10^{-2} N.\frac{m}{rad}/s$
J_r	$2.8385 \times 10^{-5} N.\frac{m}{rad}/s^2$
I_x, I_y, I_z	$(4.2, 4.2, 7.5) \times 10^{-2} N.\frac{m}{rad}/s^2$
m	$1.6\ Kg$
l	$0.23\ m$
K_m	5×10^{-5}
K_r	7×10^{-7}
C_s	2.33×10^{-2}
K_e	3×10^{-5}

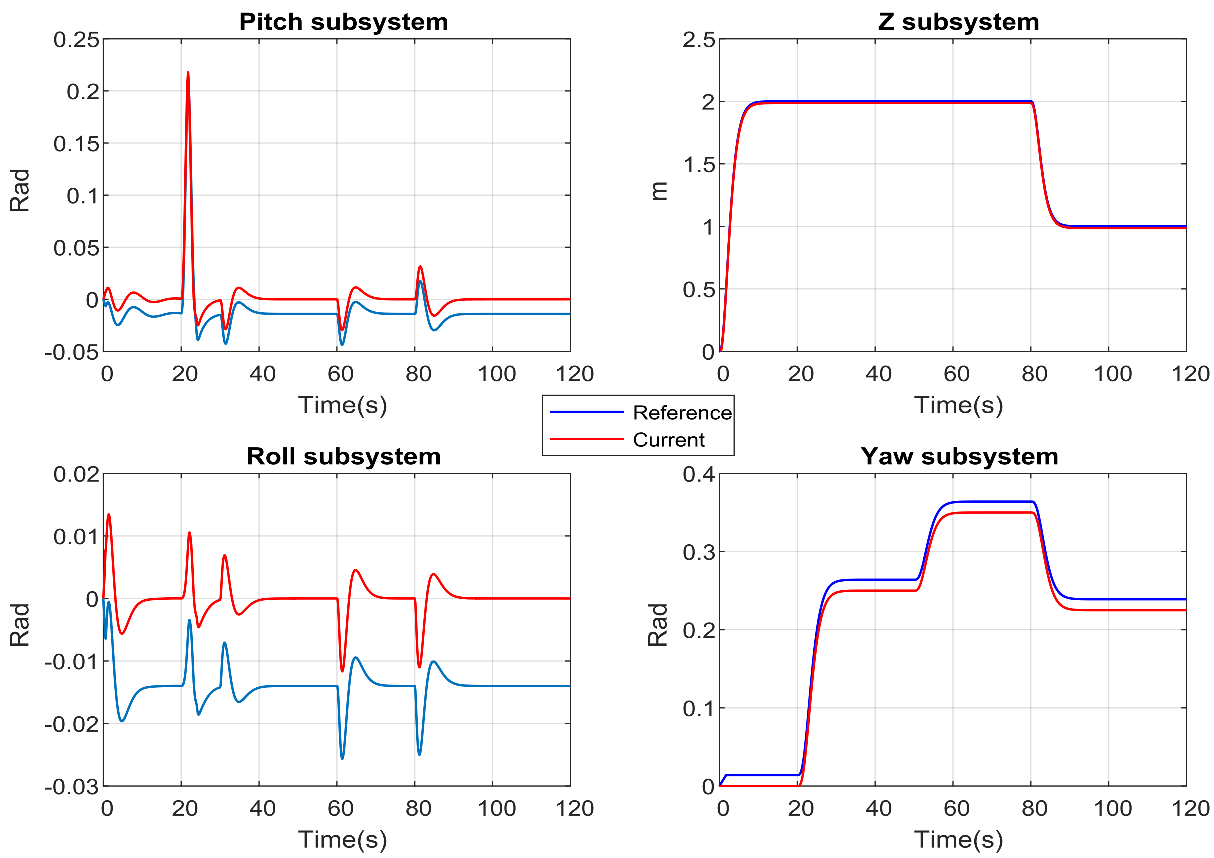


Figure I.8 Angles response

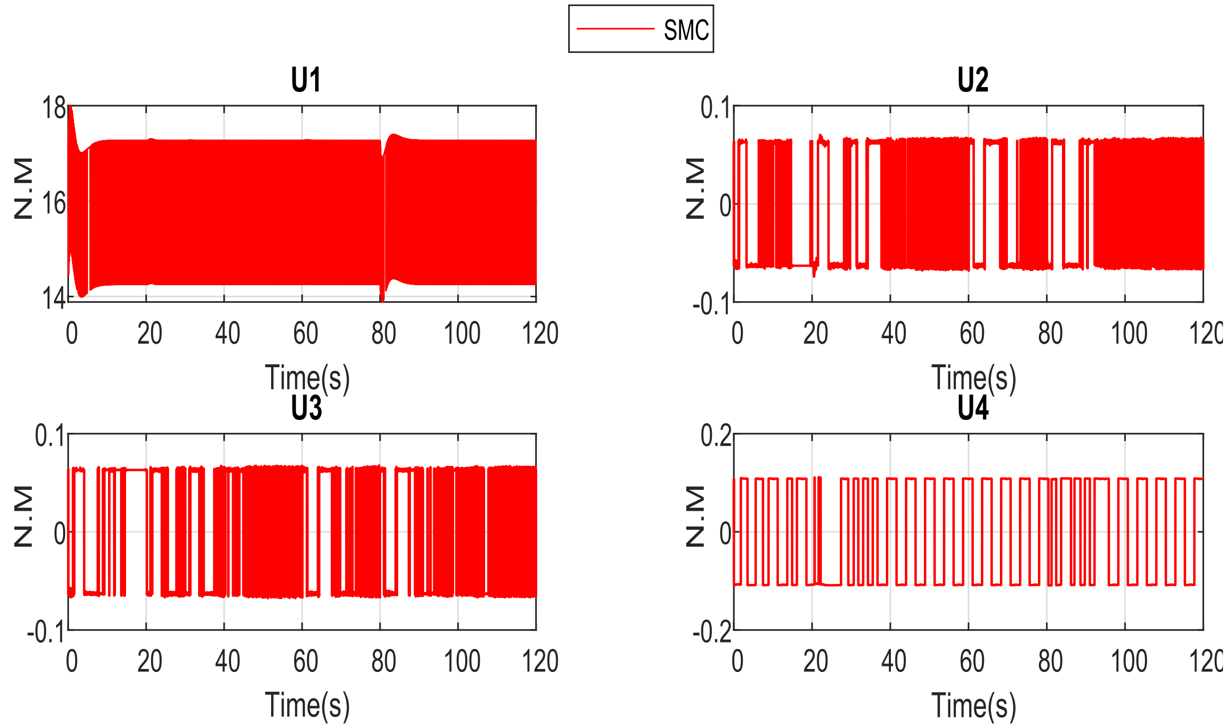


Figure I.9 Control signals

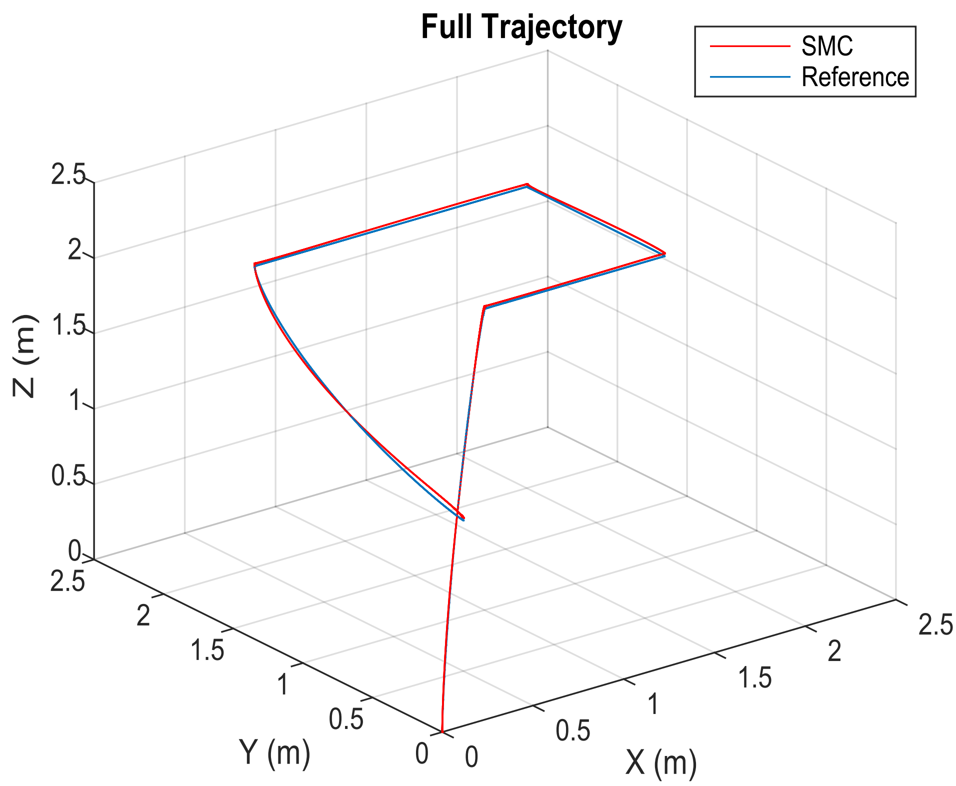


Figure I.10 The full simulation trajectory

I.5 Conclusion

This chapter has provided a comprehensive introduction to Unmanned Aerial Vehicles (UAVs), with a specific focus on multi-rotor systems and the coaxial octorotor configuration. The classification of UAVs into fixed-wing and rotary-wing categories highlighted the distinct advantages and limitations of each type, contextualizing the relevance of multi-rotor UAVsparticularly octorotors for applications requiring stability, payload capacity, and fault tolerance.The dynamic modeling of the coaxial octorotor was developed using the Newton-Euler formalism, accounting for key physical phenomena such as lift and drag forces, gyroscopic effects, rotor dynamics, and aerodynamic friction. A full six-degree-of-freedom (6DOF) nonlinear model was derived, capturing the strong coupling between translational and rotational motions. The model was further expressed in state-space form to facilitate control design and simulation.The evaluation of the dynamic model in MATLAB/Simulink demonstrated its validity and readiness for use in control system development. Simulation results confirmed the expected behavior of the system under typical input signals, showing stable responses in attitude angles (pitch, roll, yaw) and trajectory tracking.This foundational model sets the stage for the design and implementation of advanced control strategies, which will be addressed in subsequent chapters to enhance the performance, robustness, and autonomy of coaxial octorotor systems in real-world applications.

Chapter II :

Fuzzy logic controller Design for the Coaxial Octorotor

II.1 Introduction

Coaxial octorotor UAVs are increasingly being utilized in various fields such as surveillance, defense, agriculture, and delivery services. Due to their diverse application domains, these UAVs exist in multiple configurations, ranging from compact models to larger platforms capable of carrying payloads, cameras, or even weapon systems. Additionally, their design provides enhanced rotor redundancy, which improves reliability. However, these advantages also introduce significant control challenges. The system exhibits high nonlinearity and strong coupling between parameters, making it difficult to achieve stable and precise control using conventional controllers.

Addressing these challenge using the fuzzy logic controllers (FLC) is a well studied topic in many researches as [33] and [55]. The FLC give better performances and esiest application than the conventional methods for this type of dynamic models, the FLC don't need well knowed dynamic model [7] instead need the human expertise described using the linguistic rules and membership functions which make it more suitable for addressing nonlinearities and uncertainties [14].

In this chapter, we provide a brief overview of fuzzy logic, the mathematical foundations of fuzzy set theory, and fuzzy inference systems. We then present the structure of fuzzy controllers that uesd to control the coaxial cotorotor. And in the last we will evaluate the designed controllers in path tracking task under external disturbances.

II.2 Fuzzy logic and Fuzzy sets

The fuzzy logic considred as an extension of bolean logic where he can mebershipe values from 0 to 1 for linguistic variables which are functions of the input or output values. In this section we introduce the fundamental properties of classical sets and fuzzy sets[56,57,58].

II.2.1 Fundamental Concept of a Fuzzy Set :

The main difference between the classical set and fuzzy set presented in figure II.1 where the classical set (Figure II.1(a)) is defined by a characteristic function, denoted as $\boldsymbol{\mu_A(x)}$, such that:

$$U_A(x) = \begin{cases} 1, x \in A \\ 0, x \notin A \end{cases} \tag{II.1}$$

The fundamental concept of fuzzy set theory is the notion of a fuzzy subset. This concept arises from the observation that "very often, the classes of objects encountered in the physical world do not possess well-defined membership criteria."

Let **U** be a collection of objects (for example, $U=R_n$), referred to as the universe of discourse. A fuzzy set **A** in **U** is characterized by a membership function, denoted as $\mu_A(x):U\rightarrow[0,1]$, which assigns to each element $x \in U$ a degree of membership $\mu_A(x)$ indicating the extent to which x belongs to **A** (see Figure II.1(b)). A fuzzy set can thus be regarded as a generalization of a classical set. In contrast, the membership function of a classical set can only take two values: {0, 1}.

$$A = \{x, u_A(x)/x \in U\} \tag{II.2}$$

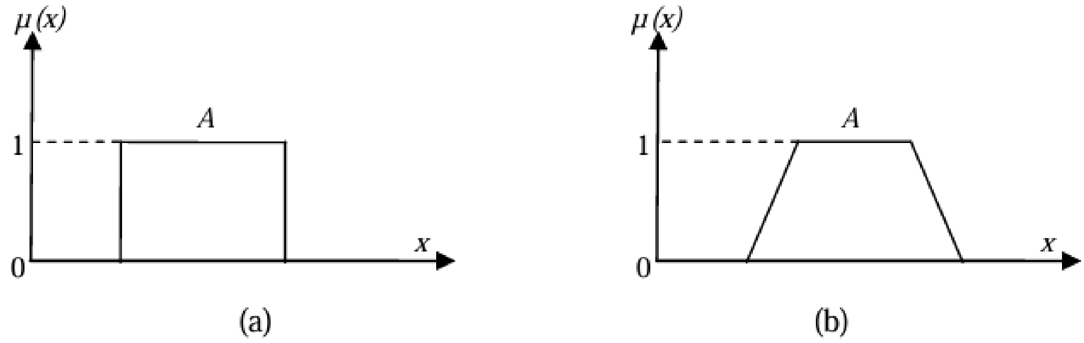


Figure II.1 Examples of membership functions (a) classical logic (b) fuzzy logic

II.2.2 Membershipe function :

A fuzzy set **A** defined on a universe of discourse **U** (which may consist of discrete or continuous elements) is characterized by its membership function. This function specifies the degree of membership of an element **x** to the set **A** [59-61]. Figuer II.1 presents the different types of the membership functions.

- **Triangular :**

The triangular membership function is defined by three parameters {a,b,c}, which determine the coordinates of the three vertices (Figure II.2(a)):

$$u_A(x) = max\left[min\left[\frac{x-a}{b-a}, \frac{c-x}{c-b}\right], 0\right] \tag{II.3}$$

- **Trapezoidal** :

The trapezoidal membership function is defined by four parameters {a,b,c,d},which specify the coordinates of its vertices (Figure II.2(b)):

$$u_A(x) = max\left[min\left[\frac{x-a}{b-a}, 1, \frac{c-x}{c-d}\right], 0\right] \tag{II.4}$$

- **Gaussian :**

The Gaussian membership function is defined by two parameters {m,σ}, representing the mean and standard deviation (Figure II.2(c)):

$$u_A(x) = exp\left\{-\frac{(x-m)^2}{2\sigma^2}\right\} \tag{II.5}$$

- **Sigmoid :**

A sigmoid membership function is defined by two parameters{a,c} (Figure II.2(d)):

$$u_A = \frac{1}{1+exp\{-a(x-c)\}} \tag{II.6}$$

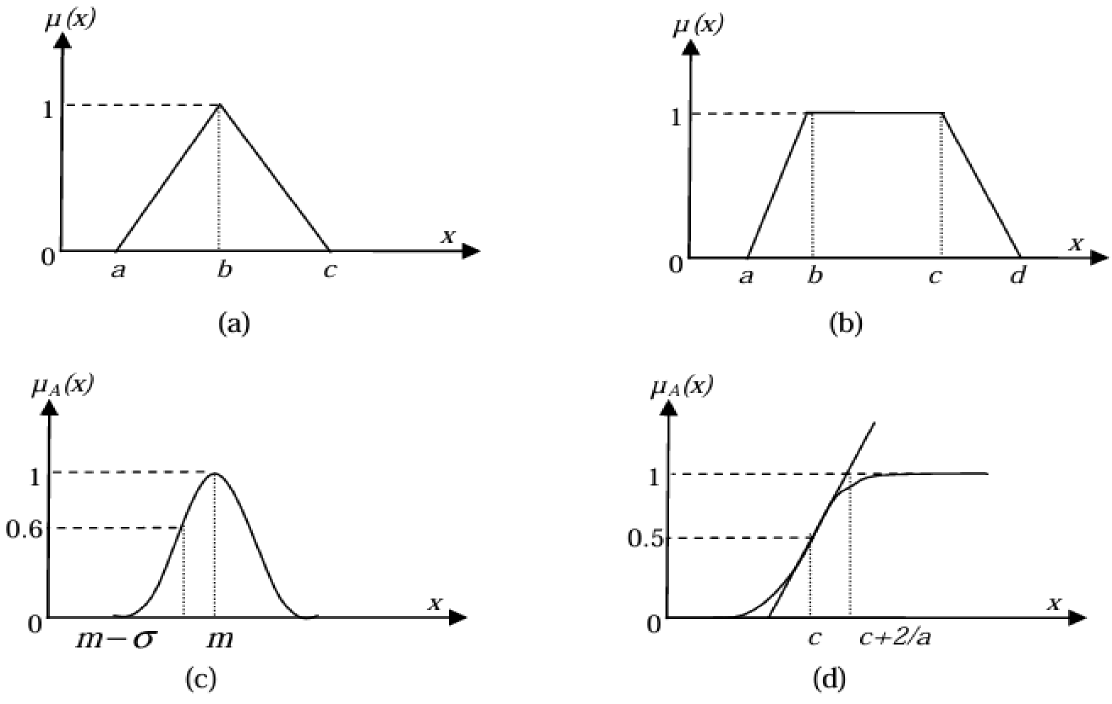


Figure II.2 Membershipe functions examples

- **Linguistic varibales :**

The membershipe functions used to define the linguistic variables which are defined as a set of terms that describe the same concept but at different degrees. This set may be finite or infinite, and each linguistic variable represents a particular state of the system to be controlled [60]. According to fuzzy set theory, a linguistic variable is expressed through linguistic terms such as Small, Medium, Large, and each linguistic term defined using a membership function [61] as seen in figure II.3.

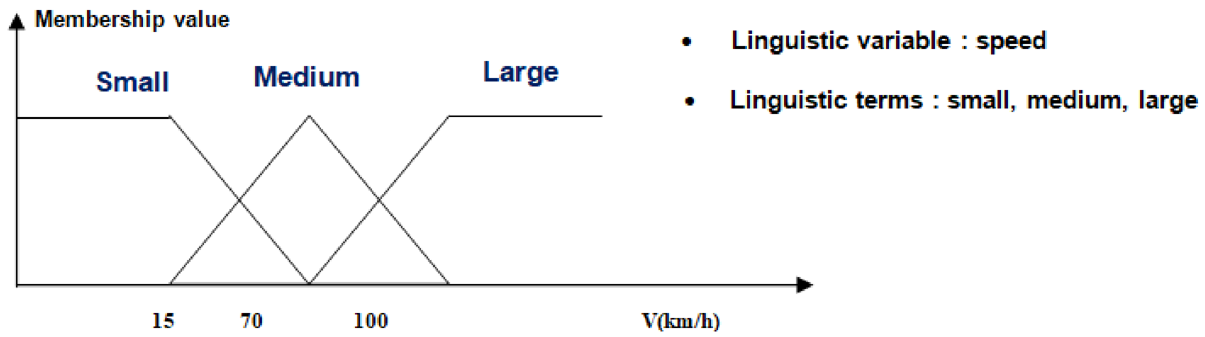


Figure II.3 Linguistic variable example

II.2.3 Rule base :

In general, fuzzy rules are expressed in the form of "IF–THEN" statements. They establish the relationship between input and output variables. A fuzzy rule can thus be written in the following form :

IF x is A **Then** y is B

Fuzzy rules may be simple, with a single antecedent and a single consequent, or compound, involving the conjunction of multiple premises. A compound rule takes the following form:

R: **IF** x1 is A1 and x2 is A2 and . . . and$x_n$isA_n **Then** y is B

Antecedents Conjunction Implication Conclusion

II.3 Fuzzy inference system (FIS)

A Fuzzy Inference System (FIS) aims to transform input data into output data by evaluating a set of rules that link the two. The input variables are obtained through the process of fuzzification, while the rule base is defined using expert knowledge. As illustrated in Figure II.4, the FIS is composed of three main stages: (a) Fuzzification, (b) Inference Engine, and (c) Defuzzification [62].

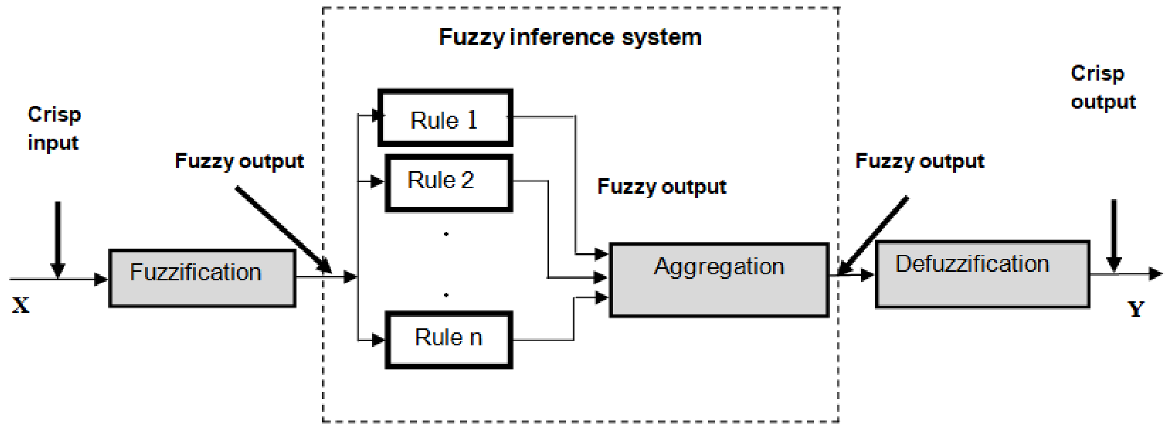


Figure II.4 FIS structure

II.3.1 Fuzzy inference engine

A Mamdani fuzzy rule base is composed of linguistic rules that employ membership functions to describe the underlying concepts. The inference mechanism built upon this structure involves the following steps [61].

- **Fuzzification** :

It evaluates the membership functions used in the predicates of the rules.

Example :**If** “pressure is high” **AND** “time is long” **THEN** “valve opening is large.”

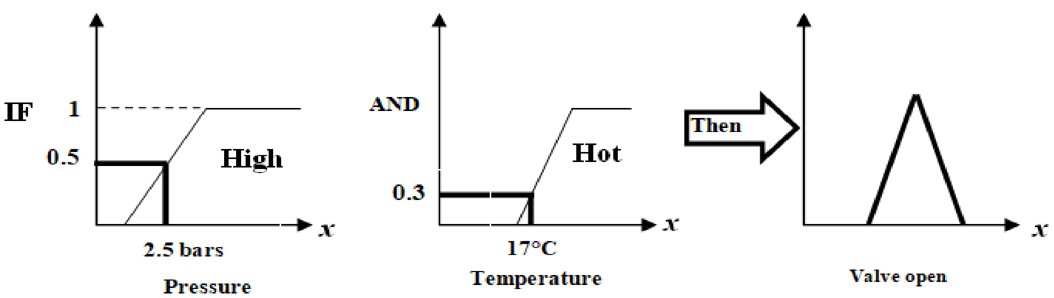


Figure II.5 Fuzzification steps

- **Rule activation degree :**

The degree of activation of a rule corresponds to the evaluation of its antecedent (predicate). It is obtained by applying logical combination operators to the membership values of the propositions contained in the predicate.

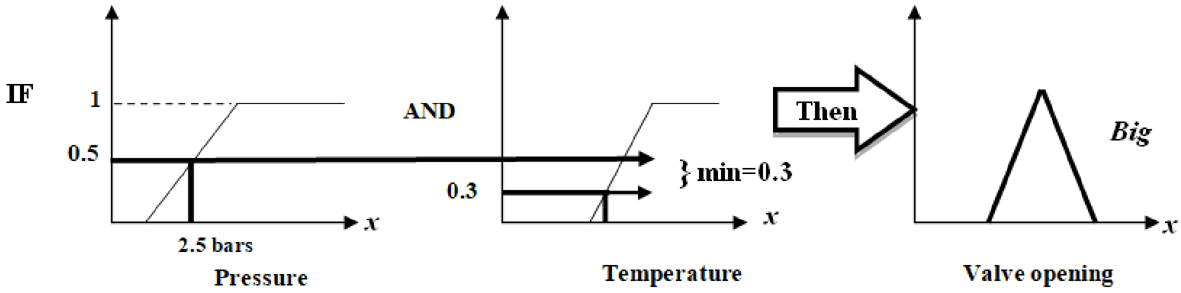


Figure II.6 Rule activation degree

- **Aggregation :**

the overall fuzzy output set is constructed by aggregating the fuzzy sets produced by each rule that contributes to this output. When multiple rules affect the same output variable, their effects are combined using a logical **OR** operator. In practice, this corresponds to taking the **maximum** of the resulting membership functions from each rule:

$$u_B(y) = \max(u_{B1}(y), u_{B2}(y), \dots u_{Bn}(y)) \quad \text{(II.7)}$$

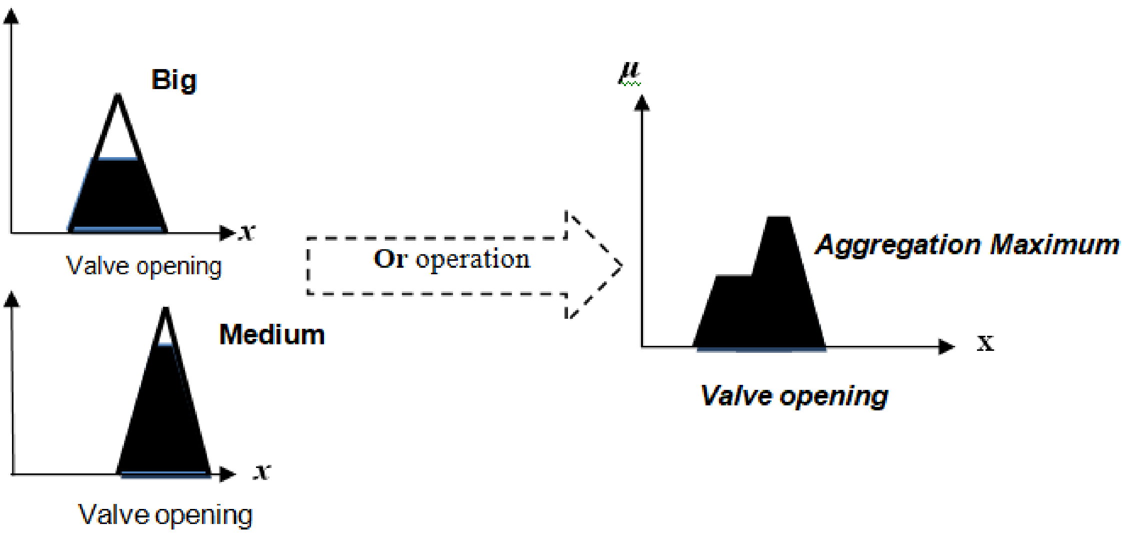


Figure II.7 Rule aggregation

- **Defuzzification :**

Defuzzification is the final stage of the fuzzy inference process, in which the aggregated fuzzy output set is transformed into a single crisp numerical value suitable for system control. Since real-world actuators require precise inputs, this step ensures that the fuzzy reasoning process produces an actionable command. Several defuzzification methods exist, the most common being:

  - **Centroid (center of gravity) :**

The most widely used method, which computes the center of mass of the aggregated fuzzy set:

$$y^* = \frac{\int_Y y\, u_B(y)dy}{\int_Y u_B(y)dy} \tag{II.8}$$

  - **Bisector of area :**

This method finds the value **y∗** such that the area under the aggregated membership function is divided into two equal parts:

$$\int_{y_{min}}^{y^*} u_B(y)dy = \int_{y^*}^{y_{max}} u_B(y)dy \tag{II.9}$$

  - **Mean of Maximum (MOM) :**

Takes the average of all values that reach the maximum membership degree:

$$y^* = \frac{1}{n}\sum_{i=1}^{n} y_i, \qquad weher\ u_B(y_i) = \max_{y\in Y} u_B(y) \tag{II.10}$$

  - **Smallest of Maximum (SOM) :**

Selects the smallest value among those that achieve the maximum membership degree:

$$y^* = \min\left\{y\ \in Y \middle| u_B(y) = \max_{y\in Y} u_B(y)\ \right\} \tag{II.11}$$

  - **Largest of Maximum (LOM) :**

Selects the largest value among those that achieve the maximum membership degree:

$$y^* = \max\left\{y\ \in Y \middle| u_B(y) = \max_{y\in Y} u_B(y)\ \right\} \tag{II.12}$$

II.3.2 Fuzzy inference engine types

There are tow main inference models dominate fuzzy logic applications: the Mamdani model**,** which uses fuzzy sets in both antecedents and consequents and requires defuzzification, and the Takagi–Sugeno–Kang (TSK) model, where the consequents are crisp functions of the inputs, allowing more efficient computation. While Mamdani systems are intuitive and interpretable, TSK systems are favored for real-time control and adaptive applications due to their accuracy and efficiency see table II.1.

Table II.1 Mamdani vs TSK(Takagi-sugeno-kang)

Aspect	Mamdani Model	TSK (Sugeno–Kang) Model
Rule Structure	**IF**x is A1 **AND** … **ANDxnis**An THEN y**isB** (fuzzy set)	**IF** x **is** A1**AND** … **AND**xn **is** An **THEN**y=f(x1,...,xn) (crisp function)
Consequent	Fuzzy set described by membership function	Crisp function (usually linear or constant)
Output Form	Aggregated fuzzy set that requires **defuzzification**	Direct crisp value obtained by weighted average of rules
Defuzzification	Mandatory (e.g., Centroid, MOM, Bisector)	Not necessary (output is already crisp)
Interpretability	High – rules expressed in linguistic terms (e.g., “Temperature is High”)	Lower – consequents expressed as mathematical functions
Computational Cost	Higher – requires fuzzification, aggregation, and defuzzification	Lower – avoids complex defuzzification
Accuracy	Moderate – good for qualitative reasoning	Higher – suitable for optimization, modeling, and adaptive control
Applications	Human-like reasoning, expert systems, decision support	Real-time control, adaptive systems, neuro-fuzzy models, optimization problems

II.4 Type 1 mamdani fuzzy logic controller design

Fuzzy logic-based control techniques provide a robust framework for autonomous UAVs, offering effective means to handle and mitigate motion uncertainties. This section outlines the development process of an advanced control strategy for a coaxial octorotor UAV, employing a fuzzy logic controller.

II.4.1 Control methodology

The octorotor model under study is decomposed into four main subsystems: altitude, pitch, roll, and yaw**.** The altitude and yaw dynamics are each regulated by a dedicated fuzzy logic controller (FLC1, FLC2, FLC3, and FLC4), following the same control structure. In contrast, the pitch and roll subsystems are organized into cascaded loops, where the outer loop governs position while the inner loop regulates the angular variable. To achieve this structure, two additional fuzzy logic controllers are employed for these subsystems (FLC5 and FLC6) [63] see figure II.8.

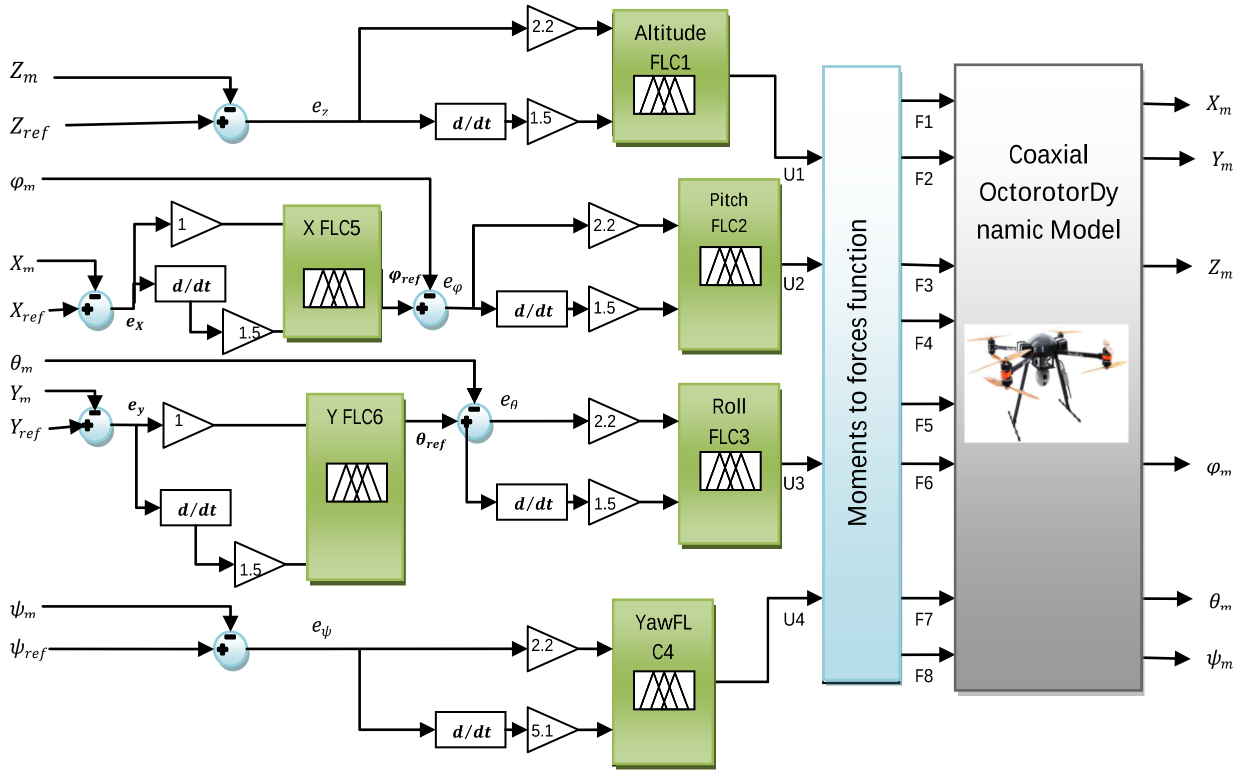


Figure II.8 The control methodology

II.4.2 FLC design

Each subsystem is controlled by a Type-1 Mamdani fuzzy logic controller (FLC) with an identical structure. For every controller, the input variables are defined as the error and its time derivative, while the single output variable corresponds to the control moment generated for the coaxial octorotor [63].

II.4.2.1 Fuzzification

Based on both operator expertise and insights from the literature on fuzzy logic theory, five triangular membership functions are considered sufficient. These functions are applied uniformly to all input and output variables of the fuzzy inference systems with different ranges [63] see figure II.9.

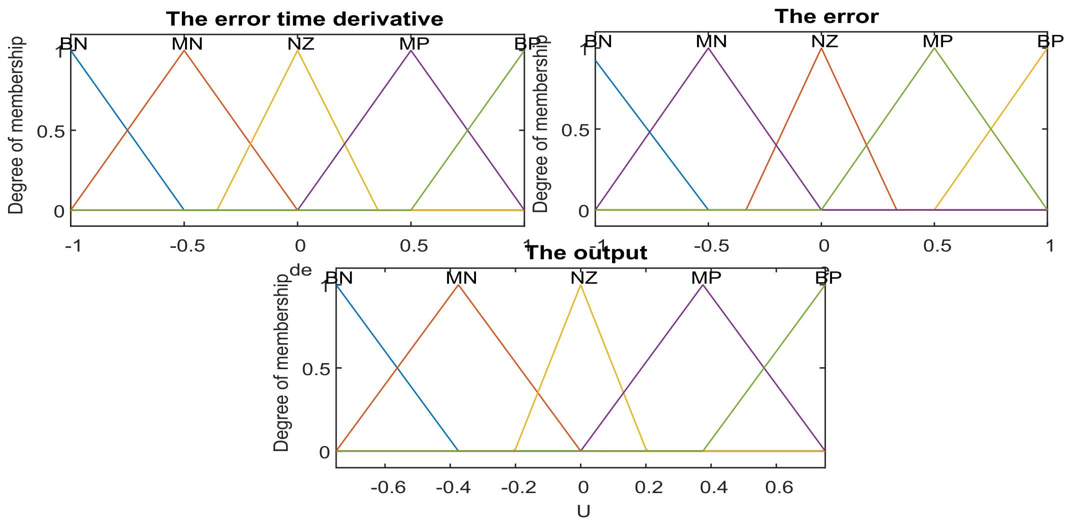


Figure II.9 Input and output variables membership functions

As in figure II.9 the membership functions named (**BN**): big negative, (**MN**): medium negative , (**NZ**): near zero, (**MP**): medium positive and (**BP**): Big positive.

II.4.2.2 Rule base :

To ensure a correct inference mechanism within the fuzzy logic theory, the implication method of the fuzzy rule base is selected according to the operator's expertise and knowledge of the process behavior. At each time step, the controller aims to minimize the error between the measured variables and their corresponding reference set points. In this case study, the adopted fuzzy rule base is formulated as follows [63]:

If e **is** BN **and** de **is** BN, **then** u **is** BN

The membership functions are denoted as AN, BN, and CN, where **e** and **de** represent the input variables and **u** corresponds to the output variable. The complete fuzzy rule base is summarized in Table II.2 and the rule surface in figure II.10.

Table II.2 Rule base

E \ DE	BN	MN	NZ	MP	BP
BN	BN	MN	BP	MN	BN
MN	MN	MN	MN	NZ	MN
NZ	MN	MN	NZ	MP	MP
MP	NZ	NZ	MP	BP	MP
BP	MP	MP	BP	BP	BP

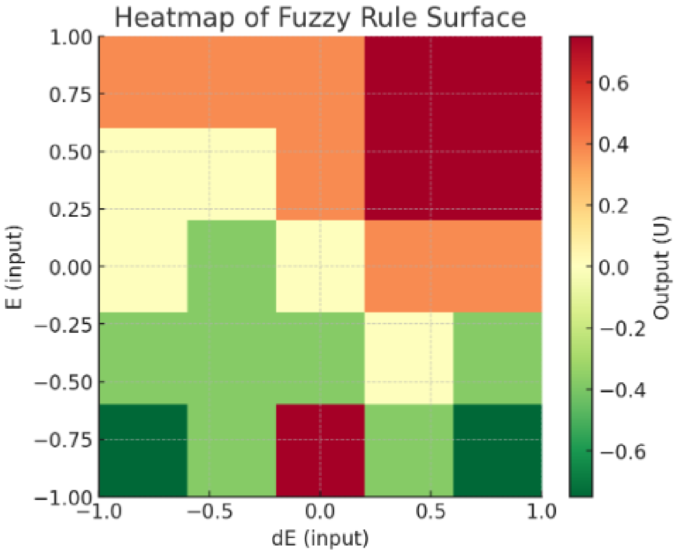


Figure II.10 Rule surface

- **Rule activation :**
  - **AND method (minimum operator):**

In fuzzy logic, the logical conjunction **AND** between two fuzzy sets A and B is most commonly implemented using the minimum operator:

$$\alpha = \min(u_{BN}(de), u_{BN}(e)) \tag{II.13}$$

This ensures that the combined truth degree of the antecedents is bounded by the least confident membership.

  - **Implication (minimum method):**

The implication step maps the degree of truth of the rule's antecedent to its consequent. In Mamdani inference, this is often realized by truncating the output membership function using the minimum operator:

$$u_{\grave{B}N}(u) = \min(\alpha, u_{BN}(u)) \tag{II.14}$$

II.4.2.3 Aggregation

- **Maximum operator :**

When multiple rules contribute to the same output variable, their fuzzy outputs must be combined into a single fuzzy set. The most widely used method is the maximum operator:

$$u_B(u) = \max(u_{BN}(u), u_{MN}(u), u_{NZ}(u), u_{MP}(u), u_{BP}(u)) \tag{II.15}$$

This ensures that the overall fuzzy output reflects the strongest contribution among the activated rules.

II.4.2.4 Defuzzification

- **Center of gravity :**

Defuzzification converts the aggregated fuzzy output into a crisp control signal. The most common approach is the centroid (center of gravity) method, which computes the balance point of the fuzzy set:

$$y^* = \frac{\sum_{i=1}^{n} u_i u_B(u_i)}{\sum_{i=1}^{n} u_B(u_i)} \tag{II.16}$$

II.5 Simulation and results

In this section, we present the results of the simulation of the fuzzy logic controller in matlabsimulink. The results presents the path-tracking performance with the robustness test under externally applied disturbances. After implementing the proposed fuzzy controller on the four subsystems (pitch, roll, position and yaw)the following observations were presented in figures II.11, II.12 and II.13 for the angles subsystems and figures II.14, II.15 for the overall trajectory and the robustness test in the pitch subsystem.The measured (real) responses closely match the desired reference values, demonstrating a high degree of tracking accuracy across all variables. The tracking error remains near zero, which confirms the effectiveness of the fuzzy controllers in regulating the pitch, roll, and yaw angles. This highlights the capability of the proposed control strategy to accurately guide the UAV along the prescribed reference trajectory. Furthermore, the position responses of the coaxial octorotor (X,Y,Z) relative to their corresponding reference trajectories (X_d,Y_d,Z_d) are illustrated as two three-dimensional paths in figure II.14.

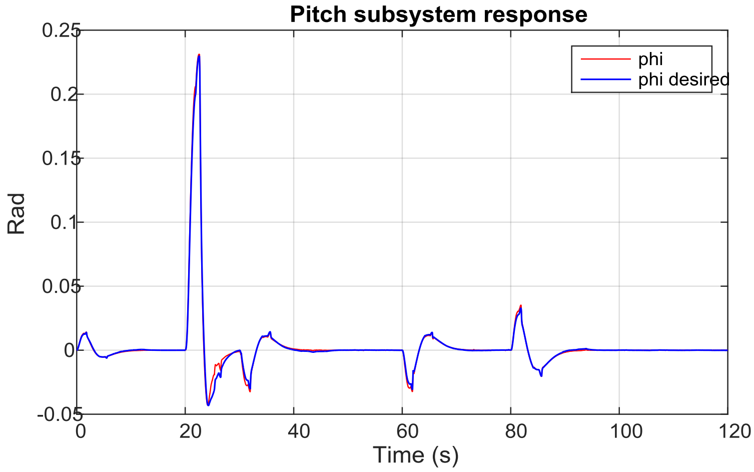


Figure II.11 Pitch subsystem response

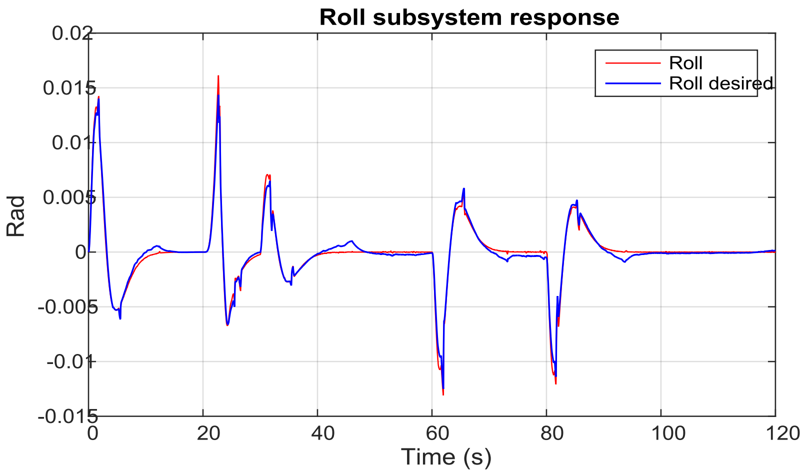


Figure II.12 Roll subsystem response

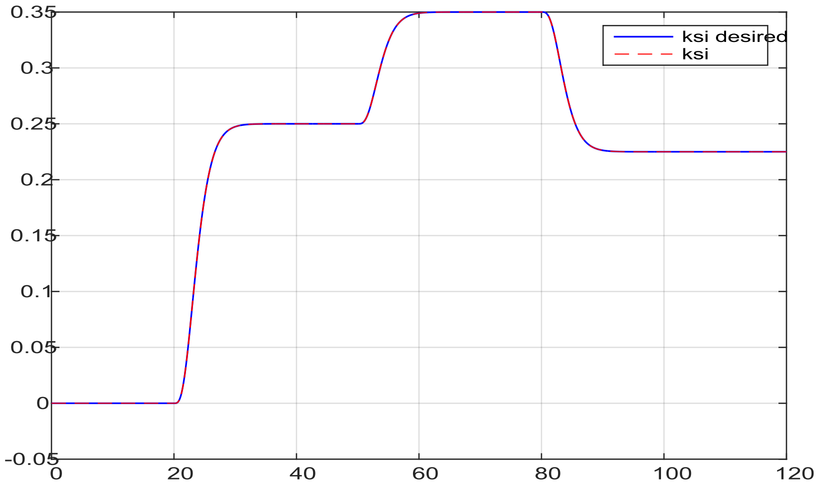


Figure II.13 Yaw subsystem response

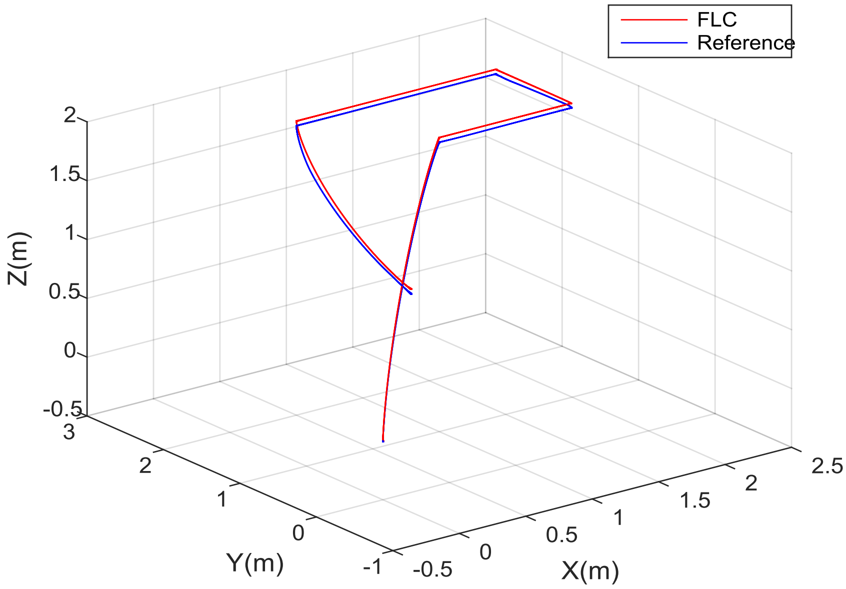


Figure II.14 The full trajectory

Figure II.15 shows the variation of the X-position under external disturbances applied in the pitch angle subsystem, evaluating the robustness of the proposed control strategy using fuzzy logic controller. The disturbance was modeled as a step signal corresponding to a 0.2 rad disturbance applied for a duration of 2 seconds at the time instant t=40s. As shown in the illustrated in figure II.15, the control system successfully compensates for the disturbance, guiding the UAV back to its reference trajectory with minimal delay. The system achieves recovery within approximately 5 seconds, thereby demonstrating the octorotor's capability to maintain stable flight and achieve accurate tracking performance despite external perturbations.

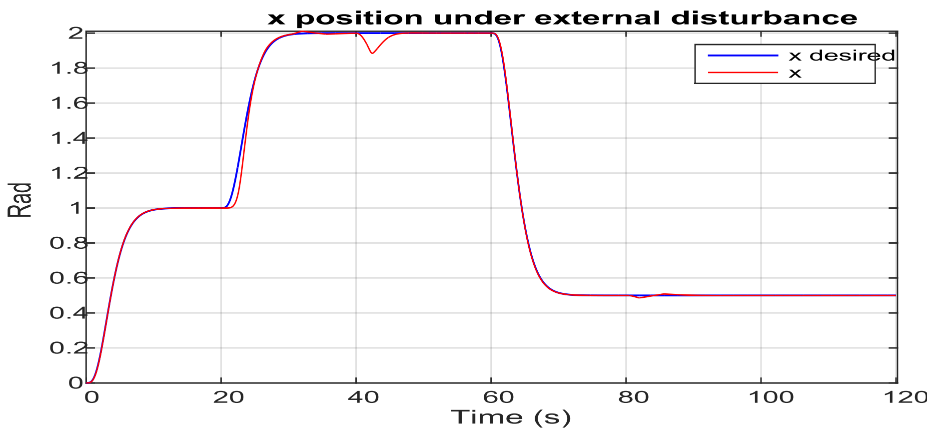


Figure II.15 Robustness test

II.6 Conclusion

In this chapter, the design and implementation of a Type-1 Mamdani fuzzy logic controller for the coaxial octorotor UAV were presented. A theoretical overview of fuzzy set theory and fuzzy inference systems was first introduced, including fuzzification, rule base construction, aggregation, and defuzzification methods. Both Mamdani and Takagi–Sugeno–Kang (TSK) inference models were discussed, highlighting their differences in terms of interpretability, computational complexity, and applicability to real-time control.

The proposed control methodology was then applied to the four main subsystems of the octorotoraltitude, pitch, roll, and yaw. Each subsystem was controlled by a fuzzy logic controller with two input variables (error and error derivative) and one output variable (control moment). Five triangular membership functions were adopted for both inputs and outputs, and the fuzzy rule base was defined using operator expertise and system behavior. The inference mechanism was based on the minimum operator for implication and the maximum operator for aggregation, while defuzzification was performed using the centroid method.

Simulation results demonstrated the effectiveness of the proposed fuzzy controllers in ensuring accurate path tracking and robust performance under external disturbances. The measured responses closely followed the reference trajectories with minimal error, and the system exhibited rapid recovery from disturbances applied to the pitch subsystem, regaining stability within a few seconds. These findings validate the ability of fuzzy logic controllers to handle the nonlinearities and strong couplings inherent in the octorotor dynamics, while providing smooth and reliable control.

This chapter establishes the fuzzy logic controller as a powerful approach for the control of coaxial octorotor UAVs, laying the foundation for further enhancements, such as hybrid intelligent controllers or adaptive fuzzy-neural approaches, which will be explored in subsequent chapters.

Chapter III :

Hybrid Control Strategies for the Coaxial Octorotor

III.1 Introduction

Unmanned Aerial Vehicles (UAVs) have become indispensable platforms in a wide range of civilian and industrial applications, including aerial surveillance, mapping, environmental monitoring, and logistics operations. Among various UAV configurations, multi-rotor systems, and particularly coaxial octorotors, offer remarkable advantages in terms of stability, payload capacity, and maneuverability, thanks to their symmetrical thrust distribution and compact geometry. Despite these benefits, their control remains a complex task due to nonlinear dynamics, actuator coupling, and external disturbance sensitivity [64]

Classical linear controllers such as PID schemes can achieve acceptable performance in near-hover or low-disturbance conditions but often fail under aggressive maneuvers or parameter variations [22,23]. To address these challenges, nonlinear control strategies such as Sliding Mode Control (SMC) and Fuzzy Logic Control (FLC) have gained increasing attention in UAV research. Recent studies show that SMC provides excellent robustness against model uncertainties and external perturbations but suffers from high-frequency chattering [65]. Conversely, FLC offers smooth and adaptive control but may lose performance if not properly tuned for nonlinear dynamics [66].

To exploit the strengths of both paradigms, researchers have proposed hybrid fuzzy–sliding mode control structures that combine the robustness of SMC with the adaptability of fuzzy inference. For instance, [67] developed an Adaptive Neuro-Fuzzy Inference System integrated with SMC to enhance trajectory tracking under external disturbances. Similarly, [68] introduced a global fast terminal fuzzy sliding-mode controller using a neural-network-based structure to improve convergence speed and reduce control chattering. In related work, [69] applied a fuzzy-logic-based adaptive control strategy to a coaxial octorotor UAV, demonstrating improved robustness and flight stability compared to conventional methods.

Building on these advances, this research focuses on the design and comparative analysis of three fuzzy-based nonlinear controllers for a six-degree-of-freedom (6-DOF) coaxial octorotor UAV. The investigated controllers are the Fuzzy PD Controller (FPDC), the Sliding-Mode PID (SMC-PID), and the Fuzzy Sliding Mode Controller (FSMC). A complete nonlinear dynamic model of the octorotor is employed to simulate and evaluate their performance in altitude and attitude control during rectangular and circular trajectory tracking scenarios.

The rest of this chapter is organized into several main sections. The second section presents the fundamental principles of Sliding Mode Control (SMC), providing the theoretical background necessary for understanding its application to UAV systems. The third section describes the design procedure of the SMC-PID controller, followed by the forth section, which details the development of the Fuzzy PD Controller (FPDC). The fivth section focuses on the design methodology of the Fuzzy Sliding Mode Controller (FSMC), emphasizing the integration of fuzzy inference with the sliding mode framework. The sixth section outlines the simulation scenarios, including trajectory-tracking experiments and external disturbance tests, followed by a comprehensive results analysis and comparative evaluation using standard error performance indices. Finally, the last section summarizes the key findings and presents the concluding remarks of the chapter.

III.2 Sliding mode controller

Sliding Mode Control (SMC) is a nonlinear, discontinuous control strategy that belongs to the family of Variable Structure Control Systems (VSCS). It was pioneered in the Soviet Union in the 1950s and formally developed by Utkin in the 1970s [64]. Its core strength lies in robustness: once the system states are confined to a sliding surface, the resulting closed-loop dynamics are invariant to matched disturbances and model uncertainties [65].

The principle of SMC is to design a control law that forces the system state trajectory onto a pre-defined manifold (the sliding surface) and maintains it there despite uncertainties. This yields a two-phase behavior:

1. Reaching phase – the trajectory is driven toward the sliding surface.

2. Sliding phase – the trajectory evolves along the surface according to reduced-order dynamics.

III.2.1 Fundamental of SMC

Consider a nonlinear dynamic system:

$$\dot{x}(t) = f(x,t) + B(x,t)u(t) + d(t) \tag{III.1}$$

where:

- $x(t) \in \mathbb{R}^n$ is the state vector,

- f(x,t) is the known nominal dynamics,

- B(x,t) is the input distribution matrix,

- u(t) is the control input,

- d(t) is the matched disturbance.

The objective of SMC is to design u(t) such that the trajectory reaches the sliding surface in finite time and remains on it thereafter.

III 2.2 Sliding surface

The sliding surface is a manifold in state space that encodes the desired closed-loop dynamics. For regulation or tracking problems, a common choice is:

$$s(x,t) = \dot{e}(t) + \lambda\, e(t)| \tag{III.2}$$

where $e(t) = x_d(t) - x_m(t)$ is the tracking error and $\lambda > 0$ is a tuning parameter.

- When s=0, the system error follows:

$$\dot{e}(t) + \lambda\, e(t) = 0 \tag{III.3}$$

ensuringexponential convergence to zero.

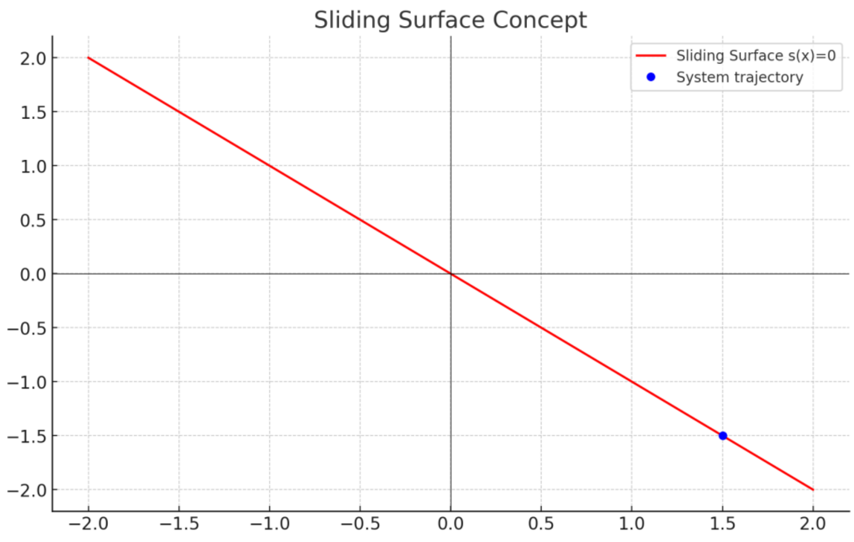


Figure III.1 Sliding surface diagram

III.2.3 Reaching surface and control law

The control law must guarantee finite-time convergence of trajectories to the sliding surface.

A Lyapunov candidate function is chosen:

$$V(s) = 1/2\ s^2 \tag{III.4}$$

Its derivative:

$$\tilde{V}(s) = s\ \dot{s} \tag{III.5}$$

To ensure stability, the sliding condition is imposed:

$$s\ \dot{s} \leq -\eta\ |s|\ \ with \eta > 0 \tag{III.6}$$

A typical discontinuous control law is:

$$u(t) = u_{eq}(t) - K.sign(s) \tag{III.7}$$

- u_{eq}: equivalent control, ensuring motion along the surface.

- K.sign(s): discontinuous term, forcing convergence to the surface.

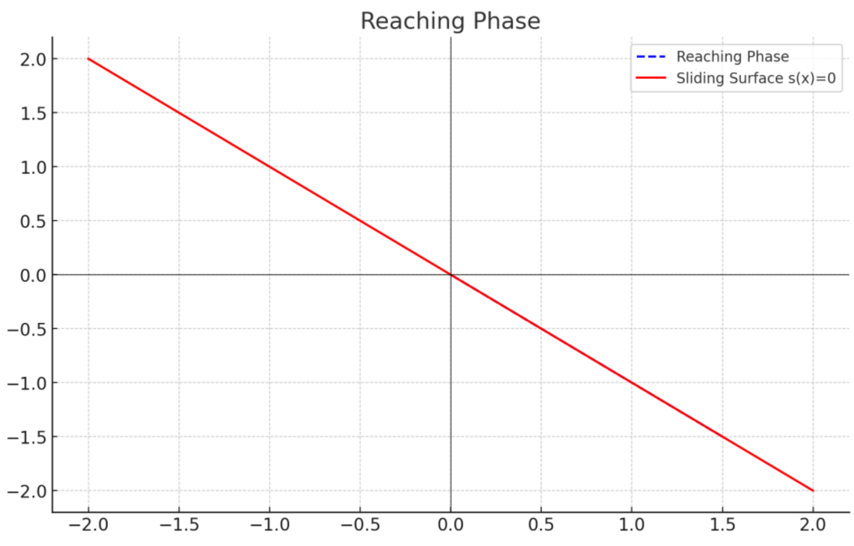


Figure III.2 Reaching surface diagram

III.2.4 Sliding dynamics

Once the system trajectory is constrained to s=0, the dynamics reduce to:

$$\dot{x}(t) = f(x,t) + B(x,t)u_{eq}(t) \tag{III.8}$$

The disturbance d(t) is canceled, yielding robustness against matched uncertainties [66].

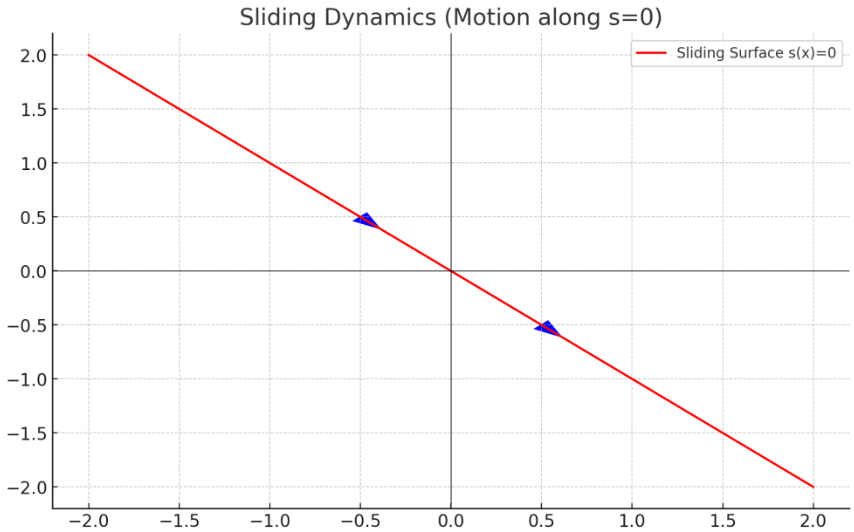


Figure III.3 Sliding dynamics

III.2.5 Chattering effect

In practice, ideal discontinuous control is impossible due to actuator bandwidth limits and unmodeled dynamics. This leads to chattering high-frequency oscillations around the sliding surface [67].

- **<u>Consequences:</u>**
  - Increased wear in actuators,
  - Excitation of unmodeled high-frequency dynamics,
  - Reduced system efficiency.
- **<u>Mitigation strategies:</u>**
  - Boundary layer method: replacing sign(s) with sat(s/φ),
  - Higher-Order Sliding Modes (HOSM),
  - Adaptive and observer-based methods [68].

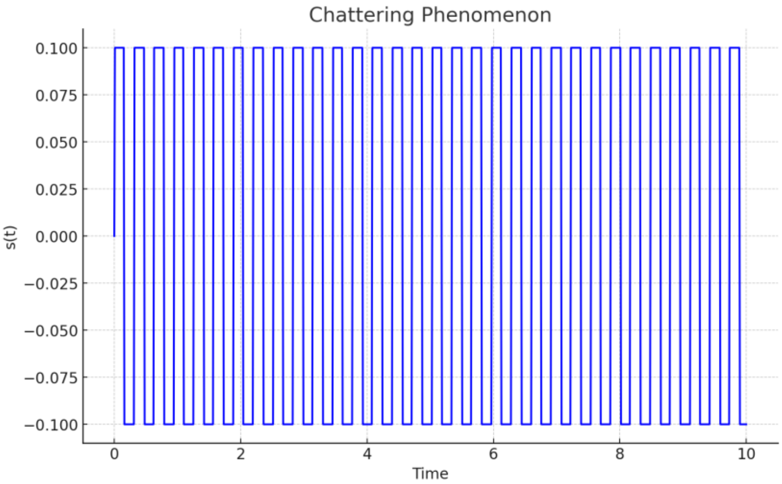


Figure III.4 Chattering effect

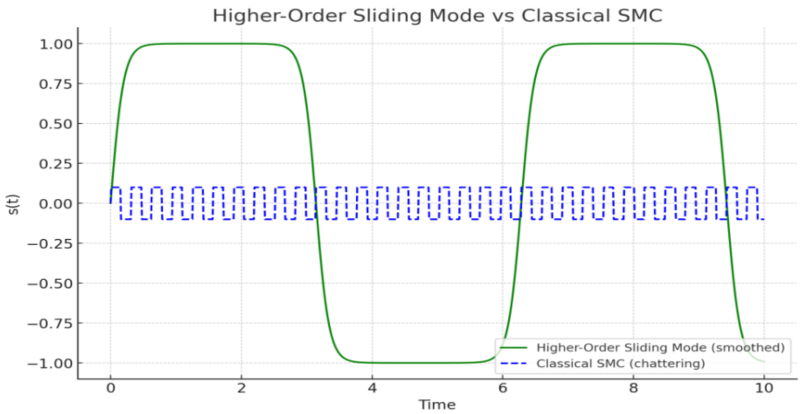


Figure III.5 Classical SMC vs higher order SMC

III.3 Sliding Mode-PID controller (SMC-PID)

A first-order sliding mode controller is applied to the octorotor subsystems: altitude, pitch, roll, and yaw. Whereas, for steering x and y subsystems, conventional PID controllers are used to produce the desired pitch and roll. The equations (III.9) to (III.22) provide a step-by-step description of the SMC development for each subsystem of the coaxial octorotor. The proposed control structure that uses the SMC-PID to steer the coaxial octorotor autonomously to accomplish the path-tracking task is presented in Figure III.6.

III.3.1 Defining the errors and sliding surfaces for each controller

Desired states: (x_d,y_d,z_d,ϕ_d,θ_d,ψ_d)

Current states : (x,y,z,ϕ,θ,ψ)

Sliding mode control parameters: (λ,K,C)

- **The error for each state:**

$$e_x=x_d-x_m,\ \ e_y=y_d-y_m,\ \ e_z=z_d-z_m \tag{|III.9}$$

$$e_\varphi=\varphi_d-\varphi_m,\ \ e_\theta=\theta_d-\theta_m,\ \ e_\psi=\psi_d-\psi_m \tag{III.10}$$

- **The sliding surface for each subsystem:**

$$S_Z=\dot{e}_Z+\lambda_Z e_Z\ ,\ S_\phi=\dot{e}_\phi+\lambda_\phi e_\phi \tag{III.11}$$

$$S_\theta=\dot{e}_\theta+\lambda_\theta e_\theta\ ,\ S_\psi=\dot{e}_\psi+\lambda_\psi e_\psi \tag{III.12}$$

III.3.2 Deriving the controller equation for each subsystem

- **Altitude control (U1) :**

- Dynamics:

$$\ddot{z}=\frac{\cos\varphi\cos\theta}{m}u_1-\frac{K_3}{m}\dot{z}-g \tag{III.13}$$

- Control input:

$$U_1=\frac{1}{\cos\varphi.\cos\theta}[K_3+mg+\ddot{z}_d.m+\dot{e}.c.m+sgn(s).K] \tag{III.14}$$

- **Pitch control (U2) :**

- Dynamics:

$$\ddot{\varphi}=\frac{(I_y-I_z)}{I_x}\dot{\theta}\dot{\psi}-\frac{K_4}{I_x}\dot{\varphi}^2-\frac{J_r\bar{\Omega}}{I_y}\dot{\varphi}+\frac{1}{I_x}u_2 \tag{|III.15}$$

- Control input :

$$U_2=I_x\left[\dot{e}_\varphi.c+\ddot{\varphi}_d-\left(\dot{\theta}.\dot{\psi}.A_1-\dot{\varphi}^2.A_2-A_3.\dot{\varphi}\right)+sgn(s).K\right] \tag{III.16}$$

Where

$$A_2=\frac{K_4}{I_x}\,A_1=\frac{(I_y-I_z)}{I_x}\,A_3=\frac{J_r\bar{\Omega}}{I_y} \tag{III.17}$$

- **Roll control (**U3**):**

- Dynamics:

$$\ddot{\theta}=\frac{(I_z-I_x)}{I_y}\dot{\varphi}\dot{\psi}-\frac{K_5}{I_y}\dot{\theta}^2-\frac{J_r\bar{\Omega}}{I_x}\dot{\theta}+\frac{1}{I_y}u_3 \tag{III.18}$$

- Control input:

$$U_3=I_y\left[\dot{e}_\theta.c+\ddot{\theta}_d-\left(\dot{\varphi}.\dot{\psi}.A_5-\dot{\theta}^2.A_6-A_7.\dot{\theta}\right)+sgn(s).K\right] \tag{III.19}$$

Where

$$A_6=\frac{K_5}{I_y}\ A_5=\frac{(I_z-I_x)}{I_y}\ A_7=\frac{J_r\bar{\Omega}}{I_x} \tag{III.20}$$

- **Yaw control (**U4**):**

- Dynamics:

$$\ddot{\psi}=\frac{(I_x-I_y)}{I_z}\dot{\theta}\dot{\varphi}-\frac{K_6}{I_z}\dot{\psi}^2+\frac{1}{I_z}u_4 \tag{III.21}$$

- Control input:

$$U_4=I_z\left[\dot{e}_\psi.c+\ddot{\psi}_d-\left(\dot{\varphi}.\dot{\theta}.A_9-\dot{\psi}^2.A_8\right)+sgn(s).K\right] \tag{III.22}$$

Where

$$A_8=\frac{K_6}{I_z}\ A_9=\frac{(I_x-I_y)}{I_z} \tag{III.23}$$

III.3.3 PID controllers for X and Y subsystems to generate θd and ϕd:

- **X position controller (Desired pitch)**

$$\theta_d=K_p.e_y+K_i.\int e_y+K_d.\frac{de_y}{dt} \tag{III.24}$$

- **Y position controller (Desired roll)**

$$\varphi_d=K_p.e_x+K_i.\int e_x+K_d.\frac{de_x}{dt} \tag{III.25}$$

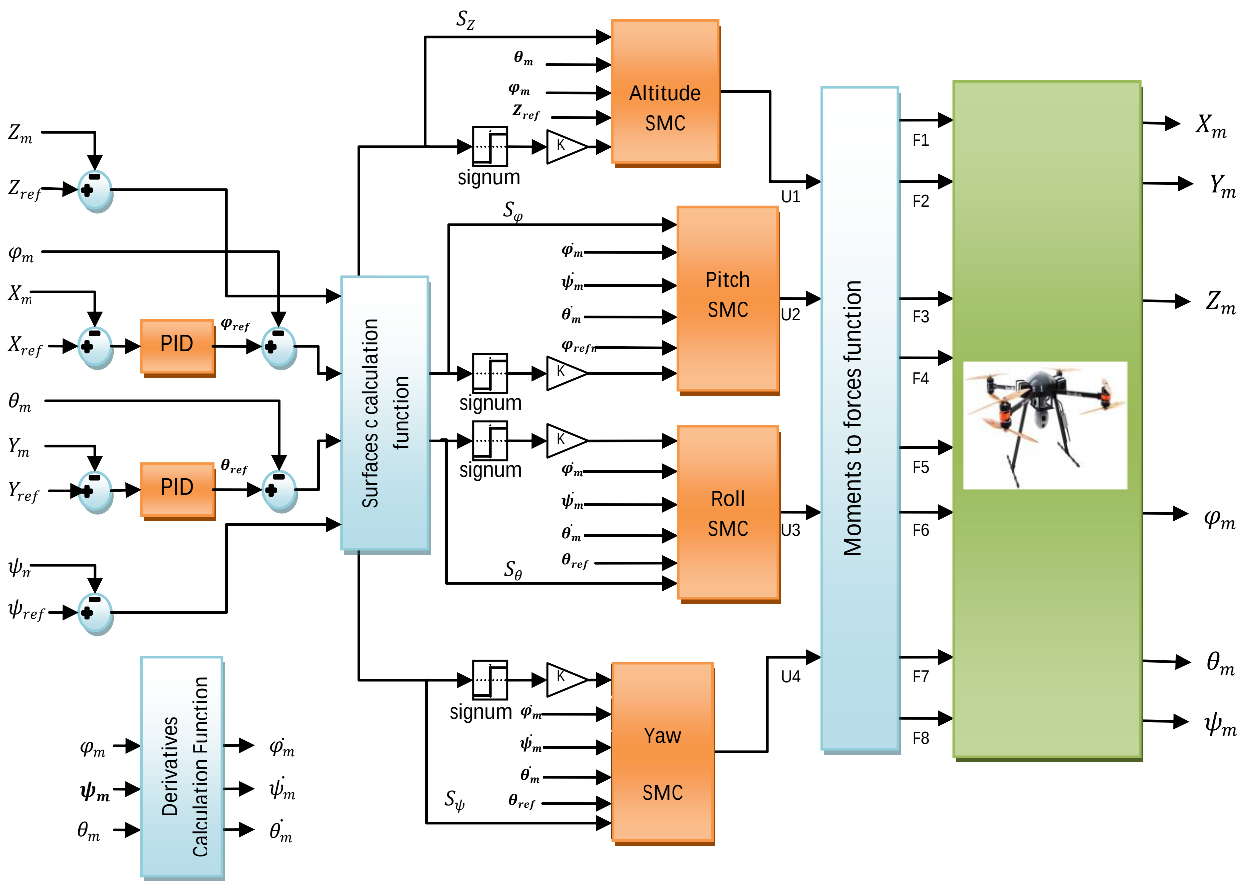


Figure III.6 SMC-PID control structure for the path tracking coaxial octorotor

III.4 Fuzzy Logic PD Controller (FPDC)

The second control approach proposed in this study is based on the Fuzzy Logic Proportional Derivative Controller (FPDC), which utilizes fuzzy inference systems to enhance the control performance of the coaxial octorotor. Fuzzy logic control has gained considerable attention in nonlinear control applications due to its ability to handle system uncertainties and approximate human reasoning in decision-making processes [69,70].In this control strategy, the altitude, yaw, roll and pitch subsystems are controlled by identical fuzzy logic controllers (FLCs), while the X and Y position subsystems are controlled using classical proportional–derivative (PD) controllers that generate the desired pitch and roll references. These pitch and roll references are then processed by two additional fuzzy logic controllers responsible for the pitch and roll dynamics. Consequently, four fuzzy controllers denoted as FLC1 (altitude), FLC2 (pitch), FLC3 (roll), and FLC4 (yaw) are implemented to achieve stable and precise control of the octorotor's main dynamic loops. The overall control strategy shon in figure III.7.

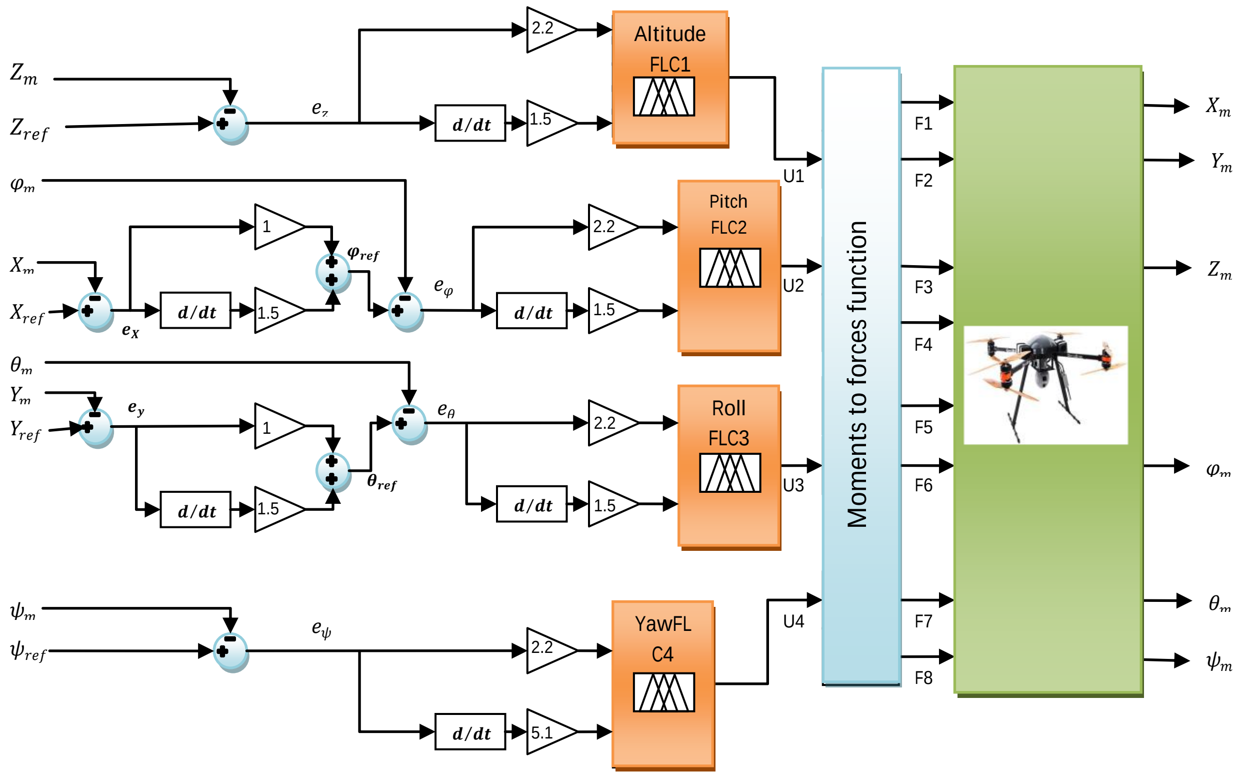


Figure III.7 FPD Ccontrol structure for the coaxial octorotor

III.4.1 Fuzzy logic controller design

III.4.1.1 Input and output variables

Each fuzzy logic controller is designed with two input variables and one output variable. The first input variable is the error (**e**), representing the deviation between the reference and actual values. It is characterized by five trapezoidal membership functions: Negative Big (**NB**), Negative (**N**), Near Zero (**NZ**), Positive (**P**), and Positive Big (**PB**), as shown in Figure III.8. The second input, the error derivative (de), captures the rate of change of the error and is described by three Gaussian membership functions: Negative (**N**), Zero (**Z**), and Positive (**P**), illustrated in Figure III.9.

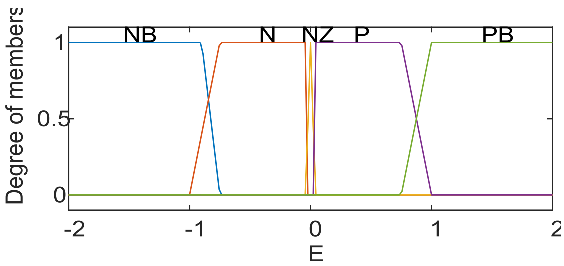


Figure III.8 The error input variable

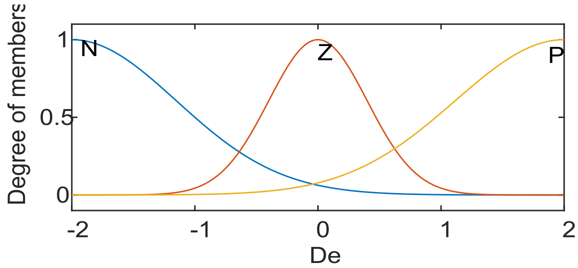


Figure III.9 The de input variable

The output variable corresponds to the controller’s corrective action and is defined using five membership functions: Go Down Much (**GDM**), Go Down (**GD**), Stay (**S**), Go Up (**GU**), and Go Up Much (**GUM**), as presented in Figure III.10.

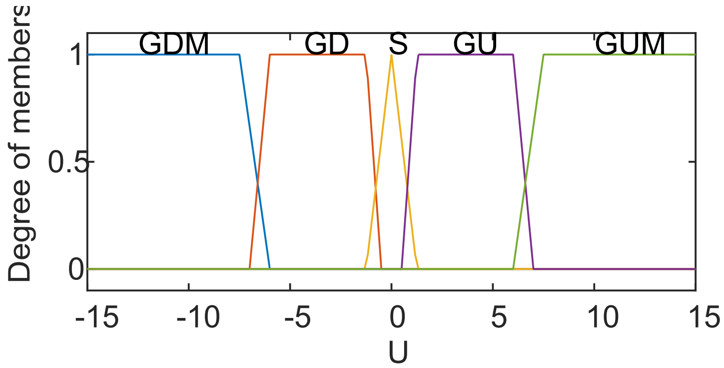


Figure III.10 The output variable membership functions

III.4.1.2 Rule base and defuzzification method

The rules chosed apropriętly to approximate the humen expertise in the control of the coaxial octorotor and the reules fired as this formula:

IF e is NB and de is N then U is GDM

The table III.1 present the rules used in the fuzzy logic controller according to the (**If and then**) formula.

Table III.1 Fuzzy Rules

de \ e	NB	N	NZ	P	PB
N	GDM	GD	GD	S	GU
Z	GUM	GD	S	GU	GUM
P	GD	S	GU	GUM	GUM

III.4.1.3 The defuzzification

The defuzzification of the output is performed using the centroid method defined by the eqation (III.26), which provides a smooth and balanced crisp output, ensuring stable system behavior [71,72].

$$Y^* = \frac{\int_X \mu_u(u)udu}{\int_X \mu_u(u)du} \tag{III.26}$$

Where $\mu_u(u)$ is the aggregated membership function of the output variable, X is the univers of discourse and Y^* is the crisp output.

III.4.2 PD controllers design

The FPDC structure integrates both fuzzy logic and classical control principles. The PD controllersare utilized for the position (X and Y) subsystems, generating desired pitch (φ_{ref}) and roll (θ_{ref}) angles. These reference angles are subsequently processed by the fuzzy controllers (FLC2 and FLC3), which adjust the actuator signals to maintain flight stability. The PD controllers used for the *X* and *Y* are deifned in the equations III.27 and III.28.

$$\theta_d = K_p.e_y + K_d.\frac{de_y}{dt} \tag{III.27}$$

$$\varphi_d = K_p.e_x + K_d.\frac{de_x}{dt} \tag{III.28}$$

III.5 Fuzzy sliding mode controller (FSMC)

In this section, a hybrid control approach combining Sliding Mode Control (SMC) and Fuzzy Logic Control (FLC) is developed for the coaxial octorotor. The purpose of this integration is to exploit the robustness of sliding mode control while mitigating its primary drawback the chattering phenomenonthrough the adaptive and smoothing capabilities of fuzzy logic.

Sliding Mode Control (SMC) is widely recognized for its robustness against modeling uncertainties and external disturbances. However, the high-frequency switching characteristic of SMC can excite unmodeled system dynamics and cause undesirable oscillations in the actuators [73,74]. To address this limitation, fuzzy logic is introduced to generate the switching signal in a continuous and adaptive manner, effectively reducing chattering while preserving system robustness [75,76].

III.5.1 The fuzzy sliding mode controller design (FSMC)

III.5.1.1 The fuzzy sliding mode concept

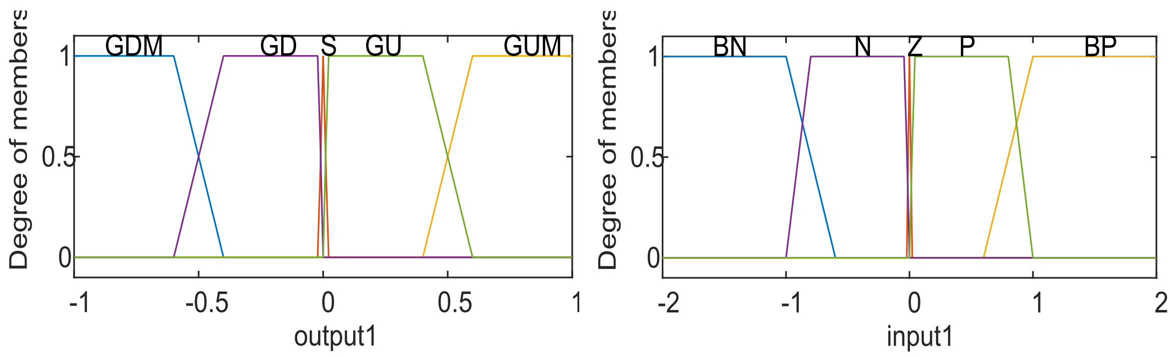
The design steps of the sliding controller is the same as the sliding controller design steps defined by the equtions III.9 to III.22. for that in this subsection we will consider on the fuzzy sliding mode concept.

The proposed controller is based on a first-order sliding mode control law, integrated with a Type-1 Mamdani fuzzy logic controller. The fuzzy system dynamically adjusts the switching function according to the instantaneous system states. The fuzzy logic controller includes two sets of membership functions:

Input variable (sliding surface, Si): represented by five trapezoidal membership functions Big Negative (BN), Negative (N), Zero (Z), Positive (P), and Big Positive (BP) as shown in Figure III.11(a).

Output variable (control action, U): defined by five trapezoidal membership functions Go Down Much (GDM), Go Down (GD), Stay (S), Go Up (GU), and Go Up Much (GUM) depicted in Figure III.11(b).

The fuzzy rule base governing the relationship between Sx and U is summarized in Table III.2. These rules are formulated to ensure smooth switching behavior near the sliding surface, thereby reducing discontinuities in the control law.



(a) (b)

Figure III.11 Input and output variables membership function

Table III.2 Fuzzy logic rule-base

S_x	BN	N	Z	P	BP
Output	GDM	GD	S	GU	GUM

III.5.1.2 Fuzzy based switching generator

To reduce chattering and enhance adaptability, the discontinuous switching term of conventional SMC is replaced by the fuzzy inference system (FIS)**.** The fuzzy logic controller outputs a continuous gain that modulates the switching effort depending on the magnitude of Si. The fuzzy switching signalfor each subsystem can be expressed as:

- **Altitude**

$$U_{SZ}=K_f(S_Z).K \tag{III.29}$$

- **Pitch**

$$U_{S\varphi}=K_f(S_\varphi).K \tag{III.30}$$

- **Roll**

$$U_{S\theta}=K_f(S_\theta).K \tag{III.31}$$

- **Yaw**

$$U_{S\psi}=K_f(S_\psi).K \tag{III.32}$$

Where the total control signal for each controller writon as :

- **Altitude**

$$U_1=U_{eq}+U_{Sz} \tag{III.33}$$

- **Pitch**

$$U_2=U_{eq}+U_{S\varphi} \tag{III.34}$$

- **Roll**

$$U_3=U_{eq}+U_{S\theta} \tag{III.35}$$

- **Yaw**

$$U_4=U_{eq}+U_{S\psi} \tag{III.36}$$

III.5.1.3 Controller equation for each subsystem

Substituting the equations III.28, III.29, III.30 and III.31 of the switching control action in the total control signal equations III.32, III.33, III.34 and III.36 produces the final control equations for each subsystem respectively as:

- **Altitude subsystem:**

$$U_1=\frac{1}{\cos\varphi.\cos\theta}\left[K_3+mg+\ddot{z}_d.m+\dot{e}.c.m+U_{SZ}\right] \tag{III.37}$$

- **Pitch subsystem:**

$$U_2=I_x\left[\dot{e}_\varphi.c+\ddot{\varphi}_d-\left(\dot{\theta}.\dot{\psi}.A_1-\dot{\varphi}^2.A_2\right)+U_{S\varphi}\right] \tag{III.38}$$

- **Roll subsystem:**

$$U_3=I_y\left[\dot{e}_\theta.c+\ddot{\theta}_d-\left(\dot{\varphi}.\dot{\psi}.A_5-\dot{\theta}^2.A_6\right)+U_{S\theta}\right] \tag{III.39}$$

- **Yaw subsystem:**

$$U_4=I_z\left[\dot{e}_\psi.c+\ddot{\psi}_d-\left(\dot{\varphi}.\dot{\theta}.A_9-\dot{\psi}^2.A_{10}\right)+U_{S\psi}\right] \tag{III.40}$$

The **X** and **Y** position subsystems are controlled using a classical PID controller, which generates the desired pitch and roll references. These references are subsequently used by the FSMC to adjust the attitude and maintain spatial stability. The PID controllers defined in the equations III.23 and III.24.

The overall control architecture is illustrated in figure III.12, showing the interaction between the fuzzy logic units, the SMC layers, and the multirotor dynamics of the coaxial octorotor.

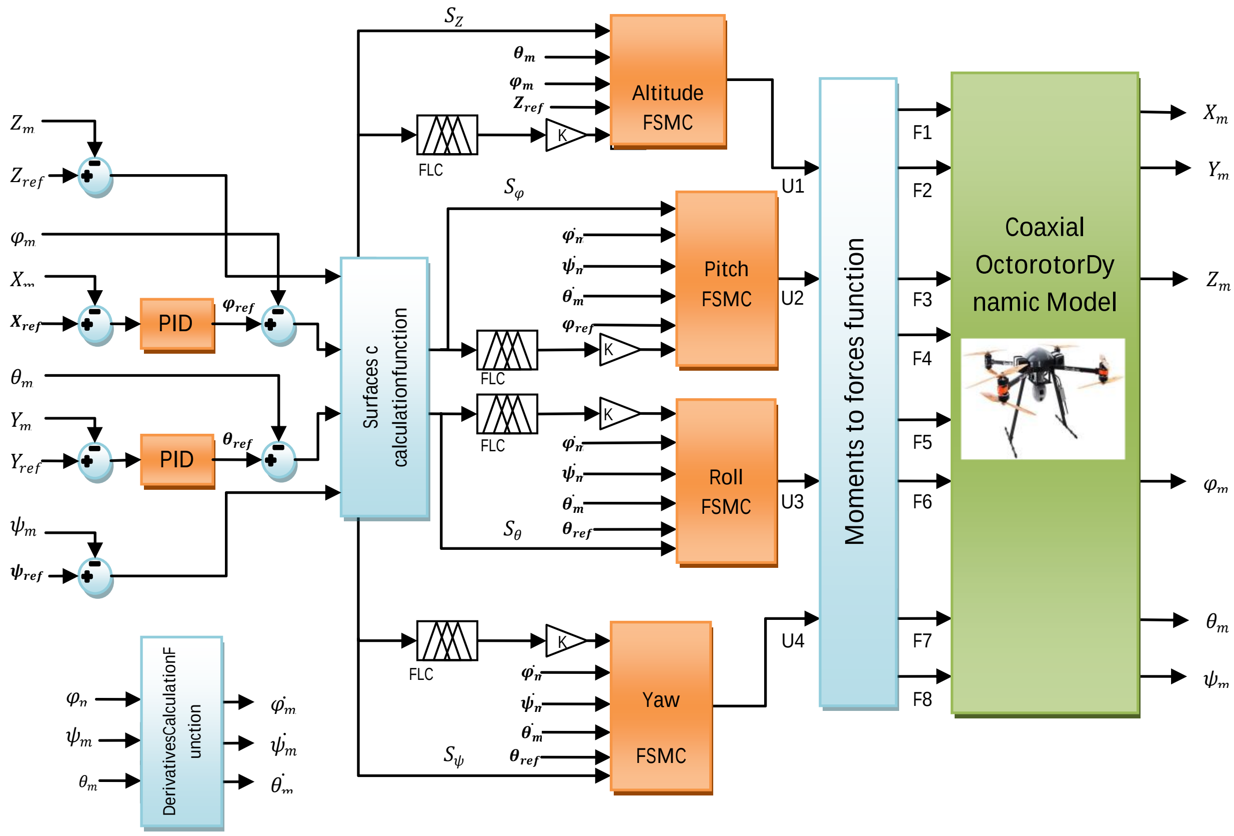


Figure III.12 The FSMC control structure for the coaxial octorotor

This hybrid design demonstrates superior stability and robustness under uncertain aerodynamic conditions, making it particularly suitable for multirotor UAVs like the coaxial octorotor, where nonlinear coupling and external disturbances are dominant.

III.6 The simulation results and discussions

In this section, we will present some simulation results for the evaluation of the proposed controllers FPDC, SMC-PID and FSMC using a MATLAB/Simulink simulation of a 6 DOF coaxial octorotor described by the dynamic model in chapter 1. Table III.3 shows the parameters of the dynamic drone model. The proposed controllers are evaluated regarding path tracking performance,the ability of each controller to steer the octorotor drone towards the desired trajectory and to keep it near its reference coordinates.

TableIII.3 Dynamic model parameters [55]

Parameter	Simulated Value
B	$2.9842\times10^{-5}\text{N}.\frac{\text{m}}{\text{rad}}/\text{s}$
d	$2.2320\times10^{-7}\text{N}.\frac{\text{m}}{\text{rad}}/\text{s}$
K_1, K_2, K_3	$(5.5670, 5.5670, 6.3540)\times10^{-2}\text{N}.\frac{\text{m}}{\text{rad}}/\text{s}$
K_4, K_5, K_6	$(5.5670, 5.5670, 6.3540)\times10^{-2}\text{N}.\frac{\text{m}}{\text{rad}}/\text{s}$
J_r	$2.8385\times10^{-5}\text{N}.\frac{\text{m}}{\text{rad}}/\text{s}^2$
I_x, I_y, I_z	$(4.2, 4.2, 7.5)\times10^{-2}\text{N}.\frac{\text{m}}{\text{rad}}/\text{s}^2$
m	1.6 Kg
l	0.23 m

III.6.1 Simulations scenarios

Two types of path tracking tasks and a robustness testapplied to evaluate the controllers performance.

III.6.1.1 Rectangular path

This reference path starts from the origin point to reach a maximum height of 2 meters. After that, the coaxial octorotor should move a maximum of 2 meters along the X-axis, and then it will reach a maximum of 2 meters along the Y-axis. While the yaw angle of the coaxial octorotor starts at 0 radians, it reaches 0.25 radians during the time range [20s, 30s] and 0.35 radians during the time range [30s, 80s]. On the other hand, the rotation finishes at 1 radian. These movements are described using step functions for the three coordinates (x, y, z, and ψ)

$$X_d=\begin{cases} 1; & t<20 \\ 2; & 20<t<60 \\ 0.5; & t>60 \end{cases} \tag{III.41}$$

$$Y_d=\begin{cases} 1; & t<30 \\ 2; & 30<t<80 \\ 0.5; & t>80 \end{cases} \tag{III.42}$$

$$Z_d=\begin{cases}2; & t<80\\ 1; & t>80\end{cases} \tag{III.43}$$

$$\psi_d=\begin{cases}0; & t<20\\ 0.25; & 20<t<30\\ 0.35; & 30<t<80\\ 1; & t>80\end{cases} \tag{III.44}$$

Where X_d,Y_d,Z_d and ψ_d are the references for the rectangular trajectory

III.6.1.2 Circular path

This reference path will force the coaxial octorotor to start from the origin point, reaching a height of 3 meters and a maximum of 2 meters along the X-axis. At this point, the coaxial octorotor will start rotating, forming a circular path with a diameter of 2 meters. The coaxial octorotor will then return to its starting point of rotation and descend straight to the origin. These movements are performed by specifying the following references

$$X_d=4.\cos\left(\left(t.0.24\text{-}1\right).\frac{\pi}{10}\right) \tag{III.45}$$

$$Y_d=4.\sin\left(\left(t.0.24\text{-}1\right).\frac{\pi}{10}\right) \tag{III.46}$$

$$Z_d=\begin{cases}1; & t<10\\ 3; & t>10\\ 0; & t>110\end{cases} \tag{III.47}$$

Where X_d,Y_d,and Z_d are the references for the circular trajectory.

III.6.1.3 Robustness test

Additionally, to the test the robustness of the controllers, an external wind disturbance in the pitch angle was applied on the coaxial octorotor model for each controller defined as follows:

$$extd=\begin{cases}0; & t<40\\ 0.2; & 40<t<42\\ 1; & t>40\end{cases} \tag{III.48}$$

Where extd is the external disturbance

III.6.2 The results analysis

This section presents the simulation results in Figures III.13 to III.18 obtained from applying the proposed control strategies to the coaxial octorotor model. While figures III.13 to III.15 illustrate the behavior of the pitch, roll, yaw, and altitude subsystems when controlledby the SMC-PID, Fuzzy PD, and FSMC controllers under a rectangular trajectory tracking scenario.

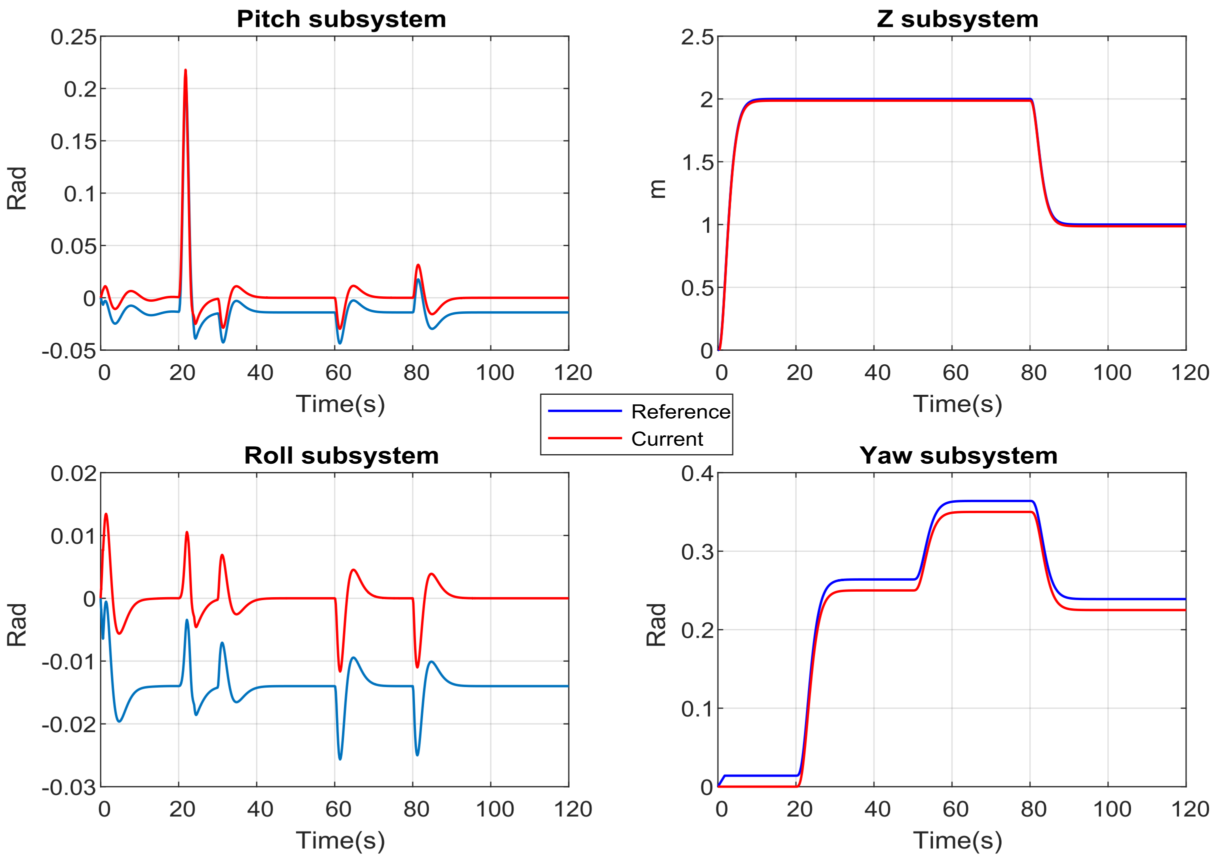


Figure III.13 SMC-PID subsystems responses

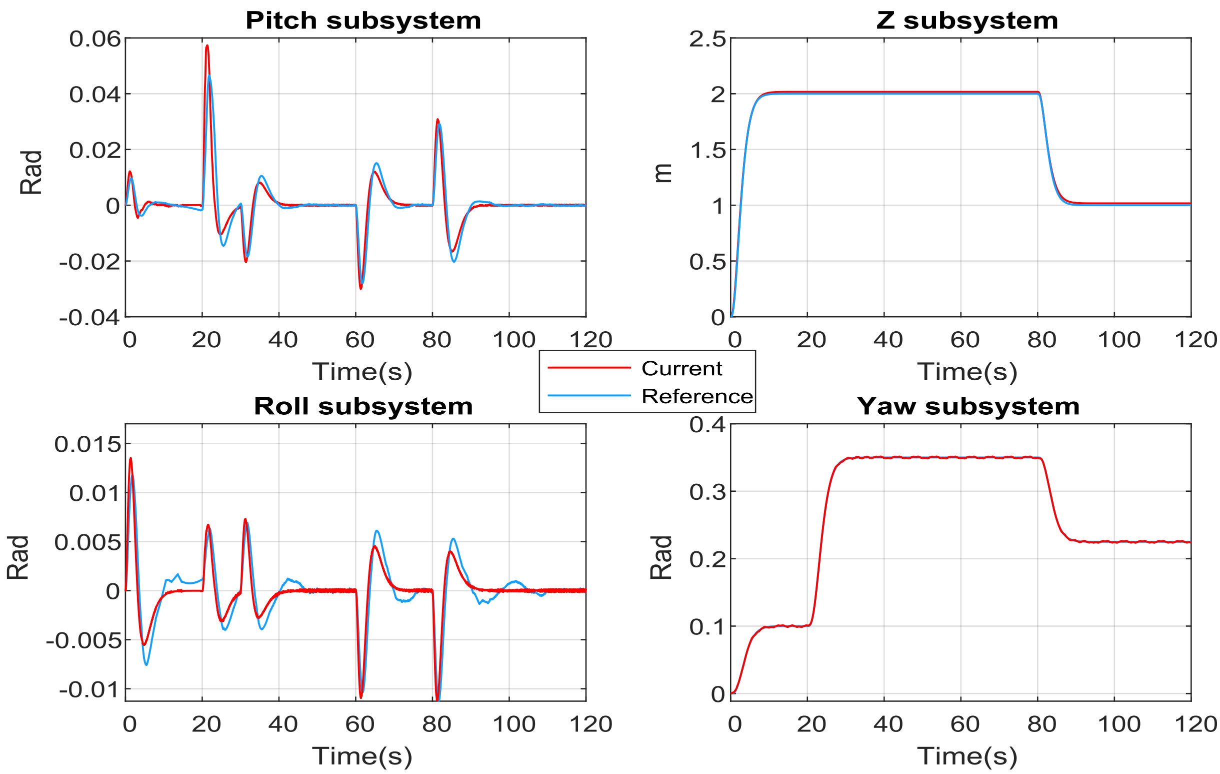


Figure III.14 Fuzzy PID subsystems responses

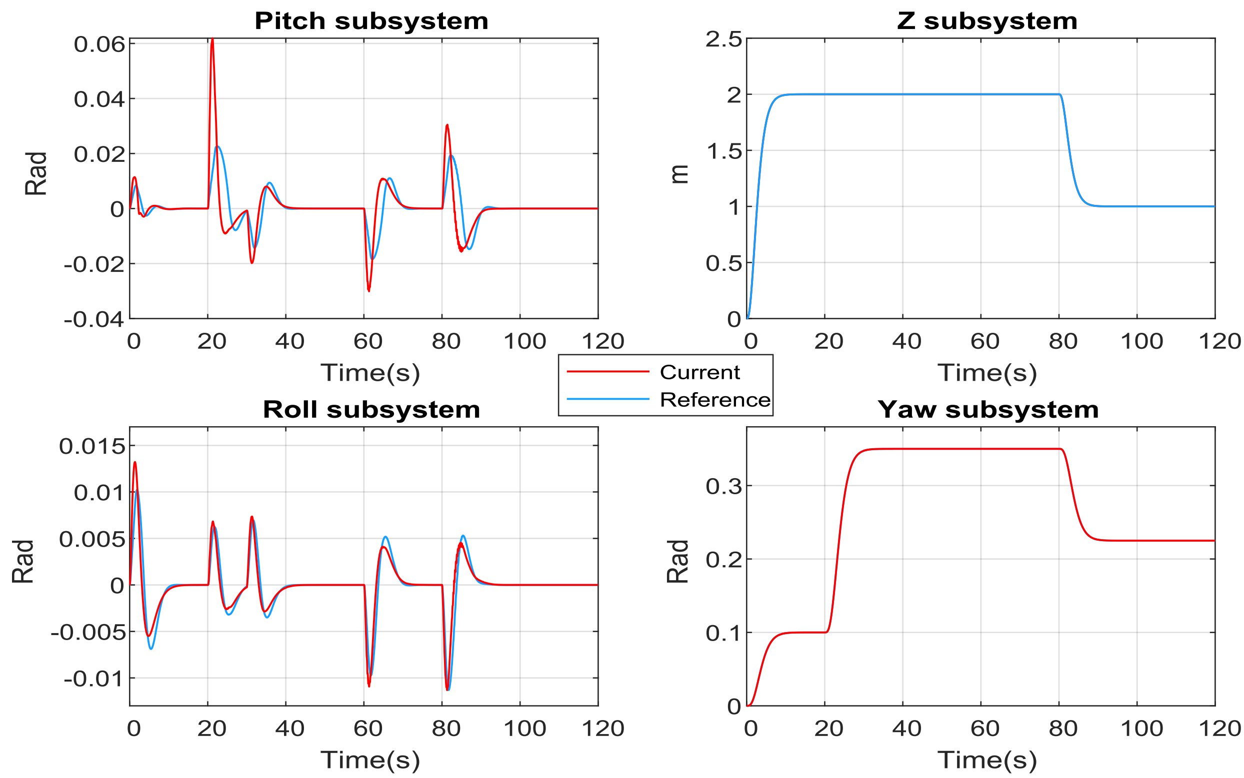


Figure III.15 FSMC subsystems responses

Figures III.13 to III.15 compare the dynamic responses of the coaxial octorotor under SMC-PID, Fuzzy PD, and FSMC controllers. The SMC-PID provides fast convergence and good robustness but suffers from strong chattering, causing oscillations in the control signals. The Fuzzy PD controller improves smoothness and reduces chattering but shows slightly slower response and minor steady-state errors. In contrast, the FSMC achieves the best overall performance by combining the robustness of sliding-mode control with the adaptability of fuzzy logic, resulting in smooth, accurate, and stable trajectory tracking with minimal overshoot and negligible steady-state error across all subsystems.

As depicted in Figure III.16 using the SMC controller, the generated action signals are affected by high chattering phenomena. Figure III.17 shows the chattering effect elimination by the means of the fuzzy logic interface.

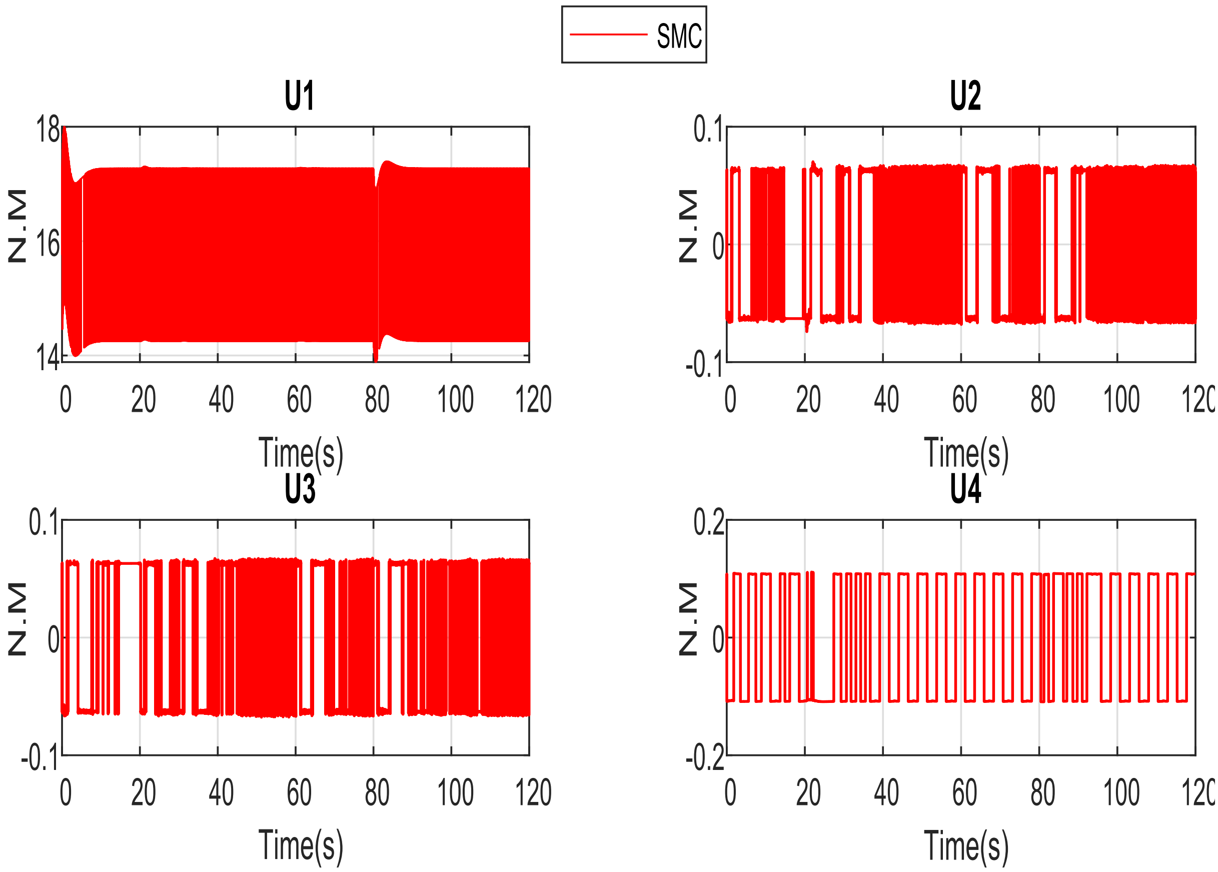


Figure III.16 SMC control signals with chattering

Figures III.18 (a) and III.18 (b) present the overall path-tracking performance of the three controllers for circular and rectangular trajectories, respectively. The simulation results confirm that all controllers maintain system stability and achieve acceptable tracking performance across the UAV's angular (pitch, roll, yaw) and positional (x, y, z) subsystems. Nevertheless, distinct differences in precision and robustness are evident among them.

The FSMC controller exhibits the best overall performance in terms of tracking accuracy, robustness, and adaptability to trajectory variations. The SMC-PID controller demonstrates strong control capability but suffers from significant chattering in the actuation signals, as shown earlier. Meanwhile, the Fuzzy PD controller performs moderately well, but its efficiency tends to decrease under complex or highly disturbed conditions. Importantly, the fuzzy-based enhancement within the FSMC effectively suppresses the chattering problem, which otherwise could lead to actuator wear or instability in practical applications.

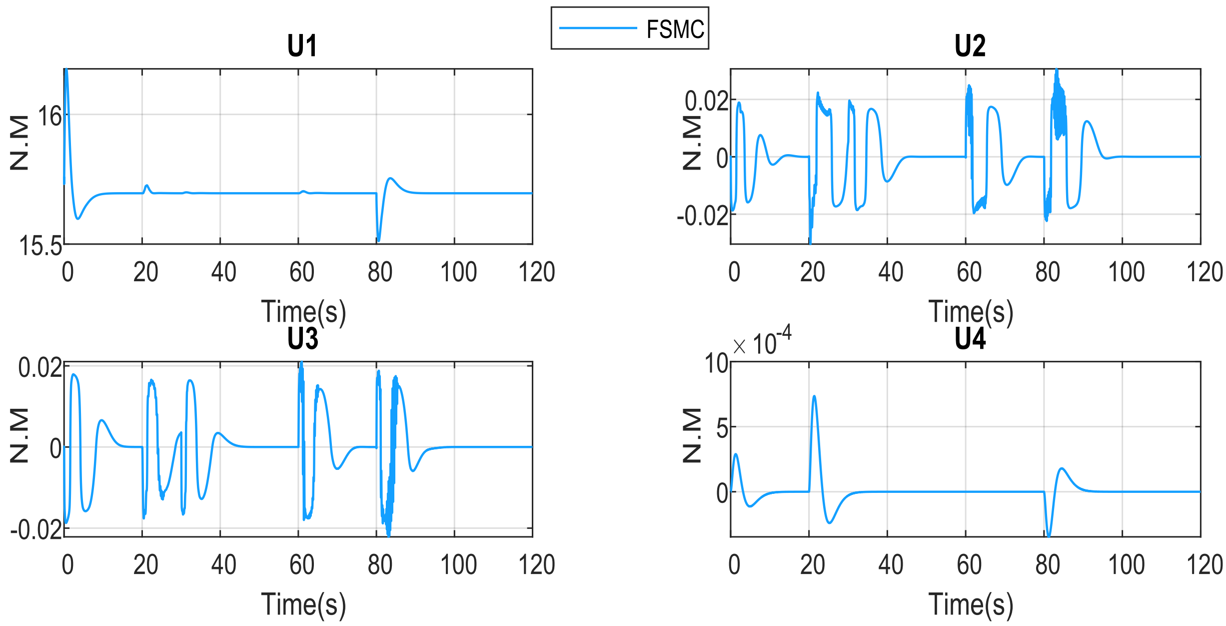
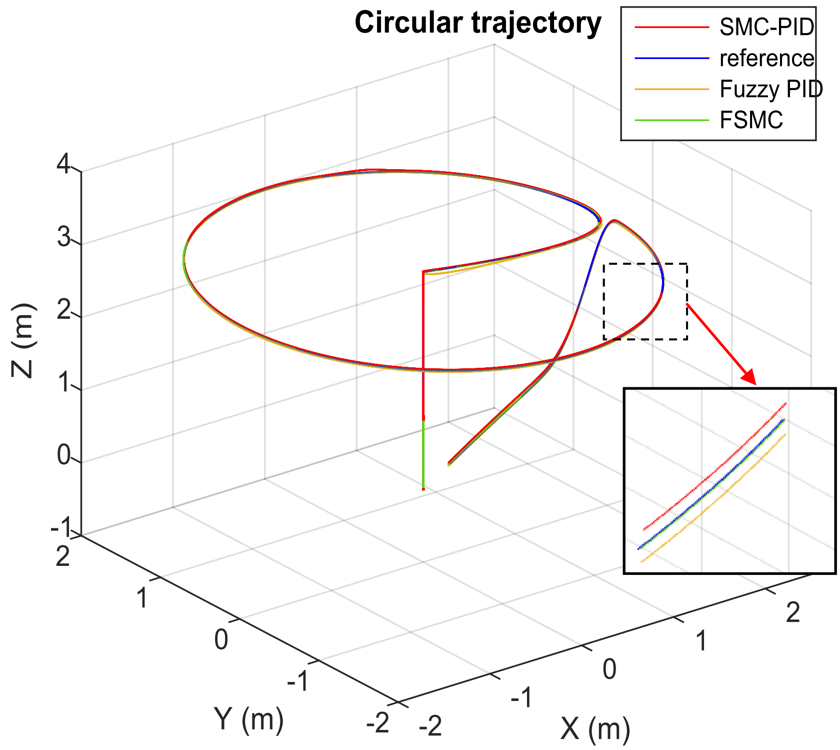
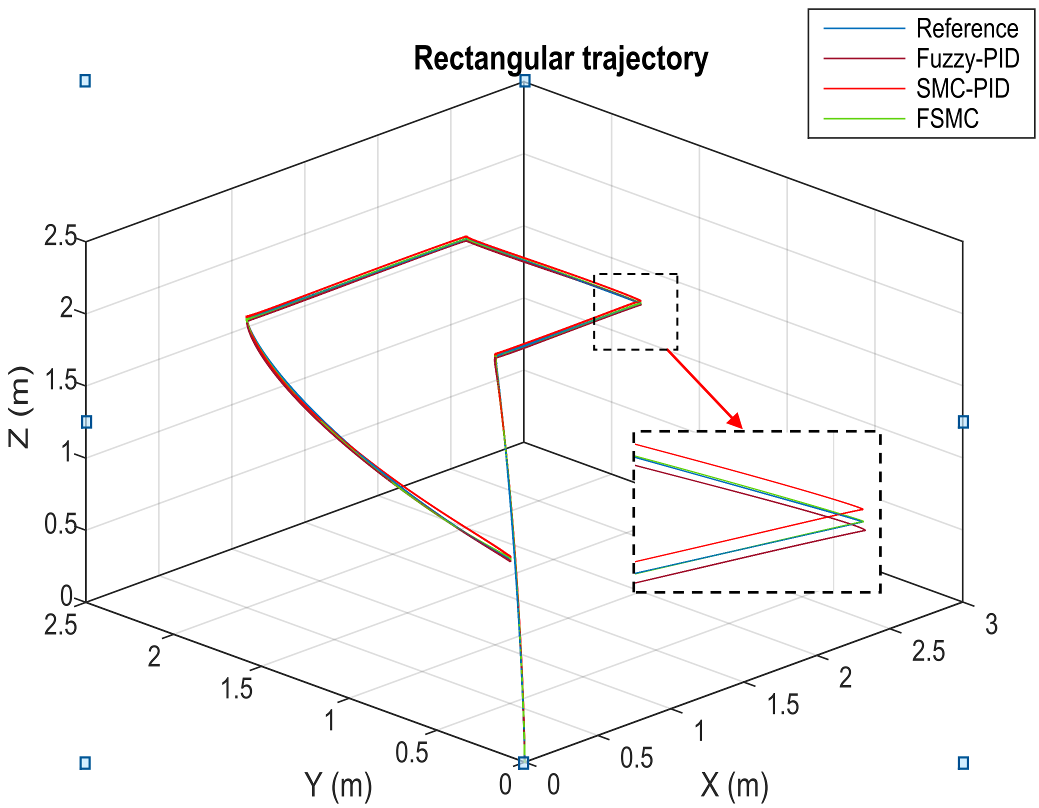


Figure III.17 FSMC control signals chattering elimination



(a)



(b)

Figure III.18 Obtained results of the tracking tasks (a) circular (b) rectangular

III.6.2.1 Robustness to the external disturbances

To assess the robustness of the proposed control schemes, a wind disturbance equivalent to a pitch deviation of 0.2 radians was introduced to the octorotor model. This disturbance primarily affects the x-position subsystem. Figure III.19 illustrates the controllers' responses to this perturbation, while Table III.4 summarizes the corresponding error performance indices Integral of Time-weighted Absolute Error (ITAE) and Integral of Absolute Error (IAE).

Table III.4 X position ITAE and IAE under external disturbance

	SMC-PID	FPDC	FSMC
ITAE	3,8136E+05	5,1022E+05	**2,7272E+05**
IAE	7,3467E+01	1,1709E+02	**5,1090E+01**

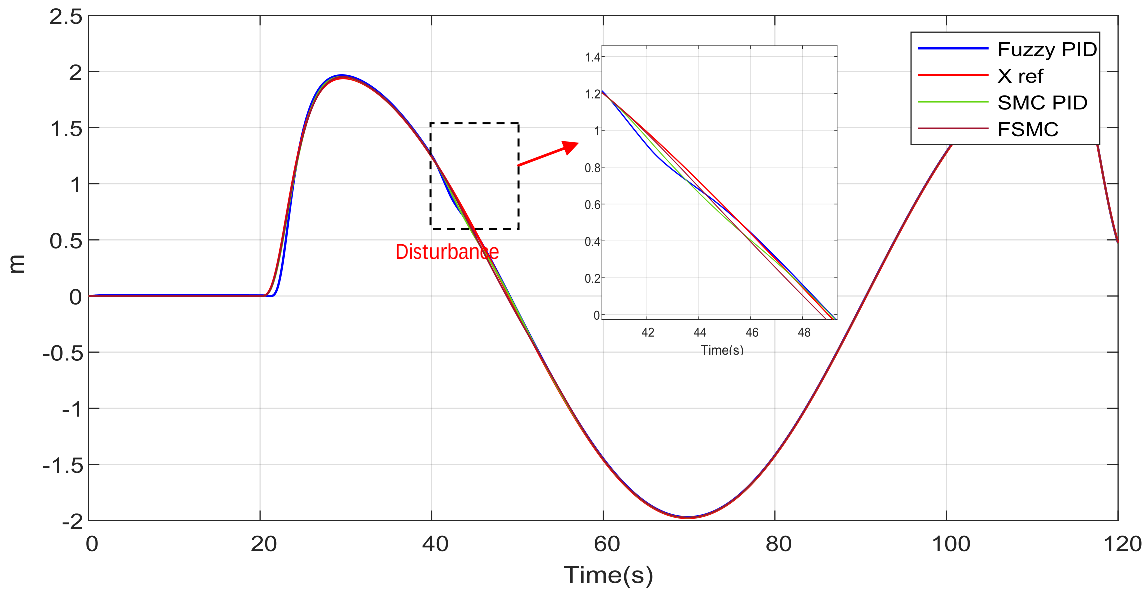


Figure III.19 Controller's responses under pitch external disturbance

All controllers successfully maintained stable flight and recovered from the disturbance, demonstrating satisfactory disturbance rejection capabilities. However, quantitative analysis of the performance indices reveals that the FSMC achieved the lowest IAE and ITAE values, indicating faster convergence and smaller deviation from the desired trajectory compared to the SMC-PID and Fuzzy PD controllers.

These findings confirm that the hybrid FSMC approach provides superior robustness and adaptability to external disturbances while ensuring precise trajectory tracking. This performance advantage stems from the synergy between fuzzy inference mechanisms and sliding-mode control principles, which collectively enhance the system's resilience and smoothness under uncertain or dynamic flight conditions.

III.6.3 Comparative study

To comprehensively assess the effectiveness of the proposed control strategies, several error performance indices and the control effort metric were employed. These quantitative indicators provide a robust means of evaluating tracking precision, transient response, and energy efficiency across different flight scenarios. Such indices are widely used in control system performance assessment because they offer measurable insights into how well a controller minimizes steady-state and transient errors [77,78].

The calculated indices namely the Integral of Time-weighted Absolute Error (ITAE), Integral of Absolute Error (IAE), and Control Effort (CE) serve to compare the performance of the three proposed controllers: the Sliding Mode PID (SMC-PID)**,** the Fuzzy PD (FPDC)**,** and the Fuzzy Sliding Mode Controller (FSMC)**.** These parameters capture the system's ability to accurately follow reference trajectories while maintaining energy-efficient actuation.

The obtained simulation results confirm that all controllers exhibit satisfactory tracking capabilities under both circular and rectangular trajectory scenarios. However, the FSMC demonstrates a clear superiority in minimizing tracking errors and improving dynamic response. This advantage arises from its hybrid structure, which combines the robustness of sliding-mode control with the adaptability and smooth decision-making of fuzzy inference [79,80]. Consequently, the FSMC achieves lower error indices and reduced control effort, confirming its enhanced stability and performance efficiency compared to the other two approaches.

- **ITAE** (integral time absolute error)

$$ITAE= \sum_{k=0}^{N} kT_s.|e(k.T_s)| \tag{III.49}$$

- **IAE** (integral absolute error)

$$IAE= \sum_{k=0}^{N} |e(k.T_s)| \tag{III.50}$$

Where k is discrete time step T_s is the sampling time

- **CE** (The control effort)

$$CE= \sum_{i=1}^{N} U_i^2 \tag{III.51}$$

Where i= [1, 2, 3, 4] is the control input number, U is the control input.

We have calculated these performance indices and the control effort for all controllers on both trajectories (circular and rectangular).

III.6.3.1 Performances indices results analysis

Figures III.20 to III.25 illustrate the performance indices obtained for both rectangular and circular trajectory-tracking experiments. These figures provide a visual comparison of the cumulative tracking errors across all subsystems (X, Y, Z, and Yaw) for each controller.

- **ITAE Analysis**

The ITAE index, which measures the cumulative tracking error over time with a time-weighted factor, serves as an indicator of long-term tracking precision. As seen in Figures III.20 and III.23, the FSMC consistently achieves the lowest ITAE values across all subsystems in both trajectories. This highlights its superior ability to suppress persistent tracking errors and maintain accurate trajectory following. The SMC-PID controller performs better than the Fuzzy PD controller, particularly in the Z-axis and yaw subsystems, owing to its robust sliding-mode dynamics. However, its performance is slightly degraded by the chattering effect that causes small fluctuations in the tracking error. The FPDC exhibits the highest ITAE values among the three, suggesting weaker long-term tracking performance and slower convergence to the desired paths.

- **IAE Analysis**

The IAE index measures the total absolute error over time without weighting by time, providing insight into overall tracking precision. As shown in Figures III.21 and III.24, the FSMC once again achieves the lowest IAE values across all subsystems, confirming its excellent short-term and steady-state accuracy. Although the SMC-PID offers better performance than the FPDC, it is still outperformed by the FSMC, which efficiently combines the robustness of SMC with the adaptability of fuzzy inference. The FPDC shows higher IAE values, especially in the X and Y subsystems, indicating less effective compensation of accumulated errors.

Overall, the FSMC demonstrates the best performance indices for both circular and rectangular paths. Its hybrid design enables it to reduce both transient and steady-state errors, offering smooth and precise tracking behavior even under dynamic disturbances.

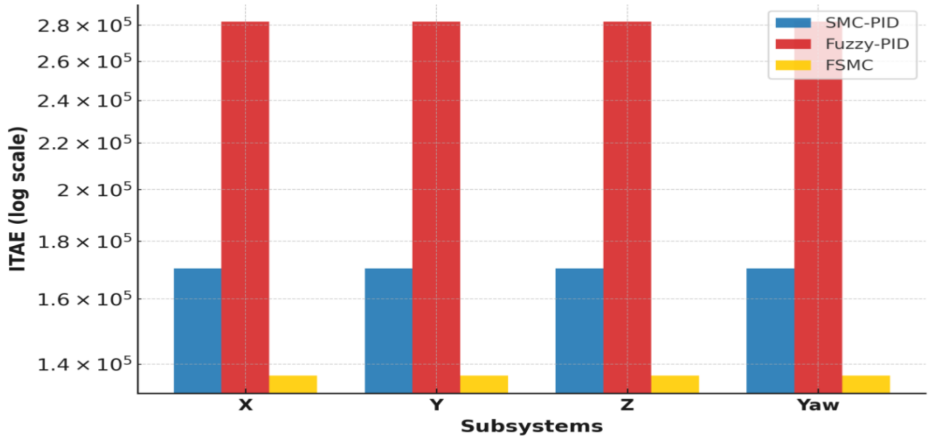


Figure III.20 ITAE values under a rectangular tracking reference

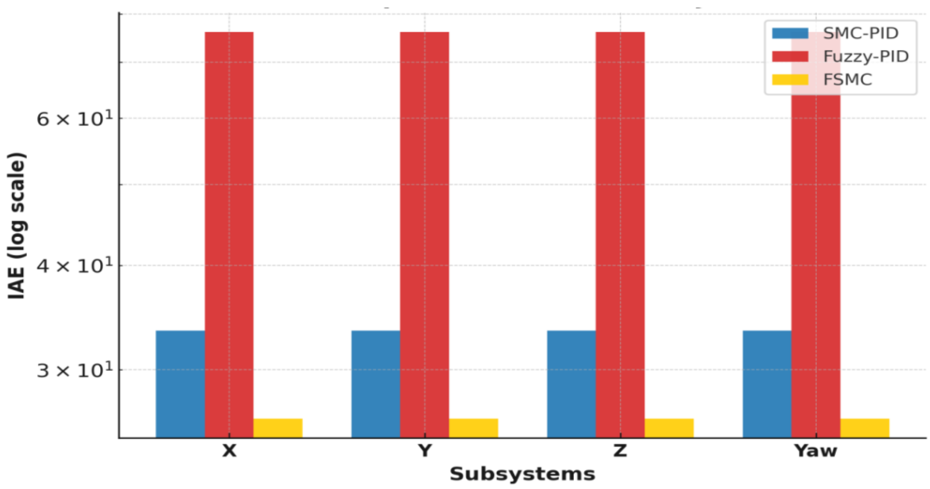


Figure III.21 IAE values under a rectangular tracking reference

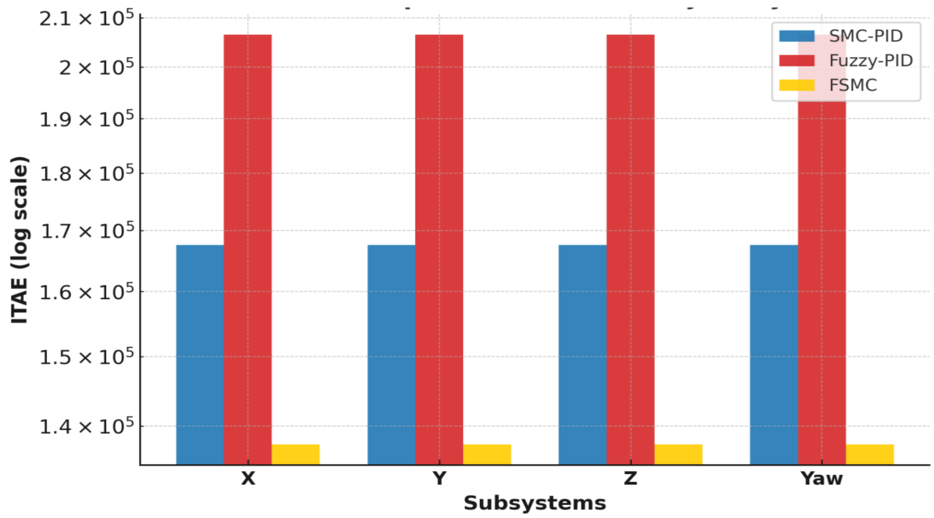


Figure III.22 ITAE values under a circular tracking reference

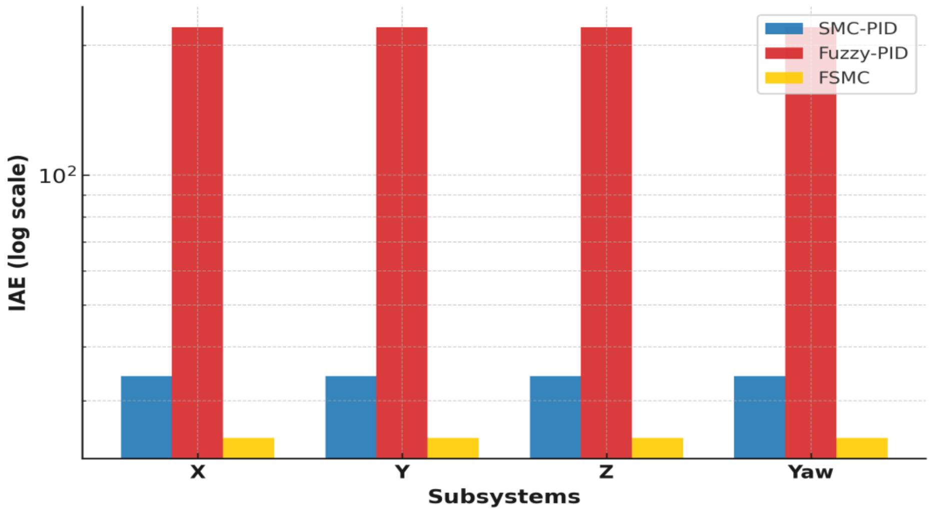


Figure III.23 IAE values under a circular tracking reference

III.6.3.2 Control effort results analysis

The control effort (CE) index reflects the total energy or actuation power required to maintain trajectory tracking. Lower CE values indicate higher energy efficiency and smoother actuator behavior, which are crucial in UAV applications to prevent actuator wear and reduce power consumption.

As depicted in Figures III.24 (rectangular) and III.25 (circular), the FSMC requires the smallest control effort among the three controllers, confirming its ability to achieve accurate control with minimal energy usage. The SMC-PID demonstrates moderate control effort lower than the FPDC but higher than the FSMC reflecting the energy cost associated with its chattering behavior. Meanwhile, the FPDC exhibits the highest control effort values, suggesting that it requires more energy to achieve comparable tracking results, which can negatively affect flight endurance and actuator stability.

For instance, in the circular trajectory, FSMC recorded a CE of approximately 1.88×10^5, compared to 1.91×10^5 for SMC-PID and 2.01×10^5 for FPDC. This confirms that FSMC not only ensures precise tracking but also operates with greater efficiency and smoother actuation an essential advantage for real-world UAV implementation.

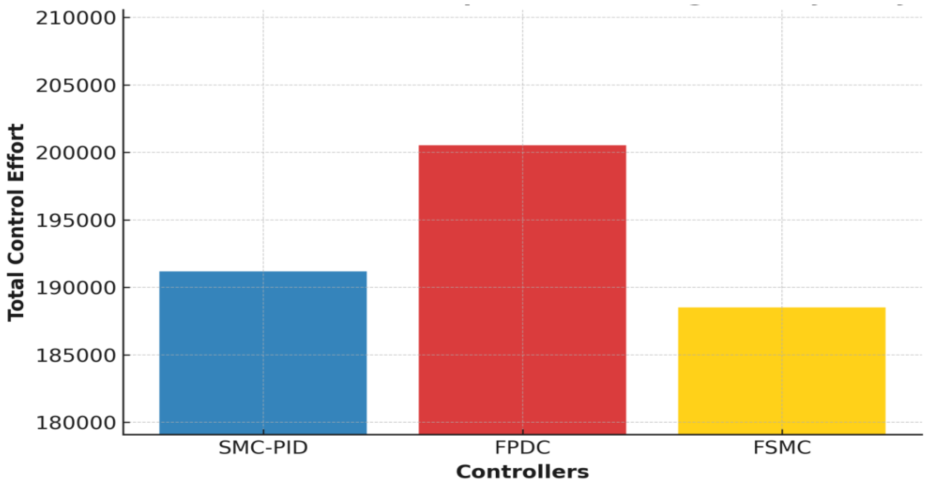


Figure III.24 The control effort using a rectangular tracking reference

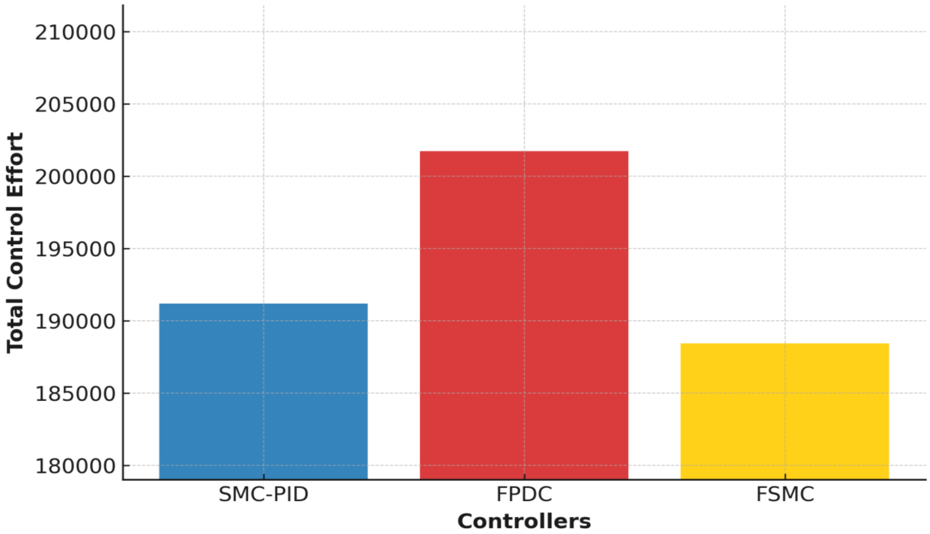


Figure III.25 The control effort using a circular tracking reference

III.6.3.3 Numerical comparison

The numerical results summarized in Tables III.5 (circular trajectory) and III.6 (rectangular trajectory) further support the observations drawn from the performance figures. The FSMC consistently outperforms the other controllers across all subsystems and trajectory types.

In the rectangular trajectory, the Z-axis ITAE achieved by FSMC is 6.78×10^3, significantly smaller than 9.98×10^5 for the FPDC and 1.23×10^6 for the SMC-PID. Similarly, for the Y-axis, the FSMC produced an IAE of 11.35, while the FPDC and SMC-PID registered 99.41 and 14.20, respectively. These values demonstrate FSMC's superior error correction and long-term stability.

In the circular trajectory, FSMC once again exhibited superior tracking precision and dynamic stability. The controller achieved the lowest ITAE and IAE values for all subsystems, confirming its excellent ability to handle nonlinear coupling and maintain trajectory accuracy throughout continuous curvilinear motion. Particularly for the yaw andaltitude (Z-axis) subsystems, the FSMC showed faster convergence and reduced steady-state deviation compared to both the SMC-PID and FPDC controllers. This demonstrates its robustness under persistent external effects such as aerodynamic coupling and centrifugal forces, which are typically more pronounced in circular motion. The results highlight the FSMC's adaptability to continuous path variations, providing a smoother and more stable trajectory tracking response.

TableIII.5 Error performance indices in the circular trajectory

		SMC-PID	FPDC	FSMC
ITAE	X	1,6761E+05	2,0651E+05	**1,3745E+05**
	Y	5,8172E+04	7,1072E+04	**5,4822E+04**
	Z	1,2945E+06	9,8957E+05	**8,1376E+03**
	yaw	5,8206E+04	1,0055E+06	**6,3881E+01**
IAE	X	3,4232E+01	2,2016E+02	**2,4583E+01**
	Y	8,3743E+00	2,7325E+01	**7,7265E+00**
	Z	2,0292E+02	1,6488E+02	**1,2371E+00**
	Yaw	9,7511E+00	1,6639E+02	**1,7965E-02**
Control effort		1,9120E+05	2,0174E+05	**1,8844E+05**

Table III.6 Error performance indices in the rectangular trajectory

		SMC-PID	FPDC	FSMC
ITAE	X	1,7030E+05	2,8192E+05	**1,3672E+05**
	Y	6,5648E+04	9,4839E+04	**5,1676E+04**
	Z	1,2315E+06	9,8957E+05	**6,7874E+03**
	yaw	5,8206E+04	1,0055E+06	**6,3881E+01**
IAE	X	3,3416E+01	7,6065E+01	**2,6215E+01**
	Y	1,4202E+01	3,0249E+01	**1,1355E+01**
	Z	2,0004E+02	1,6487E+02	**1,1841E+00**
	Yaw	9,7511E+00	1,6639E+02	**1,7965E-02**
Control effort		1,9119E+05	2,0054E+05	**1,8852E+05**

III.7 Conclusion

This chapter presented a comparative investigation of three fuzzy-based nonlinear control approaches Fuzzy PD Controller (FPDC), Sliding Mode PID (SMC-PID), and Fuzzy Sliding Mode Controller (FSMC)for the altitude and attitude control of a 6-DOF coaxial octorotor UAV. The primary objective was to evaluate their effectiveness in terms of trajectory tracking accuracy, disturbance rejection, and control efficiency using a detailed nonlinear model of the aerial system.

The FSMC controller integrates a Type-1 Mamdani fuzzy inference system into the switching component of the sliding-mode framework. This fuzzy mechanism generates a continuous control signal that replaces the discontinuous sign function, thereby eliminating chattering while preserving the robustness and finite-time convergence properties of SMC. Through careful tuning of fuzzy rules and membership functions, the FSMC achieved smooth transient responses, rapid error convergence, and stable control actions across all tested flight scenarios.

Simulation experiments were conducted under both rectangular and circular trajectory-tracking tasks, including the presence of external wind disturbances. The obtained results consistently showed that the FSMC achieved the lowest Integral of Absolute Error (IAE) and Integral of Time-weighted Absolute Error (ITAE) values, confirming its superior tracking accuracy and robustness. Additionally, it demonstrated the lowest control effort, reflecting its energy-efficient behavior and suitability for long-duration flights. In contrast, while the FPDC offered simpler implementation and satisfactory performance, it lacked the precision and adaptability required for highly nonlinear or disturbed conditions. The SMC-PID maintained strong disturbance rejection but suffered from chattering and higher energy consumption, limiting its applicability in real UAV systems.

Overall, the FSMC provided the most balanced and reliable performance, combining the robustness of sliding-mode control with the adaptive reasoning of fuzzy logic. Its ability to suppress chattering, reduce energy usage, and maintain accurate trajectory tracking under varying conditions confirms its potential for real-time autonomous UAV applications.

Looking forward, future chapter will explore adaptive and learning-based tuning techniquesincluding optimization algorithms and neural network-based adaptive fuzzy systems combined with FSMC to enhance the control performance of the coaxial octortor.

Chapter IV :

Hybrid Control Strategy using ANFIS, FSMC and PSO Optimized PID

IV.1 Introduction

The rapid evolution of robotic and unmanned aerial systems has reshaped modern control design, placing strong emphasis on autonomy, robustness, and adaptability. Authoritative references in robotics and UAV theory highlight that advanced aerial platforms-such as coaxial octorotor UAVs exhibit highly nonlinear, underactuated, and strongly coupled dynamics, while being continuously exposed to modeling uncertainties and external disturbances [1,2]. These intrinsic characteristics make conventional linear control strategies insufficient for achieving high-performance autonomous operation. Addressing this challenge has been the central motivation of the preceding chapters of this thesis, which progressed from a rigorous first-principles dynamic modeling framework (Chapter I) to the design and evaluation of intelligent control strategies based on fuzzy logic (Chapter II) and hybrid fuzzysliding mode control (Chapter III).

The results obtained in these chapters demonstrated that intelligent and hybrid control paradigms offer clear advantages over classical approaches. In particular, sliding mode control is well recognized for its robustness against parametric uncertainties and disturbances [8,9]. However, its practical implementation is often limited by the chattering phenomenon caused by the discontinuous switching function. The investigations conducted in Chapter III confirmed that the integration of fuzzy inference within the sliding mode framework resulting in the Fuzzy Sliding Mode Controller (FSMC) successfully alleviates chattering while preserving stability and robustness. Nevertheless, the application of a single control strategy uniformly across all degrees of freedom does not fully exploit the heterogeneous dynamic characteristics of a coaxial octorotor.

Motivated by this observation, this chapter presents a hierarchically structured hybrid control architecture that constitutes the culmination of the proposed control design methodology. The architecture relies on a subsystem-oriented integration of three complementary computational intelligence and optimization techniques, each selected according to the dynamic nature and control requirements of the corresponding subsystem.

First, an Adaptive Neuro-Fuzzy Inference System (ANFIS), originally introduced by Jang [6], is employed for the control of the highly coupled and fast inner-loop attitude dynamics (pitch and roll). By combining fuzzy inference with neural learning capabilities, ANFIS enables adaptive adjustment of control rules based on data, offering superior compensation for unmodeled nonlinearities compared to fixed-rule fuzzy controllers [81,82].

Second, the FSMC developed and refined in the previous chapter is adopted for altitude and yaw control. By embedding fuzzy reasoning within the sliding mode framework, this controller generates a smooth switching signal that suppresses chattering while retaining the inherent robustness of sliding mode control under disturbances and uncertainties [8,9].

Finally, for outer-loop planar position control, the classical PID structure is preserved due to its simplicity and reliability, but its parameters are optimally tuned using Particle Swarm Optimization (PSO). PSO, inspired by collective social behavior, is a powerful global optimization technique that has proven effective for controller tuning in nonlinear and multivariable systems [10,11].

The central premise of this chapter is that a deliberate, subsystem-specific synthesis of ANFIS, FSMC, and PSO-optimized PID controllers can outperform any single intelligent control strategy. This hypothesis is evaluated through high-fidelity MATLAB/Simulink simulations, including a realistic solar farm surveillance scenario and comprehensive disturbance tests. Performance is assessed in terms of tracking accuracy, robustness, and control effort, with quantitative comparisons against a conventional Type-1 Fuzzy Logic Controller, thereby demonstrating the effectiveness and practical relevance of the proposed hybrid control framework.

IV.2 Adaptive Neuro Fuzzy Inference System (ANFIS)

Fuzzy Inference Systems (FIS) and Artificial Neural Networks (ANNs) each possess distinct advantages. Hybrid neuro-fuzzy approaches combine the strengths of both techniques integrating the learning and adaptation capabilities of neural networks with the interpretability and flexibility of fuzzy inference systems. The most common form of integration involves representing a fuzzy inference system as a multilayer neural network, where the connection weights correspond to the parameters of the fuzzy system. This structure enables the hybrid model to benefit from neural learning while preserving the reasoning transparency inherent to fuzzy logic.

IV.2.1 The ANFIS structure

The combined use of neural networks and fuzzy logic makes it possible to exploit the strengths of both approaches the learning and adaptation capabilities of the former and the transparency and flexibility of the latter. Since the late 1980s, several forms of integration between these two paradigms have been developed, leading to the emergence of neuro-fuzzysystems. These systems are primarily designed for the control of complex nonlinear processes and for tackling pattern recognition and classification problems, where both learning efficiency and interpretability are essential.

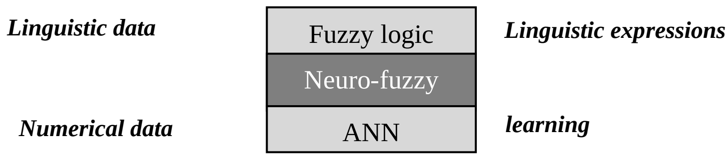


Figure IV.1 ANFIS scheme

A hybrid neuro-fuzzy system can be viewed as an intelligent framework that integrates the learning capability of artificial neural networks with the linguistic reasoning and interpretability of fuzzy logic. In such systems, fuzzy sets, membership functions, and inference rules are embedded within a neural network structure, while adaptive learning mechanisms are employed to tune the parameters automatically using available data [6]. This synergy allows the model to exploit data-driven learning while preserving the flexibility and transparency of fuzzy inference, making neuro-fuzzy approaches particularly effective for the modeling, identification, and control of complex nonlinear systems characterized by uncertainty and imprecision.

IV.2.2 Neuro fuzzy combination methods

In the literature, the integration of neural networks and fuzzy logic has been realized through several conceptual frameworks. Broadly, four principal approaches are commonly identified, each representing a distinct strategy for merging the learning and adaptation capabilities of neural networks with the interpretability and approximate reasoning mechanisms of fuzzy logic systems [12]. These integration paradigms differ in terms of system structure, learning methodology, and the degree of interaction between neural and fuzzy components, but all aim to enhance system intelligence and adaptability.

- **Neural Fuzzy Networks**

In this configuration, fuzzy concepts are embedded directly into neural network architectures to enhance their learning and performance. Fuzziness can be introduced into neurons, connection weights, or activation functions to improve adaptability, robustness, and generalization during training. As a result, the network can better handle uncertain or imprecise data, making it more flexible in real-world applications.

- **Simultaneous Neural/Fuzzy Systems**

In this method, the neural network and the fuzzy logic system operate side by side on the same problem but without tightly coupling their internal processes. Neither system modifies or controls the parameters of the other. Typically, the neural network handles pre-processing by transforming input signals before they reach the fuzzy inference stage, or it performs post-processing to refine the fuzzy system’s output. This cooperative structure allows both components to contribute to decision-making while maintaining their distinct functions.

- **Cooperative Neuro-Fuzzy Models**

In this category, the neural network plays a supportive role by tuning or determining the parameters of the fuzzy inference system such as fuzzy rules, membership functions, or rule weights. Once the learning or training phase is complete, the fuzzy system can function independently of the neural network. This approach strikes a balance between learning efficiency and interpretability, as the resulting fuzzy system remains transparent and easily understandable [12].

- **Hybrid Neuro-Fuzzy Models**

Most modern neuro-fuzzy systems fall into this integrated category. In hybrid models, neural and fuzzy components are combined into a single, cohesive framework. The system can be viewed either as a neural network with fuzzy parameters or as a fuzzy inference system implemented in a distributed, parallel fashion. This hybrid structure merges the adaptive learning capabilities of neural networks with the explainable, rule-based reasoning of fuzzy logic. It is particularly effective for complex control, pattern recognition, and decision-making applications, where adaptability and interpretability are equally important[81,12].

IV.2.3 Integrated Hybrid Neuro-Fuzzy Systems

IV.2.3.1 Mamdani-Type Neuro-Fuzzy Systems

The Mamdani-type neuro-fuzzy system employs the backpropagation learning algorithm to adjust the parameters of the membership functions. It consists of five distinct layers:

- **Input layer:** receives and distributes input variables to the next layer.
- **Fuzzification layer:** transforms crisp input values into fuzzy linguistic terms using predefined membership functions.
- **Antecedent rule layer:** represents the if-part of fuzzy rules, computing the degree of activation for each rule.
- **Consequent rule layer:** represents the then-part of the rules, where the fuzzy conclusions are formed.
- **Defuzzification layer:** converts the aggregated fuzzy output into a crisp control action or numerical value.

This structure allows the system to learn and adapt fuzzy membership parameters through iterative training. Several Mamdani-type neuro-fuzzy models have been proposed in the literature, including FALCON (Fuzzy Adaptive Learning Control Network) and GARIC (Generalized Approximate Reasoning-Based Intelligent Control), which have demonstrated strong performance in control and decision-making tasks[12,6].

IV.2.3.2 Takagi–Sugeno-Type Neuro-Fuzzy Systems

Takagi–Sugeno neuro-fuzzy systems combine the backpropagation algorithm for learning the parameters of the membership functions with the least-squares estimation method for determining the coefficients of the linear combinations in the rule consequents. This hybrid learning strategy enables the system to efficiently identify nonlinear relationships between inputs and outputs, providing faster convergence and improved numerical stability compared to purely gradient-based learning. The resulting model is particularly effective for system identification, adaptive control, and prediction tasks, where accuracy and computational efficiency are essential.

IV.2.3.3 ANFIS (Adaptive Neuro-Fuzzy Inference System)

The Adaptive Neuro-Fuzzy Inference System (ANFIS) is a specific class of adaptive network architecture proposed by [6]. ANFIS can be viewed as a feedforward neural network, where each layer represents a component of a Takagi–Sugeno fuzzy inference system. It is among the most widely used neuro-fuzzy models in practice, with successful applications in signal processing, adaptive filtering, and control engineering. Several studies have demonstrated that ANFIS provides superior performance in control and estimation tasks, particularly when applied to motor control and nonlinear dynamic systems.

The ANFIS architecture refines fuzzy rules initially defined by human experts to describe the input–output behavior of complex systems. A modified version of the ANFIS can also implement Tsukamoto-type fuzzy inference, while the original formulation is based on the first-order Sugeno model. In the Tsukamoto variant, the global output is computed as a weighted average of each rule's individual output, where the weights correspond to the degree of rule activation (calculated as the product or minimum of the antecedent membership values) and the output membership functions.

IV.3 ANFIS fundamentales

The ANFIS model consists of five layers, each performing a specific computational role. The learning process focuses exclusively on parameter adaptation within a fixed structure, where each linguistic term corresponds to a single fuzzy set. Parameter updates are carried out using a hybrid learning approach that combines backpropagation for the antecedent parameters and least-squares estimation for the consequent parameters. This approach provides a good trade-off between interpretability and learning efficiency.

IV.3.1 Structure of the ANFIS Model

In this work, the Adaptive Neuro-Fuzzy Inference System (ANFIS) is adopted as an automatic fuzzy rule generation framework based on a first-order Takagi–Sugeno fuzzy inference model. This structure, originally introduced by J.-S. R. Jang, enables the systematic construction and tuning of fuzzy rules through learning from data [6]. The ANFIS architecture is organized into five successive layers, where each layer is responsible for a distinct stage of the fuzzy inference process, including fuzzification, rule evaluation, normalization, consequent computation, and output aggregation [13]. This layered organization facilitates both interpretability and efficient parameter adaptation.

To illustrate the basic architecture of the adaptive neuro-fuzzy system, consider a first order Sugeno fuzzy model with two input variables, x1 and x2, and one output y. Suppose the rule base consists of two fuzzy rules:

Rule 1:*If* x_1 *is* A_1 *and* x_2 *is* B_1*, Then* $y_1=p1x1+q_1x_2+r_1$

Rule 2:*If* x_1 *is* A_2 *and* x_2 *is* B_2*, Then* $y_2=p_2x_1+q_2x_2+r_2$

Where x1 and x2 are the input variables, A1, A2, B1, B2 are fuzzy sets associated with the input linguistic terms, yi are the outputs of the defuzzification layer, and pi,qi,ri are the consequent parameters of rule i, determined during the training process.

The overall ANFIS network structure representing this model is illustrated in Figure IV.2, showing the five computational layers that define the flow of information from inputs to the final output through fuzzification, rule evaluation, normalization, and defuzzification.

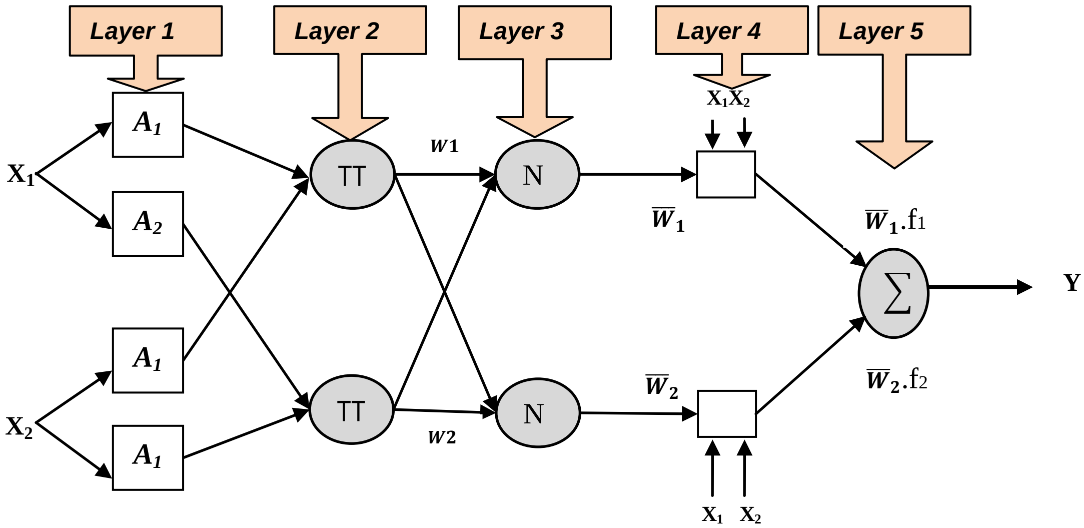


Figure IV.2 ANFIS structure

The ANFIS network Jang, 1993 is a multilayer feedforward structure in which the connections are either unweighted or assigned unit weights. The network nodes are divided into two categories based on their functionality:

- **Square nodes:**

which contain adaptable parameters subject to training, and

- **Circular nodes:**

which perform fixed operations without parameters.

The output of node i in layer k, denoted as O_i^k, depends on the signals received from the preceding layer (k−1) and on the specific parameters associated with node (i,k). Mathematically, this relationship can be expressed as:

$$O_i^k = f({x_i}^{k-1}, \theta_{i,k}) \qquad \text{(IV.1)}$$

Where ${x_i}^{k-1}$represents the input signals from layer (k−1), and$\theta_{i,k}$denotes the set of adjustable parameters linked to the node (i,k).

IV.3.1.1 Layer (1): Fuzzification

Each node in the first layer (Layer 1) of the ANFIS architecture performs a fuzzification operation, converting the crisp input values into corresponding degrees of membership. The function of each node can be defined as follows:

$$O_i^1 = \begin{cases} \mu_{A_i}(x_1), for i = 1{,}2 \\ \mu_{B_i}(x_2), for i = 3{,}4 \end{cases} \tag{IV.2}$$

where x_1 and x_2 are the input variables associated with nodes {1,2}and {3,4} respectively.

The terms Ai and B_{i-2} denote the linguistic labels corresponding to the membership functionsμ_{A_i} and μ_{B_i}.

Hence, the outputs $O_i^1$of the first layer represent the membership degrees of the input variables x_1 and x_2 with respect to the fuzzy sets Ai and B_{i-2}.

In Jang's original ANFIS model, the membership functions are typically chosen to be Generalized bell functions, expressed as IV.3 and showed in figure IV.3.

$$\mu_{Ai}(x1)\ \ ou\ \ \mu_{Bi}(x1) = \frac{1}{1+\left[\left(\frac{x_i-c_i}{a_i}\right)^2\right]^{b_i}} \tag{IV.3}$$

where c_i,a_i and b_i are the center and width (spread) parameters of the Generalized bell function, respectively. These parameters are adjusted during the learning phase to best represent the fuzzy partitions of the input space.

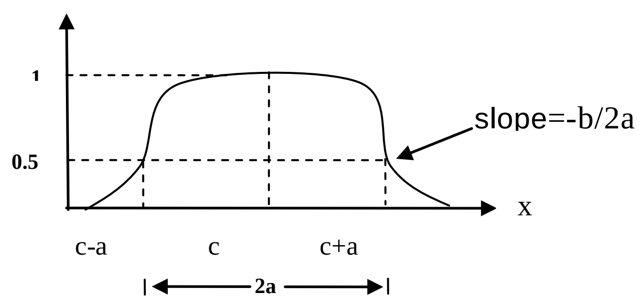


Figure IV.3 Generalized bell membershipe function

IV.3.1.2 Layer (2): Fuzzy Rule Layer

The second layer of the ANFIS structure corresponds to the fuzzy rule layer. In this layer, each neuron is associated with a single fuzzy rule and is responsible for calculating the rule's activation level, commonly referred to as its firing strength or synaptic weight. The nodes in this layer are of fixed type and are typically represented by the multiplication operator (Π), as they compute the product of the input membership degrees. This operation effectively realizes the fuzzy logical AND, allowing the simultaneous satisfaction of multiple antecedent conditions within a rule [6]. Consequently, the output of each node in this layer is given by:

$$O_i^2 = w_i = \mu_{A_i}(x_1) \times \mu_{B_i}(x_2),\ for i = 1{,}3 \tag{IV.4}$$

Here, wi represents the firing strength of the ith fuzzy rule, corresponding to the degree of activation of that rule's antecedent part. The higher the value of wi, the more strongly the corresponding rule contributes to the final system output.

IV.3.1.3 Layer (3): Normalization Layer

The third layer of the ANFIS architecture performs the normalization of the fuzzy rule firing strengths computed in the previous layer. The nodes in this layer are also fixed-type circular nodes, denoted by N. Each node calculates the normalized firing strength of a corresponding fuzzy rule according to the following relation:

$$O_I^3 = \overline{w_i} = \frac{w_i}{w_1 + w_2}, for i = 1{,}2 \tag{IV.5}$$

Here, $\overline{w_i}$ represents the normalized activation degree of the ith rule. This process ensures that the sum of all normalized firing strengths equals one, thereby defining the relative contribution of each fuzzy rule to the final output.

IV.3.1.4 Layer (4): Consequent (Defuzzification) Layer

The fourth layer of the ANFIS structure corresponds to the consequent, or defuzzification, layer. Each node in this layer is adaptive and computes the output contribution of its corresponding fuzzy rule based on the normalized firing strength obtained from the previous layer. The operation performed by each node is defined as:

$$O_i^4 = \overline{w_i} f_i = \overline{w_i}(p_i x_1 + q_i x_2 + r_i) for i = 1{,}2 \tag{IV.6}$$

Here, pi, qi and ri are the consequent parameters of the ith fuzzy rule. These parameters are determined during the learning process and represent the coefficients of the linear function defining the rule's output. Thus, this layer computes the weighted output of each rule by combining the normalized activation level with the corresponding linear consequent expression.

IV.3.1.5 Layer (5): Summation (Output) Layer

The fifth layer of the ANFIS architecture, known as the summation or output layer, consists of a single neuron that generates the overall output of the network. This node performs a weighted summation of all the outputs produced by the previous (consequent) layer. The final network output is computed using the following relation:

$$O_i^5 = y = \sum_{i=1}^{2} \overline{w_i} f_i = \frac{\sum_i^n w_i f_i}{\sum_i^n w_i}, \; for\ i = 1,2 \qquad \text{(IV.7)}$$

Here, y represents the final output of the ANFIS network, which is obtained as the normalized weighted average of all rule outputs. This operation effectively combines the contributions of each fuzzy rule according to their activation strengths, resulting in a smooth and continuous mapping from the input space to the output space. As remarks :

- The ANFIS architecture includes two adaptive layersthe first and the fourtheach responsible for learning different sets of parameters during the training process.
- The first layer contains three adjustable parameters, {ai,bi,ci}, which define the shapes of the input membership functions. These parameters are referred to as the premise parameters.
- The fourth layer also includes three adjustable parameters, {pi,qi,ri}, known as the consequent parameters. These coefficients determine the linear functions in the rule consequents and are updated during the learning phase to optimize the network's overall performance [6].

IV.3.2 ANFIS Learning Algorithm

The learning mechanism of ANFIS is designed to adjust both the antecedent (premise) parameters and the consequent parameters while maintaining a fixed network topology. This training process relies on a set of input–output data and seeks to minimize the discrepancy between the system's predicted output and the corresponding target values through iterative parameter tuning [14]. By optimizing these parameters, ANFIS is able to capture nonlinear relationships and improve its approximation and control performance.

The algorithm begins by constructing an initial network, after which a backpropagation-based learning method is applied. To improve convergence and accuracy, [6] proposed a hybrid learning rule that combines the gradient descent algorithm with the least-squares estimation (LSE) method. The overall ANFIS output can be expressed as:

$$f = \overline{w_1} f_1 \overline{w_2} f_2 \tag{IV.8}$$

Substituting the expressions for f_1 and f_2, we obtain:

$$f = \overline{w_1}(p_1 x_1 + q_1 x_2 + r_1) + \overline{w_2}(p_2 x_1 + q_2 x_2 + r_2) \tag{IV.9}$$

or equivalently,

$$f = (\overline{w_1} x_1) q_1 + (\overline{w_1} x_2) p_1 + \overline{w_1} r_1 + (\overline{w_2} x_1) q_2 + (\overline{w_2} x_2) p_2 + \overline{w_2} r_2 \tag{IV.10}$$

This expression represents a linear combination of the adjustable consequent parameters{p1,q1,r1,p2,q2,r2}

In this hybrid learning algorithm, both sets of parameters the premise and the consequentare optimized alternately during training:

- In the forward pass, the consequent parameters are identified using the least-squares estimation method, while the premise parameters are kept constant.
- In the backward pass, the premise parameters are updated using the gradient descent method, with the consequent parameters held fixed.

This alternating optimization process enables ANFIS to achieve fast convergence and improved generalization performance (as illustrated in Figure IV.4).

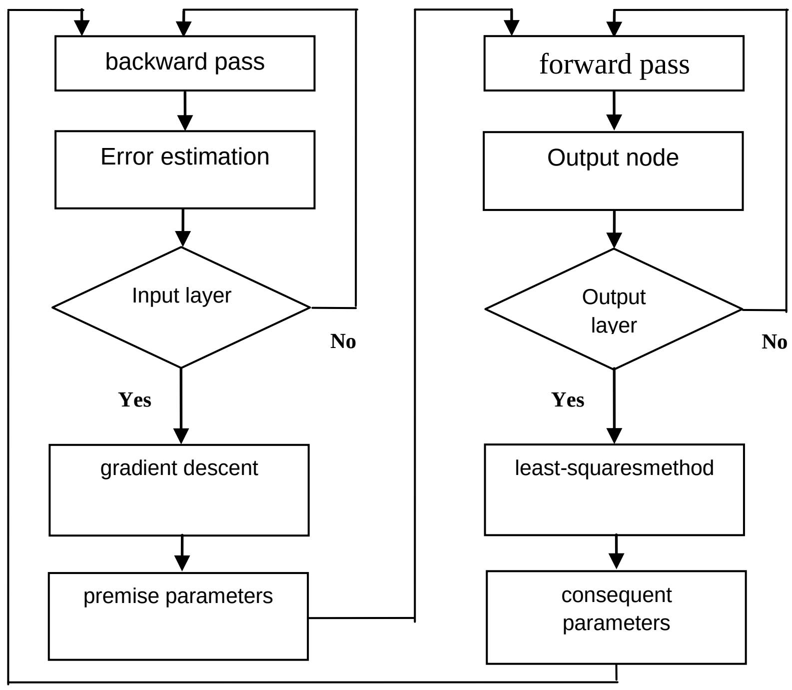


Figure IV.4 ANFIS learningalgorithm

In the Forward and Backward Passes in ANFIS Learning. The ANFIS learning process employs two distinct passesa forward pass and a backward pass to update its parameters efficiently. Eachpassplays a complementary role in optimizing the premise and consequent parameters of the network.

IV.3.2.1 Forward Pass

During the forward pass Figure IV.4, the network computes the output values based on the current parameter estimates. The least-squares estimation (LSE) method is then applied to identify the consequent (linear) parameters [6]. For the set of premise (nonlinear) parameters, the training data are used to construct a system of equations expressed as:

$$AX = B \tag{IV.11}$$

where X represents the matrix containing the unknown consequent parameters. This constitutes a linear optimization problem, where the goal is to minimize the squared error $\|AX-B\|^2$. The optimal solution is given by:

$$X^* = \left(A^T A\right)^{-1} A^T B \tag{IV.12}$$

In thisstep:

- S1 represents the premise parameters, which are kept fixed,
- S2 represents the consequent parameters, which are computed using the least-squares estimate (LSE) method.

IV.3.2.2 Backward Pass

During the backward pass Figure IV.4, the error signals are propagated backward through the network. In this phase, the premise parameters are updated using the gradient descent algorithm, while the consequent parameters remain fixed [6].

Thus:

- S1 (premise parameters) are adapted using backpropagation,
- S2 (consequent parameters) are held constant during this stage.

Table IV.1 ANFIS parameters during the learning process.

Learning Phase	*Premise Parameters*	*Consequent Parameters*
Forward Pass	Fixed	Updated using least squre
Backward Pass	Updated using back propagation	fixed

IV.4 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a population-based optimization algorithm inspired by the collective behavior of social organisms such as bird flocks, fish schools, and insect swarms. It was first introduced by Kennedy and Eberhart (1995) as part of the emerging field of swarm intelligence, which studies how simple agents interacting locally can collectively exhibit complex global behavior.

PSO is particularly valued for its simplicity, robustness, and effectiveness in handling nonlinear, multidimensional, and non-differentiable optimization problems. Over the years, it has been successfully applied in areas such as control system design, neural network training, and fuzzy logic optimization.

IV.4.2 The algorithm principle

In PSO, a population of candidate solutions called particles explores the search space to locate the global optimum. Each particle represents a potential solution and is characterized by two main parameters:

- **Position (xi)**, which represents the current solution;
- **Velocity (vi)**, which determines the particle's movement through the search space.

Each particle adjusts its trajectory based on both individual experience (the best solution it has found so far, called personal best pi) and social experience (the best solution found by the entire swarm, called global best g). This combination enables a balance between exploration (searching new regions) and exploitation (refining known good regions).

IV.4.3 Mathematical Model

The position and velocity of each particle are updated iteratively according to the following equations:

$$v_i(t+1) = w_i v_i(t) + [p_i - x_i(t)]c_1 r_1 + [g - x_i(t)]c_2 r_2 \tag{IV.13}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{IV.14}$$

where:

- vi(t) is the velocity of particle i at iteration t;
- xi(t) is its position;
- pi is the particle's best-known position;
- gis the global best position found by the swarm;
- wis the inertia weight controlling the trade-off between exploration and exploitation;
- c1 and c2 are the cognitive and social acceleration coefficients;
- r_1 and r_2 are random numbers uniformly distributed in the interval [0, 1].

The algorithm continues updating these parameters until a stopping criterion such as a maximum number of iterations or convergence threshold is met.

IV.5 The Hybrid Control Strategy Design

The coaxial octorotor model is divided into four main subsystems: altitude, yaw, pitch, and roll. These subsystems operate within a coupled, multi-loop control structure, where the **x** and **y** axes form the outer loops and the pitch and roll dynamics act as the inner loops.The control strategy adopted in this work combines FSMC and ANFIS to handle the different subsystems. FSMC is applied to the altitude and yaw channels, while ANFIS is responsible for controlling pitch and roll. The outer-loop position controllers for the **x** and **y** axes use standard PID control.Figure IV.5 provides an overview of the complete control architecture, and each component of the controller is detailed in the following subsections.

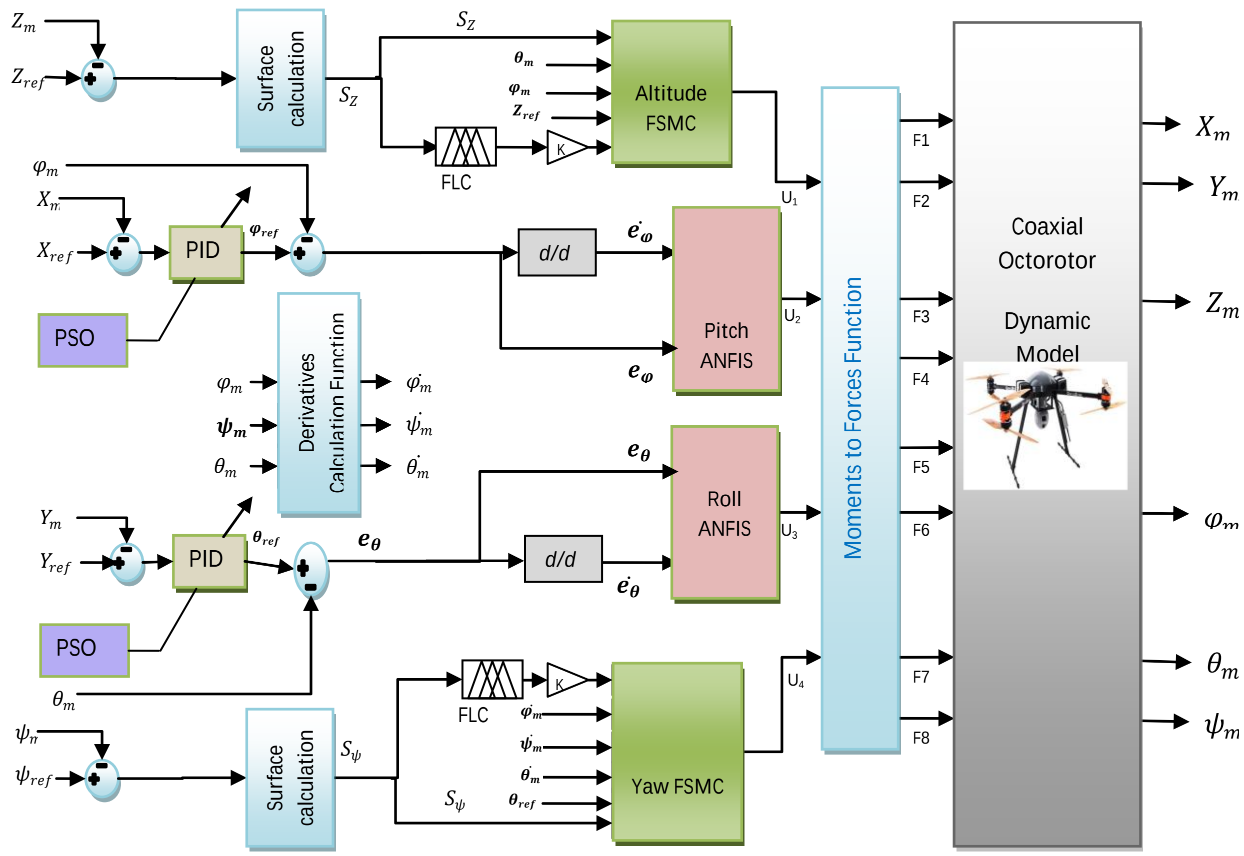


Figure IV.5 Overall control methodology.

IV.5.1 The FSMC controller

The fuzzy sliding mode controller is implemented for both the altitude and yaw subsystems. In this approach, a conventional sliding mode controller is combined with a Type-1 Mamdani fuzzy logic controller, which acts as the switching signal generator. The fuzzy component replaces the traditional discontinuous switching action with a smoother, more adaptive signal, reducing chattering while preserving the robustness of sliding mode control. This integration allows the controller to better handle nonlinearities and external disturbances, which are particularly significant in coaxial octorotor dynamics.

IV.5.1.1 Sliding surface

For the altitude (z) and yaw (ψ) dynamics of the coaxial octorotor, the sliding surface is defined as:

$$e_z = Z_d - Z_m \qquad \text{(IV.15)}$$

$$S_Z = \dot{e}_Z + \lambda_Z e_Z \qquad \text{(IV.16)}$$

$$e_\psi = \psi_d - \psi_m \qquad \text{(IV.17)}$$

$$S_{\psi} = e^{\cdot}_{\psi} + \lambda_{\psi} e_{\psi} \tag{IV.18}$$

The control objective is to drive s(t) → 0, ensuring error convergence.

IV.5.1.2 Control law

The standard SMC control law is:

$$u(t) = u_{eq}(t) + u_{sw}(t) \tag{IV.19}$$

With

- Equivalent control:

$$u_{eq}(t) = f(x) + \frac{1}{g(x)}(r(t) + \lambda \dot{e}(t)) \tag{IV.20}$$

- Switching control:

$$u_{sw}(t) = -K.sign(s(t)) \tag{IV.21}$$

Where K > 0, ensures robustness.

The discontinuous term sign(s) may cause the chattering effect that cause actuators saturation or damage which an important part in the coaxial octorotor.

IV.5.1.3 Fuzzy logic for chattering reduction

The type 1 mamdani FLC (fuzzy logic controller) which is used as switching signal generator to eliminate the chattering effect is a SISO (single input single output) controller, where the surface S is the input and the switching signal Usw is the output. Figures IV.6 and IV.7 presents the input / output membership functions where table 1 shows the rule base.

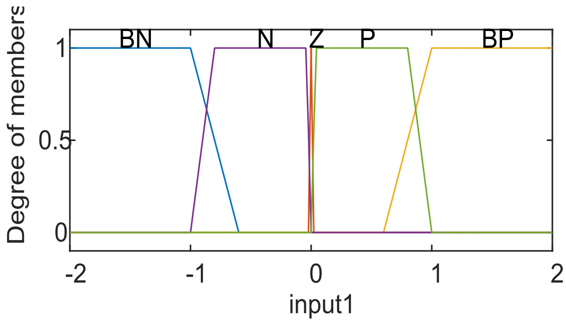


Figure IV.6 Input membership functions

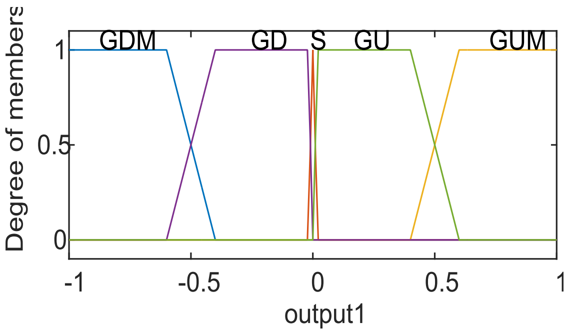


Figure IV.7 Output membership functions

Table IV.2 FLC rule base

Input	BN	N	Z	P	BP
Output	GDM	GD	S	GU	GUM

This produces continuous approximation for switching control as:

$$u_{sw}(t) = -K.\mu(s) \tag{IV.22}$$

Where $\mu(s)$ is the fuzzy output.

IV.5.1.4 FSMC controller design

Assumption that: $-\frac{\pi}{2} < \varphi \leq \frac{\pi}{2}, -\frac{\pi}{2} < \theta \leq \frac{\pi}{2}, -\pi < \psi \leq \pi$

The altitude and yaw subsystems dynamics defined as:

$$\ddot{z} = \frac{\cos\varphi \cos\theta}{m} u_1 - \frac{K_3}{m}\dot{z} - g \tag{IV.23}$$

$$\ddot{\psi} = \frac{(I_x - I_y)}{I_z}\dot{\theta}\dot{\varphi} - \frac{K_6}{I_z}\dot{\psi}^2 + \frac{1}{I_z}u_4 \tag{IV.24}$$

Defining the state variables as:

$x_1 = z$ (Altitude position)

$x_2 = \dot{z}$ (Velocity)

$x_3 = \psi$ (Yaw angle)

$x_4 = \dot{\psi}$ (Yaw angular Velocity)

The dynamic model becomes as:

$$\dot{x_1} = x_2 \tag{IV.25}$$

$$\dot{x_2} = \frac{\cos\varphi \cos\theta}{m} u_1 - \frac{K_3}{m} x_2 - g \tag{IV.26}$$

$$\dot{x_3} = x_4 \tag{IV.27}$$

$$\dot{x_4} = \frac{(I_x - I_y)}{I_z}\dot{\theta}\dot{\varphi} - \frac{K_6}{I_z} {x_3}^2 + \frac{1}{I_z}u_4 \tag{IV.28}$$

Applying error and sliding surface defined in (IV.12), (IV.13), (IV.14) and (IV.15):

$$s_z = \dot{e_z} + \lambda_z e_z = (\dot{z}_d - x_2) + \lambda(z_d - x_1) \tag{IV.29}$$

$$s_\psi = \dot{e_\psi} + \lambda_\psi e_\psi = (\dot{\psi}_d - x_4) + \lambda(\psi_d - x_3) \tag{IV.30}$$

$\lambda_\psi, \lambda_z > 0$ designer parameters

IV.5.1.5 Lyapunov function

The Lyapunov candidateis :

$$V = \frac{1}{2}s^2 \tag{IV.31}$$

This guarantees that V > 0 for s ≠ 0 and V=0 for s = 0.

The time derivative of V:

$$\dot{v} = s\dot{s} \tag{IV.32}$$

Finding $\dot{s}$ surface time derivative using (IV.29) and (IV.30)

$$\dot{s_z} = \ddot{e_z} + \lambda_z \dot{e_z} = (\dot{z}_d - \dot{x}_2) + \lambda(\dot{z}_d - \dot{x}_1) \tag{IV.33}$$

$$\dot{s_\psi} = \ddot{e}_\psi + \lambda_\psi \dot{e}_\psi = \left(\ddot{\psi}_d - \dot{x}_4\right) + \lambda\left(\dot{\psi}_d - \dot{x}_3\right) \tag{IV.34}$$

Substitute $\dot{x}_2$ and $\dot{x}_4$from dynamic models (IV.26) and (IV.28)

$$\dot{s_z} = \ddot{e_z} + \lambda_z \dot{e_z} = \left(\dot{z}_d - \frac{\cos\varphi\cos\theta}{m}u_1 + \frac{K_3}{m}x_2 + g\right) + \lambda_z(\dot{z}_d - \dot{x}_1) \tag{IV.35}$$

$$\dot{s_\psi} = \ddot{e}_\psi + \lambda_\psi \dot{e}_\psi = \left(\ddot{\psi}_d - \frac{(I_x - I_y)}{I_z}\dot{\theta}\dot{\varphi} + \frac{K_6}{I_z}{x_3}^2 - \frac{1}{I_z}u_4\right) + \lambda_\psi\left(\dot{\psi}_d - \dot{x}_3\right) \tag{IV.36}$$

IV.5.1.6 Control law

The control law is divided into two parts equivalent and switching control as (IV.16). The u_{eq} it would maintain the $\dot{s} = 0$ in the ideal state, so we will set $\dot{s} = 0$ to derive the u_{eq} as:

$$0 = \left(\dot{z}_d - \frac{\cos\varphi\cos\theta}{m}u_{eqz} + \frac{K_3}{m}x_2 + g\right) + \lambda_z(\dot{z}_d - \dot{x}_1) \tag{IV.37}$$

$$u_{eqz} = \frac{m}{\cos\varphi\cos\theta}\left[\dot{z}_d + \frac{K_3}{m}x_2 + g + \lambda_z(\dot{z}_d - \dot{x}_1)\right] \tag{IV.38}$$

$$0 = \left(\ddot{\psi}_d - \frac{(I_x - I_y)}{I_z}\dot{\theta}\dot{\varphi} + \frac{K_6}{I_z}{x_3}^2 - \frac{1}{I_z}u_{eq\psi}\right) + \lambda_\psi\left(\dot{\psi}_d - \dot{x}_3\right) \tag{IV.39}$$

$$u_{eq\psi} = I_z\left[\ddot{\psi}_d - \frac{(I_x - I_y)}{I_z}\dot{\theta}\dot{\varphi} + \frac{K_6}{I_z}{x_3}^2 + \lambda_\psi\left(\dot{\psi}_d - \dot{x}_3\right)\right] \tag{IV.40}$$

The switching control law which the part that drives the system into the sliding surface and rejects the disturbances.

$$u_{swz} = \frac{m}{\cos\varphi\cos\theta}.K_z.\mu(s_z) \quad \text{(IV.41)}$$

$$u_{sw\psi} = I_Z.K_\psi.\mu(s_\psi) \quad \text{(IV.42)}$$

The $K_\psi, K_z > 0$ are the switching gains.

The total control laws U_1 and U_4 derived using (IV.38), (IV.40), (IV.41) and (IV.42) as:

$$u_1 = \frac{m}{\cos\varphi\cos\theta}\left[\dot{z}_d + \frac{K_3}{m}x_2 + g + \lambda_z(\dot{z}_d - \dot{x}_1) + K_z.\mu(s_z)\right] \quad \text{(IV.43)}$$

$$u_4 = I_z\left[\ddot{\psi}_d - \frac{(I_x - I_y)}{I_z}\dot{\theta}\dot{\varphi} + \frac{K_6}{I_z}{x_3}^2 + \lambda_\psi(\dot{\psi}_d - \dot{x}_3) + K_\psi.\mu(s_\psi)\right] \quad \text{(IV.44)}$$

IV.5.1.7 Stability analysis using lyapunov

Reforming the equation (IV.35) and (IV.36) using (IV.41) and (IV.42) we can have lyapunov function time derivative as:

$$\dot{s_z} = \dot{z}_d - \frac{\cos\varphi\cos\theta}{m}\left(\frac{m}{\cos\varphi\cos\theta}\left[\dot{z}_d + \frac{K_3}{m}x_2 + g + \lambda_z(\dot{z}_d - \dot{x}_1) + K_z.u(s)\right]\right) + \frac{K_3}{m}x_2 + g + \lambda_z(\dot{z}_d - \dot{x}_1) \quad \text{(IV.42)}$$

$$\dot{s_\psi} = \ddot{\psi}_d - \frac{(I_x - I_y)}{I_z}\dot{\theta}\dot{\varphi} + \frac{K_6}{I_z}{x_3}^2 - \frac{1}{I_z}I_z\left[\ddot{\psi}_d - \frac{(I_x - I_y)}{I_z}\dot{\theta}\dot{\varphi} + \frac{K_6}{I_z}{x_3}^2 + \lambda_\psi(\dot{\psi}_d - \dot{x}_3) + K_\psi.u(s)\right] + \lambda_\psi(\dot{\psi}_d - \dot{x}_3) \text{(IV.43)}$$

After simplifying we have $\dot{s}$ as.

$$\dot{s_z} = -K_z.\mu(s_z) \quad \text{(IV.45)}$$

$$\dot{s}_\psi = -K_\psi.\mu(s_\psi) \quad \text{(IV.46)}$$

Reforming (IV.29) using (IV.44) and (IV.45) we get.

$$\dot{V_z} = s_z.(-K_z.u(s_z)) = -K_z.s_z.\mu(s_z) \quad \text{(IV.47)}$$

$$\dot{V_\psi} = s_\psi.\left(-K_\psi.u(s_\psi)\right) = -K_\psi.s_\psi.\mu(s_\psi) \quad \text{(IV.48)}$$

Now we have $s.u(s) > 0$ for $s \neq 0$ and this because of the range of the fuzzy logic is [-1 1].

When s is positive large the fuzzy output is positive large +1

When s is negative large the fuzzy output is negative large -1

Therefore $\dot{V} = s.\left(-K.u(s)\right) < 0$ for all $s \neq 0$ and $\dot{V} = 0$ when $s = 0$.

This proofs that the tow systems are stable globally asymptotically

IV. 5.2 Position controllers

Conventional PID controllers are used as position controllers for x and y, which are the outer control loops for the φ and θ subsystems. The PID controllers wildly used to their implementation simplicity but they need well tuned parameters KP, KI and KD. Manual tuning or classical tuning for the PID parameters often fail in nonlinear systems and to overcome this limitation we used PSO algorithm to tune the PID parameters for its fast convergence. The PID controllers used to generate the desired φ and θ for the ANFIS controllers. The x and y errors are defined as in the equations (IV.49) and (IV.50).

IV.5.2.1 PID controller :

$$e_x = x_d - x_m \tag{IV.49}$$

$$e_y = y_d - y_m \tag{IV.50}$$

The final control equations for the outer control loops x and y derived as in (IV.51) and (IV.52).

$$\theta_d = K_p.e_y + K_i.\int e_y + K_d.\frac{de_y}{dt} \tag{IV.51}$$

$$\varphi_d = K_p.e_x + K_i.\int e_x + K_d.\frac{de_x}{dt} \tag{IV.52}$$

IV.5.2.2 Particle swarm optimization algorithm

Each particle represents a candidate PID parameter set (Kp, Ki, Kd).

-Initialization: random swarm within bounds.

- Velocity & position update:

$$v_i^{(k+1)} = wv_i^{\,k} + c_1r_1(Pbest_i - x_i^{\,k}) + c_2r_2(gbest - x_i^{\,k}) \tag{IV.53}$$

$$x_i^{(k+1)} = x_i^{\,k} + v_i^{(k+1)} \tag{IV.54}$$

- **Fitness function**

The used fitness function based on the MSE (mean square error) and overshot penality

$$MSE = (target - output)^2 \tag{IV.55}$$

$$overshoot = \max(0, (\max(target) - \max(output)) \tag{IV.56}$$

$$Fitnessfunction = MSE + 10 * overshoot^2 \tag{IV.57}$$

- **Particle swarm parameters:**

Particle number: 30.

Inertia weight ω: 0.729.

Acceleration coefficients: c1 = c2 = 1.49445.

Iterations: 100.

Bounds: Kp[0.1,5], Ki E [0.01,2], Kd E [0.001,1].

- **Offline training data**

The data that used to tune PID controllers using PSO is generated from a Mamdani fuzzy logic controller e controller in [63]. The data generated under circular path tracking task with number of samples 12001.Table IV.3 presents the resulted PID parameters for each controller. While Figures 8 and 9 shows the tuning MSE for tow PIDs.

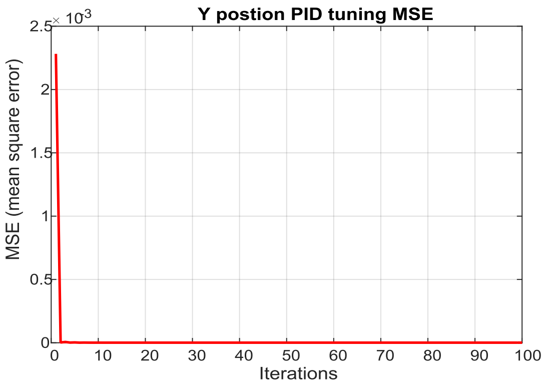


Figure IV.8 Y position PID tuning MSE

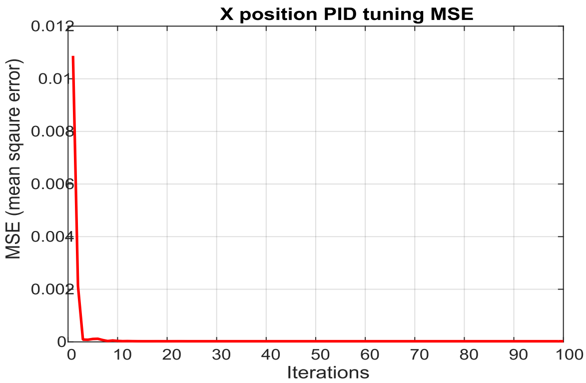


Figure IV.9 X position PID tuning MSE

Table IV.3 The best PID gains

Gains	K_p	K_i	K_d
Y controller	1.7	0.010	1
X controller	1.29	0.010	1

IV.5.3 ANFIS Controller Design

Pitch and roll subsystems of the coaxial octorotor exhibit the strongest nonlinear coupling and the fastest dynamics. An Adaptive Neuro-Fuzzy Inference System (ANFIS) is a good choice here because it combines the interpretability and rule structure of fuzzy systems with the learning and approximation capability of neural networks. Thus ANFIS provides an adaptive nonlinear compensator for the inner loops, improving tracking and disturbance rejection while remaining computationally light.

The adaptive neuro-fuzzy inference system designed basing on a Takagi-sugeno fuzzy logic controller with tow inputs the error and its time derivative. Each input has three Gaussian membership functions where the output considered as linear function.

IV.5.3.1 Training data

The training data generated from a smiluation of coaxial octorotor in circular path tracking task with 12001 samples. Where the controller used in this task is type 1 mamdani fuzzy logic controller applied to all subsystems, this FLC proofed its stability and robustness under external disturbances in paper [63]. Figure IV.10 presents the training data.

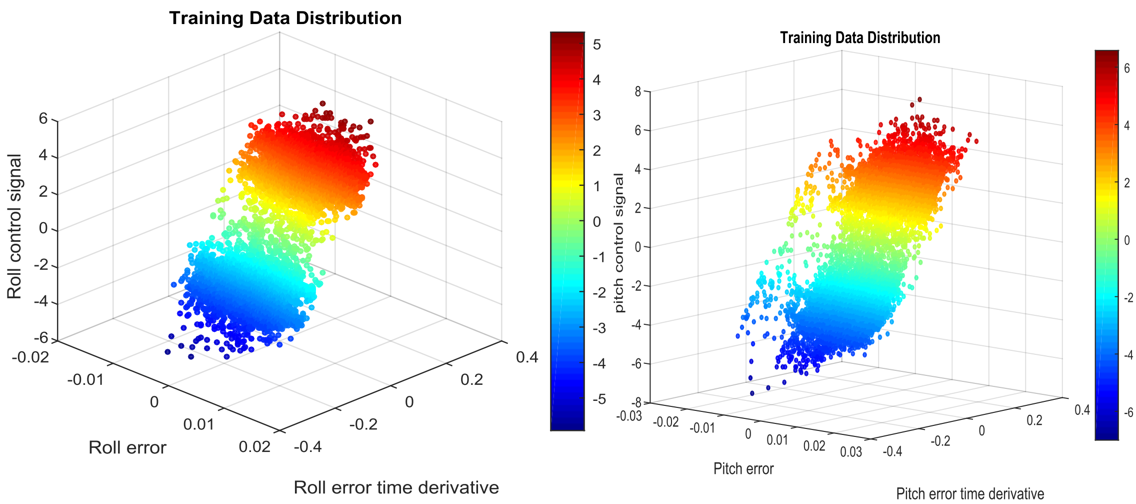


Figure IV.10 Training data

IV.5.3.2 Takagi sugeno generation using grid partion

Figures IV.11, IV.12, IV.13 and IV.14 show the input membership functions for both input variables for the subsystems φ and θ while figures IV.15 and IV.16 show the training error for both subsystems. Figures IV.17 and IV.18 shows the output error for both subsystems φ and θ.

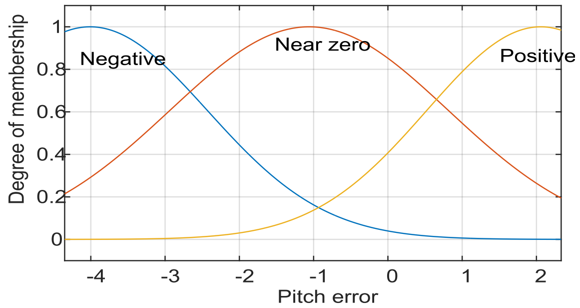


Figure IV.11 Pitch controller input error MFs

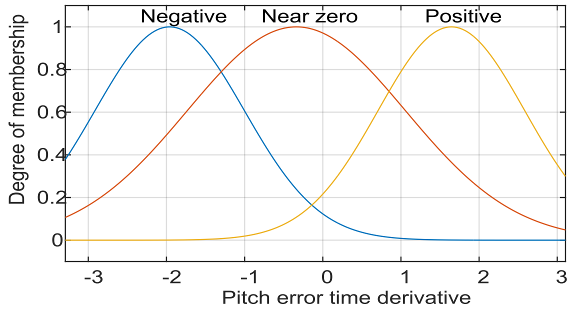


Figure IV.12 Pitch controller error time derivative MFs

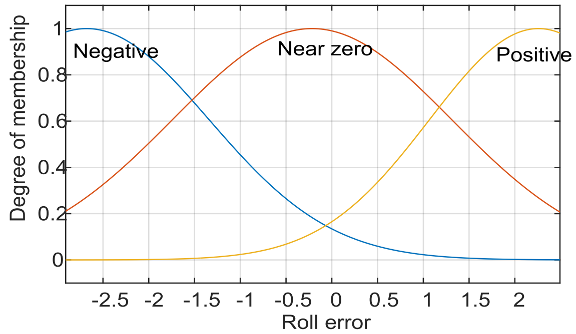


Figure IV.13 Roll controller input error MFs

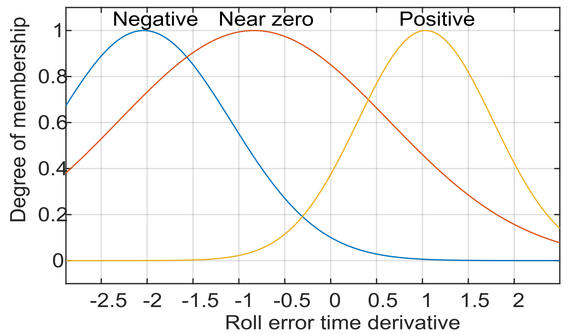


Figure IV.14 Roll controller error time derivative Mfs

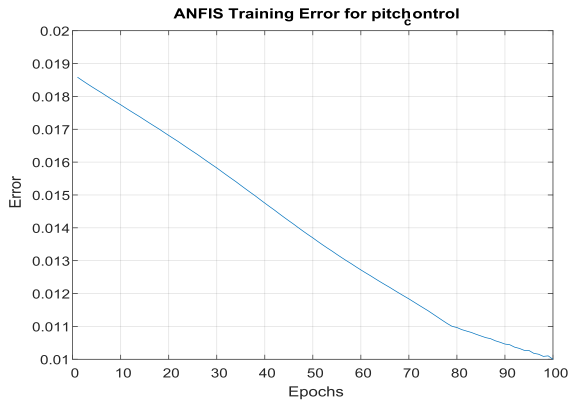


Figure IV.15 Pitch subsystem ANFIS training error

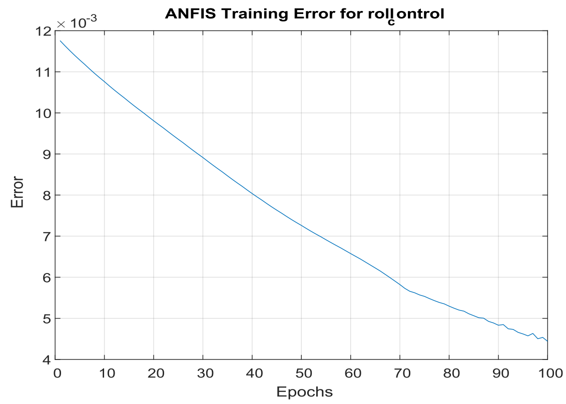


Figure IV.16 Roll subsystem ANFIS training error

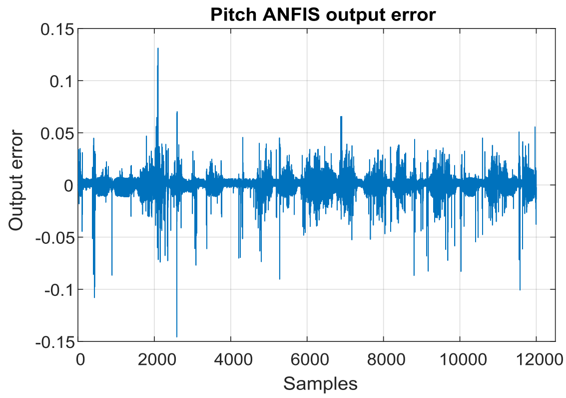


Figure IV.17 Pitch subsystem ANFIS output error

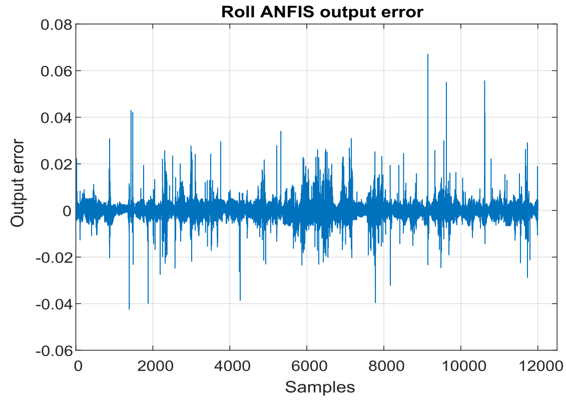


Figure IV.18 Roll subsystem ANFIS output error

From figures IV.15 to IV.18 we can see that the ANFIS controller has been trained well and it give a remarkable training error as figure IV.15 and IV.16 for both subsystems. Also the ANFIS controller give small error compared to the target data for both subsystems as figures IV.17 and IV.18. The resulted rules surface for the trained sugeno FLCs for both φ and θ shown in figures IV.19 and IV.20 respectively.

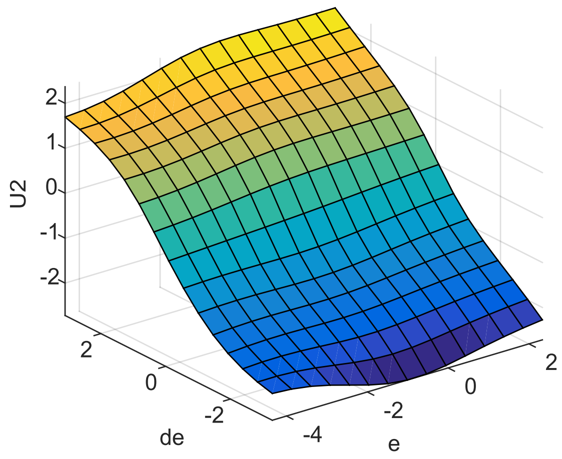


Figure IV.19 Pitch ANFIS controller surface

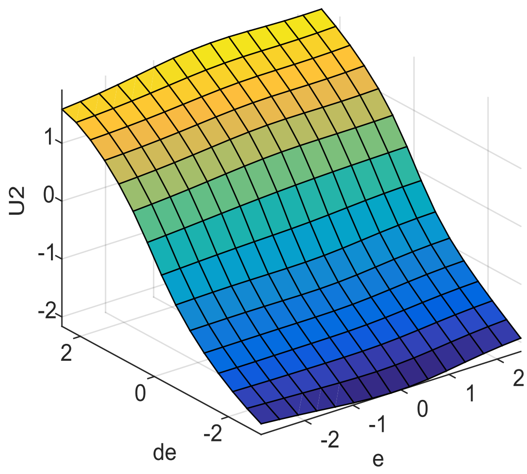


Figure IV.20 Roll ANFIS controller surface

IV.6 Simulation results and discussion

The proposed control approach is evaluated in a 6 DOF coaxial octorotor dynamic model designed in MATLAB/Simulink. A solar farm surveillance trajectory scenario was applied to the controller. Also a robustness test is done to the controllers and a comparison using performance indices between the proposed controller and the FLC controller in [63]. Table IV.4 shows the simulation parameters.

Table IV.4 Dynamic model parameters [55]

Parameter	Simulated Value
b	$2.9842 \times 10^{-5}\mathrm{N}.\frac{\mathrm{m}}{\mathrm{rad}}/\mathrm{s}$
d	$2.2320 \times 10^{-7}\mathrm{N}.\frac{\mathrm{m}}{\mathrm{rad}}/\mathrm{s}$
K_1, K_2, K_3	$(5.5670, 5.5670, 6.3540) \times 10^{-2}\mathrm{N}.\frac{\mathrm{m}}{\mathrm{rad}}/\mathrm{s}$
K_4, K_5, K_6	$(5.5670, 5.5670, 6.3540) \times 10^{-2}\mathrm{N}.\frac{\mathrm{m}}{\mathrm{rad}}/\mathrm{s}$
J_r	$2.8385 \times 10^{-5}\mathrm{N}.\frac{\mathrm{m}}{\mathrm{rad}}/\mathrm{s}^2$
I_x, I_y, I_z	$(4.2, 4.2, 7.5) \times 10^{-2}\mathrm{N}.\frac{\mathrm{m}}{\mathrm{rad}}/\mathrm{s}^2$
m	1.6 Kg
l	0.23 m

IV.6.1 Solar farm surveillance task

Monitoring and inspecting solar farms is one of the key applications where UAVs prove highly valuable. For this task, a solar-farm surveillance trajectory was generated in MATLAB, set at an altitude of 18 meters and covering a farm area 50 meters wide and 100 meters long. This path represents a typical inspection route that a UAV would follow during real operational conditions. Figures IV.21, IV.22, and IV.23 illustrate how effectively the proposed controllers guide the coaxial octorotor along this trajectory, demonstrating their capability to handle the surveillance mission

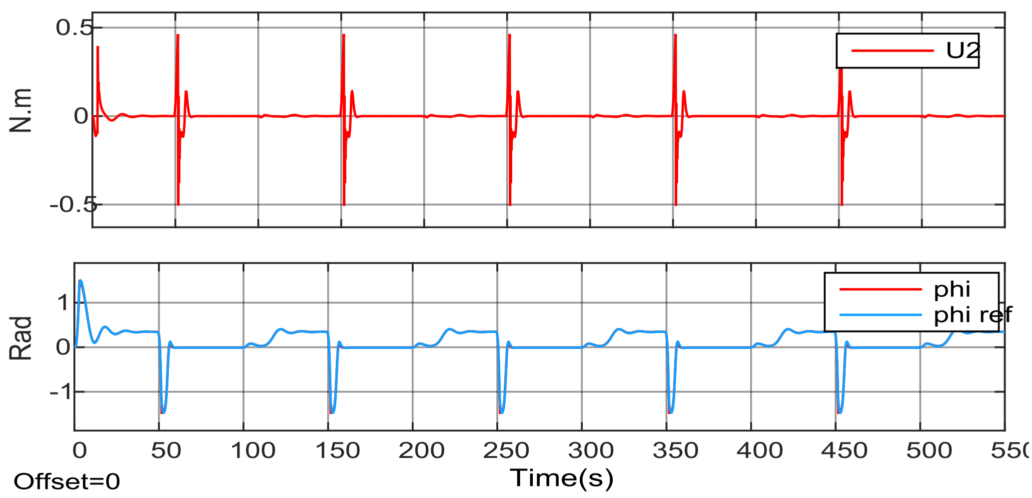


Figure IV.21 Pitch ANFIS controller response

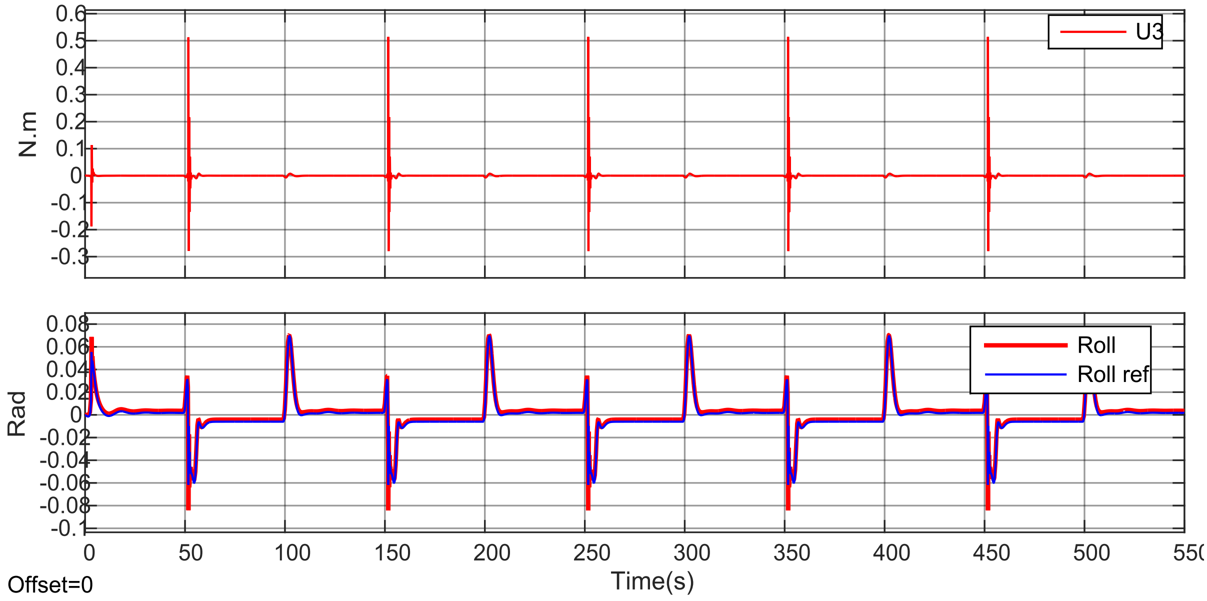


Figure IV.22 Roll ANFIS controller response

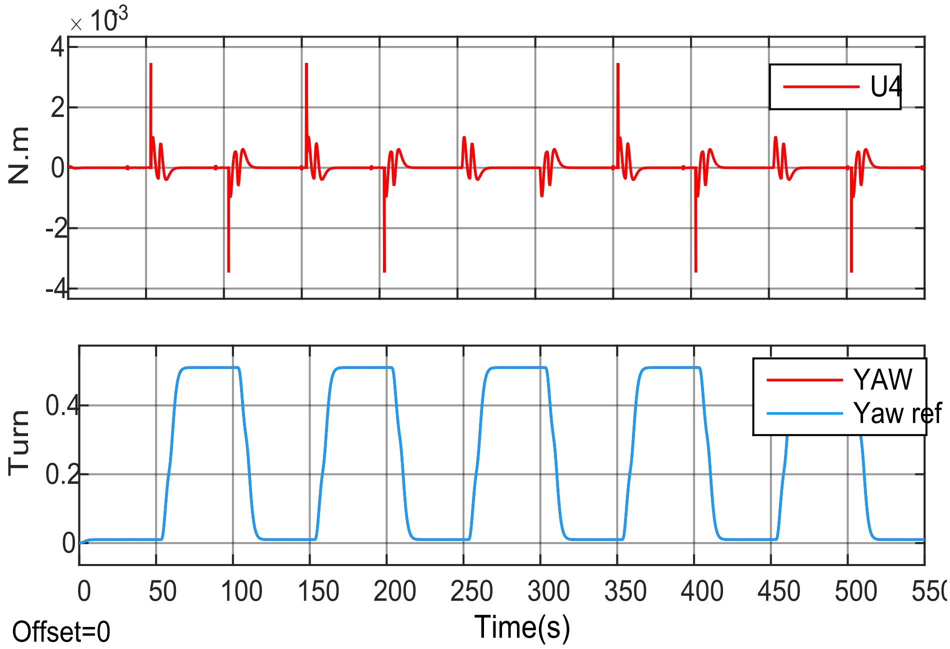


Figure IV.23 Yaw FSMC controller response

The full trajectory tracking task is shown in the figure IV.24 which proofs the efficiency of the proposed controllers in term of error tracking.

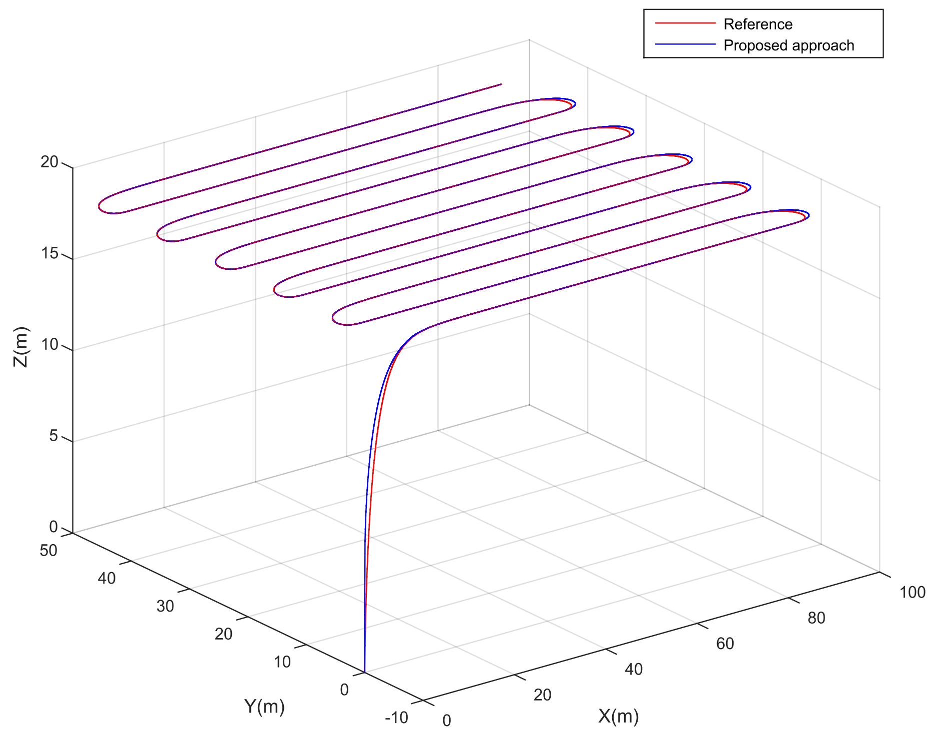


Figure IV.24 Solar farm surveillance task trajectory

IV.6.2 Robustness test

As robustness test in order to verify the effectiveness of the proposed control strategy, a wind simulation effects are applied on the proposed control scheme. The wind applied to angles φ, θ and Ψ and to the position as a trajectory drift for the x, y and z. Figures IV.25 to IV.26 shows response of the controller to the wind disturbances which the ANFIS controller responded well to the deviation of 0.2 rad in the angles pitch and roll in figures IV.27 and IV.28, also the FSMC responded well to a deviation of 0.2 turn in figure IV.28.

In figures IV.25 and IV.26 we can remark that the controller recovers the tracking of the desired trajectory after a time period of 90 second and that returns to the disturbance applied to position subsystems x and y. This high coupling between the angle and position parameters because that the x and y subsystems are outer loops and the angles φ and θ are the inner loops.

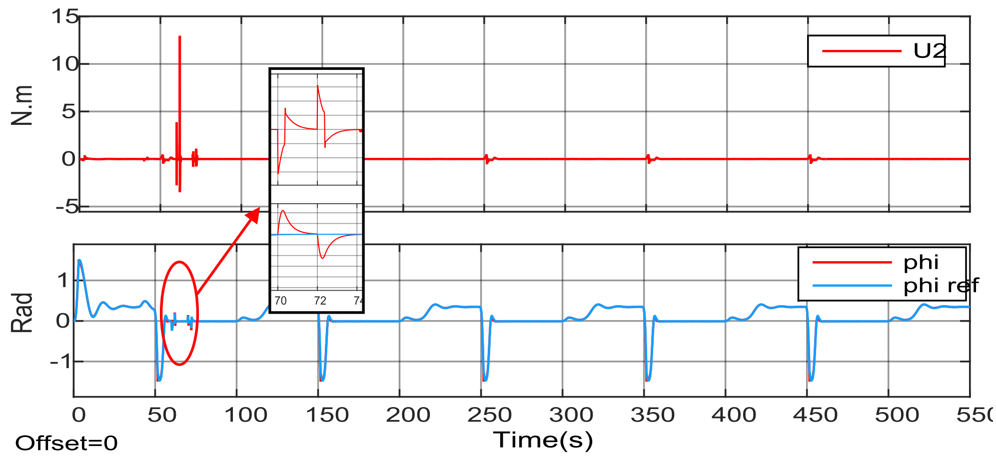


Figure IV.25 Disturbance Pitch ANFIS response

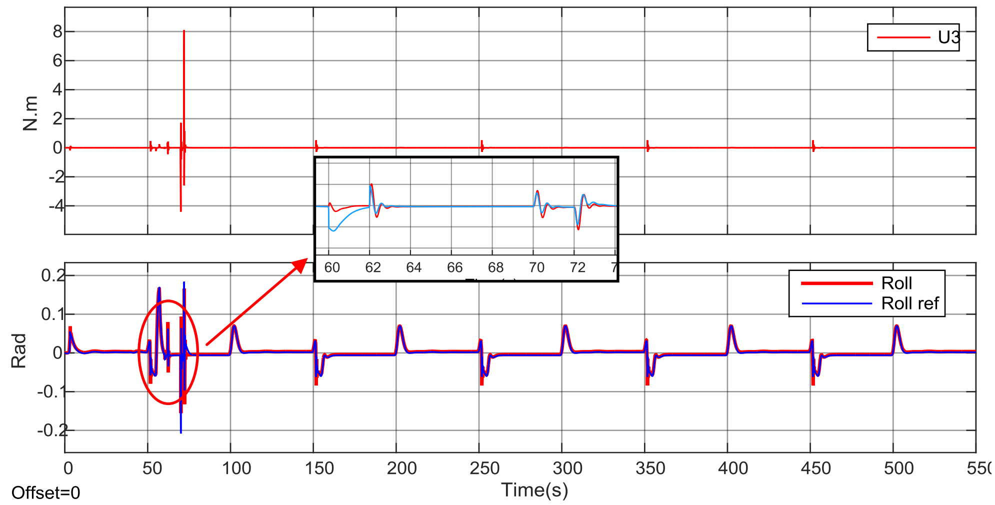


Figure IV.26 Disturbance Roll ANFIS response

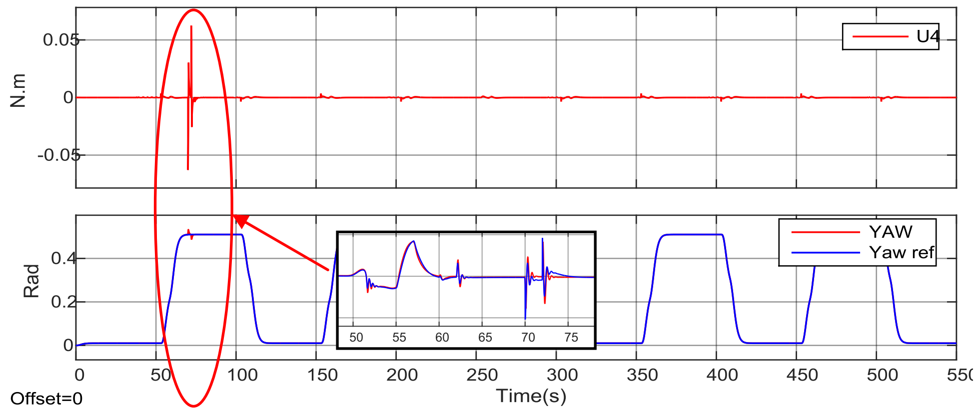


Figure IV.27 Disturbane yaw FSMC response

The external disturbances in position subsystems x, y and z is shown in figure IV.28 which is a full trajectory under disturbances. We can remark that the controller responded well to the disturbances where its keep tracking the desired trajectory.

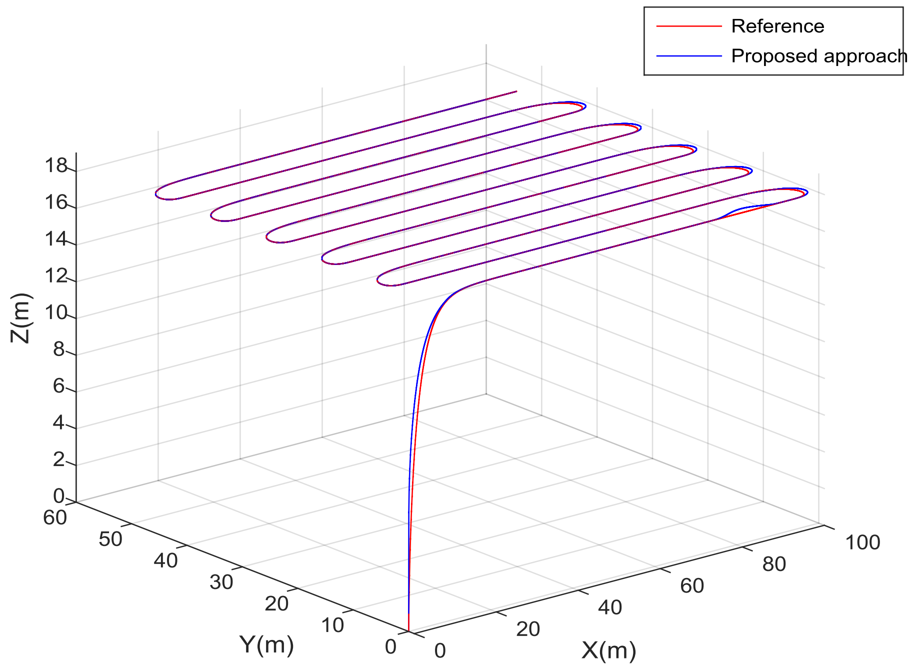


Figure IV.28 Disturbances in the trajectory

IV.6.3 Comparisonof Performances

A comparison study is investigated to this proposed hybrid control approach. The objective is show the effctivemess of this control strategy with the standard fuzzy logic controller proposed in paper [63]. The comparison tets is realized using the usually known creterians as error performance indices: IAE (integral absolute error), ITSE (integral time squre error) and the control effort (CE) under rectangular path tracking task. The following equations: (IV.58),(IV.59),(IV.60) describes these indices and the control effort.

$$\mathrm{IAE}= \sum_{k=0}^{N}|e(k.T_s)| \quad \text{(IV.58)}$$

$$ITSE = \sum_{K=0}^{N} kT_s.(e(k.T_s))^2 \quad \text{(IV.59)}$$

$$\mathrm{CE}= \sum_{i=1}^{N} U_i^{\,2} \quad \text{(IV.60)}$$

where:

e: Error between the two responses

K: discrete time step

Ts: sampling time

The comparison results are presented in tables IV.5 and IV.6, while the results in the table IV.5 prsentes the IAE and ITSE comparision between the FLC and the proposed hybrid approach, the results shows that the hybrid aproach has the superioir performance than the FLC in the subsystems Z, yaw, pitch and roll and that returns to the use of FSMC controllers in the z and yaw and the ANFIS controllers in the subsystems angles φ and θ, in the other hand the PID used in the x and y subsytems give a lower performance than the FLC. As an overal result the FSMC ANFIS hybride appproache has the superior performance than the FLC in the error performance indices.

The results in table IV.6 gives the comparision in the control effort parameter and we can remark easily that the proposed hybrid approach had the superior performance than the FLC, this superiority make the coaxial octorotor that controlled using this hybrid approacheflys longer distance and more time than the FLC. Figure IV.29 shows the superiority of the proposed hybrid appraoche than the FLC controller in a rectangular trajectory we can remark that the green line which for the hybrid approach is the nearest trajectory to the reference trajectory in red.

Table IV.5 IAE and ITSE comparision

	The hybrid approach		*FLC*	
	IAE	*ITSE*	*IAE*	*ITSE*
X	*0.41247*	*0.20507*	***0.24007***	***0.060747***
Y	*0.15391*	*0.015028*	***0.11169***	***0.0089207***
Z	***0.011738***	***6.75e-05***	*4.6739*	*11*
Pitch	***0.24974***	***0.032777***	*0.34995*	*0.093469*
Roll	***0.22554***	***0.026065***	*0.2303*	*0.037977*
Yaw	***0.00014652***	***3.8467e-0***	*2.0116*	*2.062*

Table IV.6 Control effort comparison

	FSMC_ANFIS	*FLC*
U1	*29,570*	***29,569.84***
U2	***0.0011678***	*4,449.50*
U3	***7.3699e-05***	*4,490.38*
U4	***2.9036e-06***	*332.10*
Total Control effort	***29570.4566***	*38838.6581*

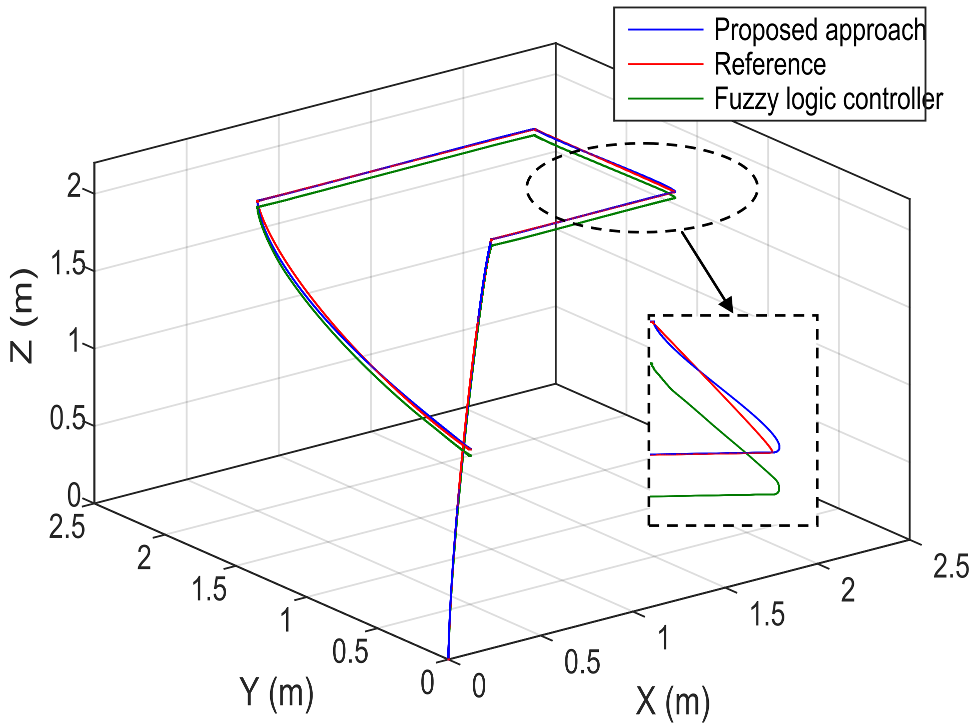


Figure IV.29 FLC vs the hybrid approach under rectangular path

IV.7 Conclusion

This chapter was dedicated to presenting, analyzing, and validating a sophisticated hybrid control strategy for a coaxial octorotor UAV. The core objective was to address the complex, nonlinear, and coupled dynamics of the system by intelligently combining several advanced control techniques.

We began by laying the theoretical groundwork for the main components: the Adaptive Neuro-Fuzzy Inference System (ANFIS), which merges the learning power of neural networks with the interpretability of fuzzy logic; the Fuzzy Sliding Mode Control (FSMC), which enhances robust sliding mode control by using a fuzzy system to suppress harmful chattering; and Particle Swarm Optimization (PSO), used to automatically and efficiently tune the parameters of conventional PID controllers.

The heart of the chapter was the design of the hybrid control architecture itself. Recognizing the different characteristics of the octorotor's subsystems, we allocated the most suitable controller to each technique such as :

- FSMC was deployed for the altitude and yaw channels, leveraging its inherent robustness to handle uncertainties and external disturbances effectively while eliminating chattering.

- ANFIS was chosen to control the pitch and roll anglesthe fastest and most tightly coupled dynamics. Its ability to learn and adapt from data made it ideal for compensating for nonlinearities and providing precise, adaptive inner-loop control.
- PSO-optimized PID controllers were used for the outer position (x, y) loops, where PSO ensured an optimal set of gains that would be difficult to find manually, improving the overall tracking performance.

The proposed strategy was then put to the test through comprehensive MATLAB/Simulink simulations. First, in a realistic solar farm surveillance scenario, the controllers demonstrated excellent trajectory tracking capabilities. More importantly, a dedicated robustness test under simulated wind disturbances proved that the hybrid system could recover stable flight and continue tracking after significant perturbations, a critical requirement for real-world UAV operation.

Finally, a quantitative performance comparison against a standard Fuzzy Logic Controller (FLC) benchmark underscored the superiority of our approach. The hybrid FSMC-

ANFIS-PID strategy achieved lower tracking errors (IAE, ITSE) in most subsystems and, crucially, a significantly reduced control effort. This latter point is not just a numerical advantage; it translates directly into practical benefits like longer flight times and reduced actuator wear for the octorotor.

In summary, this chapter successfully demonstrated that by synergistically combining ANFIS, FSMC, and PSO-optimized PID into a single, cohesive framework, we can create a control system that is not only highly performant and robust but also intelligently adapted to the distinct challenges posed by each axis of a complex aerial vehicle.

General Conclusion

This thesis has contributed to the advancement of autonomous aerial robotics by proposing a coherent and comprehensive framework for the modeling and intelligent control of a coaxial octorotor unmanned aerial vehicle (UAV). By following a structured progression from rigorous dynamic modeling to the development of advanced hybrid control architectures this work demonstrates that the integration of computational intelligence techniques is essential for effectively addressing the complex, nonlinear, and strongly coupled dynamics characteristic of modern multirotor platforms.

The research began in Chapter I with the systematic derivation of a complete six-degree-of-freedom (6-DOF) nonlinear dynamic model based on the Newton–Euler formulation. The proposed model explicitly incorporates key physical phenomena, including gyroscopic effects, aerodynamic drag, and rotor dynamics. Its implementation and validation in the MATLAB/Simulink environment provided a high-fidelity representation of the octorotor behavior, forming a reliable foundation for the design, simulation, and evaluation of all subsequent control strategies developed in this thesis.

Building on this foundation, Chapter II addressed the control challenges arising from the system's pronounced nonlinearities and underactuated nature. A Type-1 Mamdani fuzzy logic controller (FLC) was designed and evaluated, demonstrating the effectiveness of rule-based and expert-inspired control approaches for aerial systems. The obtained results confirmed that fuzzy control can achieve stable hovering and satisfactory trajectory tracking without requiring precise linearization, highlighting its suitability for UAV applications that demand adaptability, smooth control action, and robustness to modeling uncertainties.

In Chapter III, the study progressed toward more advanced nonlinear control solutions through a comparative analysis of three hybrid controllers: a Sliding Mode PID (SMC-PID), a Fuzzy PD controller (FPDC), and a Fuzzy Sliding Mode Controller (FSMC). This analysis revealed the inherent trade-offs of each approach. While the SMC-PID exhibited strong robustness at the expense of chattering, and the FPDC ensured smooth control with reduced robustness under disturbances, the FSMC successfully reconciled these limitations. By embedding fuzzy inference within the sliding mode framework, the FSMC effectively suppressed chattering while maintaining robustness and finite-time convergence. Its superior performance across multiple tracking and error-based performance indices, as well as its enhanced disturbance rejection capability, established it as the most effective controller among the evaluated strategies.

The research reached its culmination in Chapter IV with the development of an optimized, multi-layered hybrid control architecture integrating FSMC, ANFIS, PSO, and PID controllers. This architecture represents the most advanced contribution of the thesis, employing a task-oriented allocation of control techniques: FSMC for robust altitude and yaw regulation, ANFIS for adaptive and data-driven control of the highly coupled roll and pitch dynamics, and PSO-optimized PID controllers for the outer position control loops. When evaluated in a realistic solar farm inspection scenario and subjected to sustained wind disturbances, this hybrid strategy significantly outperformed the baseline fuzzy controller. It achieved high-precision trajectory tracking, rapid disturbance recovery, and notably reduced overall control effort an important factor for extending flight endurance and minimizing actuator wear in practical UAV deployments.

In summary, this thesis demonstrates that the evolution from classical modeling techniques toward intelligent and hybrid control architectures is a fundamental requirement for achieving reliable autonomy in next-generation UAV systems. The principal contribution lies in establishing that hybrid control paradigms combining the robustness of variable structure control [8], the interpretability of fuzzy logic [7], and the adaptability of neuro-learning systems [6] offer a powerful and generalizable solution for controlling coaxial octorotors in uncertain and dynamic environments [27]. The proposed FSMC and the final FSMC–ANFIS–PSO–PID architecture provide a solid foundation for future research, particularly in the areas of adaptive learning-based control and experimental validation on real aerial platforms.

The proposed FSMC and the final FSMC–ANFIS–PSO–PID architecture provide a solid foundation for future research where the future work will focus on Hardware-in-the-Loop (HIL) validation and experimental flight testing on a physical coaxial octorotor platform to further assess the practical feasibility and real-time performance of the proposed controllers. Additional research directions include the development of fault-tolerant and vision-based autonomous inspection systems to enhance operational reliability and autonomous mission capabilities, as well as the investigation of advanced artificial intelligence techniques, such as deep reinforcement learning and transformer-based intelligent control, to further improve adaptability, robustness, and decision-making in complex and dynamic environments.

References

[1]. Siciliano, B., Sciavicco, L., Villani, L., &Oriolo, G. (2016). Robotics: Modelling, planning and control. Springer.

[2]. Beard, R. W., & McLain, T. W. (2012). Small unmanned aircraft: Theory and practice. Princeton University Press.

[3]. Horn, B. K. P., &Schunck, B. G. (1981). Determining optical flow. Artificial Intelligence, 17(1–3), 185–203. https://doi.org/10.1016/0004-3702(81)90024-2

[4]. Lucas, B. D., & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. Proceedings of the 7th International Joint Conference on Artificial Intelligence, 674–679.

[5]. Beauchemin, S. S., & Barron, J. L. (1995). The computation of optical flow. ACM Computing Surveys, 27(3), 433–466. https://doi.org/10.1145/212094.212141

[6]. Jang, J.-S. R. (1993). ANFIS: Adaptive-network-based fuzzy inference system. IEEE Transactions on Systems, Man, and Cybernetics, 23(3), 665–685. https://doi.org/10.1109/21.256541.

[7]. Castillo, O., &Melin, P. (2021). Type-2 fuzzy logic and its applications. Springer.

[8]. Utkin, V. (2013). Sliding mode control in electro-mechanical systems (2nd ed.). CRC Press.

[9]. Li, S., & Wang, H. (2020). Sliding mode control for uncertain systems. Springer.

[10]. Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. Proceedings of the IEEE International Conference on Neural Networks, 1942–1948.

[11]. Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization: An overview. Swarm Intelligence, 1(1), 33–57. https://doi.org/10.1007/s11721-007-0002-0

[12]. Nauck, D., Klawonn, F., & Kruse, R. (1997). Foundations of neuro-fuzzy systems. John Wiley & Sons.

[13]. Mendel, J. M. (2017). Uncertain rule-based fuzzy systems: Introduction and new directions (2nd ed.). Springer.

[14]. Wang, L. X. (1997). A course in fuzzy systems and control. Prentice Hall.

[15]. Corke, P. (2017). Robotics, vision and control: Fundamental algorithms in MATLAB (2nd ed.). Springer.

[16]. Floreano, D., & Wood, R. J. (2015). Science, technology and the future of small autonomous drones. Nature, 521(7553), 460–466. https://doi.org/10.1038/nature14542

[17]. Khatib, O., Siciliano, B., &Kröger, T. (Eds.). (2021). Springer handbook of robotics (2nd ed.). Springer.

[18]. Zhang, Y., Li, S., & Yang, J. (2020). Robust control of autonomous robots under uncertainties: A survey. IEEE Access, 8, 153899–153917. https://doi.org/10.1109/ACCESS.2020.3018456

[19]. Nguyen, T. H., La, H. M., &Feil-Seifer, D. (2019). Autonomous mobile robots: A survey on perception, navigation, and learning. Robotics, 8(4), 95. https://doi.org/10.3390/robotics8040095.

[20]. Bouabdallah, S., Murrieri, P., & Siegwart, R. (2004). Design and control of an indoor micro quadrotor. Proceedings of the IEEE International Conference on Robotics and Automation, 4393–4398. https://doi.org/10.1109/ROBOT.2004.1302409

[21]. Mahony, R., Kumar, V., &Corke, P. (2012). Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. IEEE Robotics& Automation Magazine, 19(3), 20–32. https://doi.org/10.1109/MRA.2012.2206474

[22]. Pounds, P., Mahony, R., &Corke, P. (2010). Modelling and control of a large quadrotor robot. Control Engineering Practice, 18(7), 691–699. https://doi.org/10.1016/j.conengprac.2010.02.008

[23]. Alaimo, A., Artale, V., Milazzo, C., & Orlando, A. (2023). Modeling and control of multirotor UAVs: A comprehensive review. Aerospace Science and Technology, 134, 108091.https://doi.org/10.1016/j.ast.2023.108091

[24]. Slotine, J. J. E., & Li, W. (1991). Applied Nonlinear Control. Prentice Hall.

[25]. C. L. P. Chen, C. -E. Ren and T. Du, "Fuzzy Observed-Based Adaptive Consensus Tracking Control for Second-Order Multiagent Systems With Heterogeneous Nonlinear Dynamics," in IEEE Transactions on Fuzzy Systems, vol. 24, no. 4, pp. 906-915, 1 Aug. 2016, doi: 10.1109/TFUZZ.2015.2486817

[26]. Puchalski, R., &Giernacki, W. (2022). UAV Fault Detection Methods, State-of-the-Art. Drones, 6(11), 330. https://doi.org/10.3390/drones6110330

[27]. Utkin, V., Guldner, J., & Shi, J. (2009). Sliding mode control in electro-mechanical systems (2nd ed.). CRC Press.

[28]. Li, S., Wang, H., & Yu, X. (2013). Fuzzy sliding mode control for a class of nonlinear systems. IEEE Transactions on FuzzySystems, 21(6), 1079–1090. https://doi.org/10.1109/TFUZZ.2013.2254493

[29]. J. M. Anderson, and F. F. Taylor, Unmanned Aircraft Systems: Design, Development, and Deployment, New York, NY: McGraw-Hill, 2021. ISBN 978-0-470-05819-0

[30]. P. B. Sujit, E. Vino, and C. A. Tisdale, "Aerial surveying using UAVs: Capabilities and applications," Journal of UAV Applications, vol. 5, no. 3, pp. 44-55, 2020.

[31]. M. Schulz, "Fixed-wing UAVs in long-range surveillance: Performance analysis," IEEE Transactions on Aerospace and Electronic Systems, vol. 56, no. 4, pp. 2383-2391, 2020.

[32]. B. E. Olson, and K. P. Watkins, "Rotary-wing vs. fixed-wing UAVs: A comparative analysis for environmental monitoring," Environmental Monitoring Systems, vol. 12, no. 2, pp. 153-170, 2021.

[33]. Tellez-Belkotosky, P. A., Cabriales-Ramirez, L. E., Gutierrez-Martinez, M. A., &Ollervides-Vazquez, E. J. (2023). Intelligent PIV Fuzzy Navigation and Attitude Controller for an Octorotor Mini-UAV. Machines, 11(2), 266. https://doi.org/10.3390/machines11020266

[34]. D. R. Howard, and T. J. Carson, "Performance of rotary-wing UAVs in urban environments," AIAA Journal of Aerospace Computing, Information, and Communication, vol. 15, no. 2, pp. 333-345, 2021.

[35]. A. Kalantari, and M. Spenko, "Design and application of quadrotors for aerial inspection," Journal of UAV Design and Control, vol. 17, no. 3, pp. 290-308, 2019.

[36]. R. R. Singh, "Multi-rotor UAV design configurations: An overview," Journal of UAV Engineering, vol. 8, no. 1, pp. 112-120, 2020.

[37]. L. J. Morell, “Analysis of hexarotor UAVs: A study on control and stability,” Journal of UAV Dynamics, vol. 19, no. 4, pp. 444-462, 2021.

[38]. T. H. Nguyen, “Octorotor UAVs: Design and applications in aerial robotics,” IEEE Transactions on Robotics, vol. 35, no. 6, pp. 1215-1225, 2020.

[39]. A. K. Raj, and P. M. Tiwari, “Review of octorotor UAV platforms: Design and control strategies,” International Journal of UAV Applications, vol. 6, no. 2, pp. 65-78, 2021.

[40]. M. J. Calvo, and G. Parra, “Quadrotor drones: Past, present, and future applications,” UAV Systems Review, vol. 7, no. 1, pp. 11-21, 2022.

[41]. I. U. Junaid, “Payload and stability enhancement in hexarotor UAVs: A comparative study,” Journal of UAV Systems Engineering, vol. 9, no. 3, pp. 221-238, 2021.

[42]. P. Smith, “Star-shaped octorotor UAV design for increased stability,” UAV Engineering Journal, vol. 10, no. 2, pp. 99-108, 2019.

[43]. K. D. Sharma, and N. Gupta, “Coaxial rotor systems in UAVs: Performance analysis and applications,” International Journal of Aerospace Engineering, vol. 15, no. 1, pp. 333-350, 2020.

[44]. D. V. Long, “Compact coaxial octorotor designs for industrial UAVs,” IEEE Access, vol. 8, pp. 55622-55635, 2020.

[45]. A. E. Smith, “Fault-tolerant control in multi-rotor UAVs: A comprehensive review,” Journal of UAV Systems and Control, vol. 14, no. 4, pp. 555-573, 2021.

[46]. M. Saied, D. ALShamaa et al., ”Model Identification and Validation for translational movements of an Octorotor UAV”, IEEE Workshop on Research, Education and Development of Unmanned Aerial Systems. (REDUAS2015), 2015, Canc´un, Mexico. pp.102-108. hal-01263761.

[47]. Derakhshan, R.E., Danesh,M., Moosavi, H. "Disturbance observer-based model predictive control of a coaxial octorotor with variable centre of gravity". IET Control Theory Appl. 18,764–783 (2024). doi.org/10.1049/cth2.12611

[48]. **K. Toudji,** L. Cherroun, M. Nadour, "Modeling and Simulation of Coaxial Octorotor Unmanned Aerial Vehicle", 1st International Conferenc on Modern Electrical Engineering and Technology (ICMEET’22) At: Souk Ahras University, Algeria.

[49]. J. Diebel ”Representing Attitude : Euler Angles, Unit Quaternions, and Rotation Vectors”, Stanford University, 2006.

[50]. Majd S, Benjamin L, Isabelle F, Clovis F, Hassan S, Guillaume S. Fault diagnosis and faulttolerant control strategy for rotor failure in an octorotor. IEEE International Conference on Robotics and Automation. Seattle 5266-5271, 2015.

[51]. S. J. Haddadi, P. Zarafshan and Farahnaz J. Niroumand, ”Dynamics Modelling and Implementation of an Attitude Control on an Octorotor”, IEEE 28th Canadian Conference on Electrical and Computer Engineering May 3-6, 2015.

[52]. Giernacki, W. ; et al., Mathematical Modeling of the Coaxial Quadrotor Dynamics for Its Attitude and Altitude Control. Energies 2021, 14, 1232.

[53]. D. Driankov, H. Hellendoorn, and M. Reinfrank, An Introduction to Fuzzy Control. Berlin: Springer-Verlag, 1993. doi: 10.1007/978-3-662-11131-4

[54]. M. O. Efe, "Fractional order sliding mode control with reaching law for a quadrotor UAV," Asian J. Control, vol. 23, no. 5, pp. 2136-2148, 2021. doi: 10.1002/asjc.2360

[55]. Mekki, H., Djerioui, A., Zeghlache, S., &Bouguerra, A. (2018). Robust adaptive control of coaxial octorotor UAV using Type-1 and Interval Type-2 Fuzzy Logic Systems. Advances in Modelling and Analysis C, 73(4), 158–170. https://doi.org/10.18280/ama_c.730405

[56]. K. M. Passino and S. Yurkovich, "Fuzzy Control", Addison Wesley Longman, 1998.

[57]. Godjevac, J. (1999). Clear Ideas on Fuzzy Logic. Informatics Collection, Presses Polytechniques et Universitaires Romandes, Lausanne..

[58]. M.Mourad "Modelisation des Effets Electromagnetiques « Apport de la logique floue et neuro-floue » " Doctoral thesis in Science universityofBatna . 26 Juin 200

[59]. A.Mohamed Amine "Commande par PID-flou d’un simulateur d’hélicoptère " MasterthesisUniversity of Akli Mohand Oulhadj– Bouira 27/09/2017

[60]. Kh.Fares, A.Abdelmoumen "Variation de La Vitesse du Moteur à Courant Continu par PID Flou" Academic master thesisUniversity of KasdiMerbah Ouargla 13/06/2022

[61]. M.halim "commande predictive par la theorie des intervalles flous et metaheuristiques" doctoral thesis en sciences universiteof Constantine, 2015.

[62]. **<u>Toudji, K</u>**., Nadour, M., L.Cherroun, L. (2024). Fuzzy logic controllers design for the path tracking of an autonomous coaxial octorotor. Electrotehnica, Electronica, Automatica, 72(1), 39–46. https://doi.org/10.46904/eea.24.72.1.1108005

[63]. Jing, Y., Mirza, A., Sipahi, R., & Martinez-Lorenzo, J. (2023). Sliding Mode Controller with Disturbance Observer for Quadcopters: Experiments with dynamic disturbances in turbulent indoor space. Drones, 7(5), 328. https://doi.org/10.3390/drones7050328

[64]. Ye, Y., Hu, S., Zhu, X., & Sun, Z. (2024). An Improved Super-Twisting Sliding Mode Composite Control for Quadcopter UAV Formation. Machines, 12(1), 32. https://doi.org/10.3390/machines12010032.

[65]. Medjghou, A., Slimane, N., &Chafaa, K. (2024). Fuzzy Sliding Mode Control Based on Backstepping Synthesis for Unmanned Quadrotors. Advances in Electrical and Electronic Engineering, 16(2), 137–148.

[66]. Darwito, P. A., &Indayu, N. (2023). Adaptive Neuro-Fuzzy Inference System Based on Sliding Mode Control for Quadcopter Trajectory Tracking under External Disturbances. Journal of Intelligent Systems and Control, 2(1), 33–46.

[67]. Chen, W., Ding, Y., Weng, F., Liang, C., & Li, J. (2025). Global Fast Terminal Fuzzy Sliding Mode Control of Quadrotor UAV Based on RBF Neural Network. Sensors, 25(4), 1060. https://doi.org/10.3390/s25041060

[68]. Kou, X., Li, F., & Wang, Y. (2023). Robust Adaptive Control of Coaxial Octorotor UAV Using Type-1 and Interval Type-2 Fuzzy Logic Systems. AMA C, 7(3).

[69]. Lee, C. C. (1990). Fuzzy logic in control systems: Fuzzy logic controller—Part I & II. IEEE Transactions on Systems, Man, and Cybernetics, 20(2), 404–435. https://doi.org/10.1109/21.52552

[70]. Ross, T. J. (2010). Fuzzy Logic with Engineering Applications (3rd ed.). Wiley.

[71]. Mendel, J. M. (2001). Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice Hall.

[72]. Y. Shtessel, C. Edwards, L. Fridman, and A. Levant, Sliding Mode Control and Observation. New York: Birkhäuser, 2014.

[73]. H. K. Khalil, Nonlinear Systems, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2002.

[74]. S. Haykin, Neural Networks and Learning Machines, 3rd ed. Upper Saddle River, NJ: Pearson, 2009.

[75]. A. Levant, "Higher-order sliding modes, differentiation and output-feedback control," Int. J. Control, vol. 76, no. 9-10, pp. 924-941, 2003.

[76]. M. Krstic, I. Kanellakopoulos, and P. Kokotovic, Nonlinear and Adaptive Control Design. New York: Wiley, 1995.

[77]. L. Rutkowski, Computational Intelligence: Methods and Techniques. Berlin: Springer, 2008.

[78]. Yu, X. (2001). Fuzzy-based sliding mode control design for nonlinear systems. IEEE Transactions on Fuzzy Systems, 9(3), 411–420. https://doi.org/10.1109/91.928928

[79]. Ogata, K. (2010). Modern control engineering (5th ed.). Prentice Hall.

[80]. L. Cherroun, " Navigation Autonome d’un Robot Mobile par des Techniques Neuro-Floues ”, Thèse de Doctorat en Automatique, Université de Biskra, 2014.

[81]. **K. Toudji**, L. Cherroun, M. Nadour and A. Hafaifa, "Neural Network Controller Design for the Path Tracking of a Coaxial Octorotor," 2024 International Conference on Telecommunications and Intelligent Systems (ICTIS), Djelfa, Algeria, 2024, pp. 1-6, doi: 10.1109/ICTIS62692.2024.10894667.

[82]. **K. Toudji**, L. Cherroun, M. Nadour, V. Puig and A. Hafaifa, "A Comparative Study between Different Fuzzy-based Nonlinear Controllers for UAV Coaxial Octorotor”, International Journal of Fuzzy Systems (IJFS), Springer, vol. xx, no. x, pp. xx-xx, 2026.https://doi.org/10.1007/s40815-025-02219-2.

ملخص

تقدم هذه الأطروحة إطارًا متكاملًا لنمذجة والتحكم الذكي في طائرة بدون طيار بثمانية م راوح مرتبة بشكل المحوري على اربع زوايا . تم تطوير نموذج ديناميكي غير خطي بست درجات حرية باستخدام صياغة نيوتن–أويلر، والتحقق من صحته عن طريق المحاكاة، مما يوفر أساسًا موثوقًا لتطوير أنظمة التحكم. بالاعتماد على هذا النموذج، تم تصميم استراتيجيات تحكم ذكية قائمة على المنطق الضبابي، حيث أظهرت فعاليتها في التعامل مع الديناميكيات غير الخطية والمتداخلة بقوة للمركبات الجوية متعددة المراوح. ولتعزيز المتانة، تم دراسة تقنيات تحكم هجينة تجمع بين المنطق الضبابي والتحكم بالأنماط الانزلاقية، مما أدى إلى تطوير متحكم بنمط انزلاقي معزز بمتحكم ضبابي قادر على تقليل ظاهرة الاهتزاز مع الحفاظ على المتانة في مواجهة الاضطرابات و اعدادات النموذج الغير دقيقة .وتُتوِّج هذه الدراسة بهندسة استراتيجية تحكم هجينة مُحسَّنة تدمج أنظمة التحكم الضبابي المعزز بشبكات التعلم العصبي للتحكم في زوايا الاتجاه، والتحكم الانزلاقية المعزز بالمتحكم الضبابي للتحكم في الارتفاع والانحراف، إضافة إلى متحكمات PID مضبوطة الاعدادات باستخدام خوارزمية أسراب الجسيمات لتحسين تتبع الموضع. وتؤكد نتائج المحاكاة تحسن دقة التتبع، وزيادة قدرة التفوق على الاضطرابات، وتقليل جهد التحكم، مما يبرز مزايا دمج تقنيات الذكاء الاصطناعي في تحقيق التحكم الجيد في الطائرات بدون طيار خلال المهمات ذات التحكم المستقل.

الكلمات المفتاحية : مركبة طائرة غير مأهولة ذاتية التحكم، طائرة بدون طيار بثمانية م راوح مرتبة بشكل المحوري موزعة على أربع زوايا ، التحكم بنمط الانزلاق، نظام هجين ضبابي معزز بشبكات التعلم العصبي ، التحكم بالمنطق الضبابي، خوارزمية تحسين أسراب الجسيمات، تتبع المسار، المتانة.

Abstract:

This thesis presents a comprehensive framework for the modeling and intelligent control of a coaxial octorotor UAV. A nonlinear six-degree-of-freedom dynamic model is derived using the Newton–Euler formalism and validated through simulation, providing a reliable basis for control design. On this foundation, intelligent control strategies based on fuzzy logic are developed, demonstrating the effectiveness of rule-based approaches for handling nonlinear and coupled UAV dynamics. To enhance robustness, hybrid control techniques combining fuzzy logic and sliding mode control are investigated, leading to a Fuzzy Sliding Mode Controller that effectively suppresses chattering while preserving robustness against disturbances and model uncertainties.The study culminates in an optimized hybrid control architecture that integrates Adaptive Neuro-Fuzzy Inference Systems for adaptive attitude control, Fuzzy Sliding Mode Control for robust altitude and yaw regulation, and Particle Swarm Optimization–tuned PID controllers for efficient position tracking. Simulation results confirm improved tracking accuracy, disturbance rejection, and reduced control effort,highlighting the benefits of integrated computational intelligence for autonomous UAV operation.

Keywords : Unmanned areil vehicle, Coaxial octorotor, Sliding mode control, adaptive neuro fuzzy inference system, Fuzzy logic control, Particle swarm optimiziation, path tracking, robustness

Résumé:

Cette thèse présente la modélisation et le contrôle intelligent d'un drone octorotor coaxial. Un modèle dynamique non linéaire à six degrés de liberté est établi à l'aide du formalisme de Newton–Euler et validé par simulation, constituant ainsi une base fiable pour la conception des lois de commande. Sur la base de ce modèle, des stratégies de contrôle intelligent fondées sur la logique floue sont développées, démontrant l'efficacité des approches à base de règles pour traiter les dynamiques non linéaires et fortement couplées des véhicules aériens multi-rotors. Afin d'améliorer la robustesse, des techniques de commande hybrides combinant la logique floue et la commande en mode glissant sont étudiées, conduisant à un contrôleur flou en mode glissant capable de supprimer le phénomène de chattering tout en conservant une forte robustesse face aux perturbations et aux incertitudes de modélisation.Ce travail aboutit à une architecture de commande hybride optimisée intégrant des systèmes neuro-flous adaptatifs pour la régulation de l'attitude, une commande floue en mode glissant pour le contrôle robuste de l'altitude et du lacet, ainsi que des régulateurs PID optimisée par l'algorithme d'optimisation par essaim de particules pour un suivi de position efficace. Les résultats de simulation confirment une amélioration significative de la précision de suivi, de la capacité de rejet des perturbations et une réduction de l'effort de commande, mettant en évidence l'intérêt de l'intégration des techniques de l'intelligence computationnelle pour l'autonomie avancée des drones.

Mots-clés: Drone autonoume, Octorotor coaxial, Commande en mode glissant, Système d'inférence neuro-flou adaptatif, Commande par logique floue, Optimisation par essaim de particules, Suivi de trajectoire, Robustesse.