RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
ÉCOLE NATIONALE POLYTECHNIQUE


**Département d'Automatique**

# End of Studies Project Thesis

**For the attainment of an Engineer degree in Control Engineering**

---

## Modeling and Nonlinear Control of Coaxial Octorotor Drone

---

**CHICHIOU Zakaria & ALLALOU Abdelghani**

Under the supervision of **Pr. BOUDANA Djamel**, **Pr. BOUCHHIDA Ouahid**

Publicly presented and defended on **23/06/2025**

**Composition of the Jury:**

| | | |
|---|---|---|
| President: | Pr. BOUCHRIT Mohamed Sghir | ENP |
| Supervisor: | Pr. BOUDANA Djamel | ENP |
| Co-Supervisor: | Pr. BOUCHHIDA Ouahid | University of Medea |
| Examiner: | Pr. TADJINE Mohamed | ENP |

**ENP 2025**

---

Département d'Automatique / ENP, 10 Rue des Frères OUDEK, El Harrach 16200 Alger, Algérie.

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
ÉCOLE NATIONALE POLYTECHNIQUE


**Département d'Automatique**

# End of Studies Project Thesis

**For the attainment of an Engineer degree in Control Engineering**

---

## Modeling and Nonlinear Control of Coaxial Octorotor Drone

---

**CHICHIOU Zakaria & ALLALOU Abdelghani**

Under the supervision of **Pr. BOUDANA Djamel**, **Pr. BOUCHHIDA Ouahid**

Publicly presented and defended on **23/06/2025**

**Composition of the Jury:**

| | | |
|---|---|---|
| President: | Pr. BOUCHRIT Mohamed Sghir | ENP |
| Supervisor: | Pr. BOUDANA Djamel | ENP |
| Co-Supervisor: | Pr. BOUCHHIDA Ouahid | University of Medea |
| Examiner: | Pr. TADJINE Mohamed | ENP |

**ENP 2025**

---

Département d'Automatique / ENP, 10 Rue des Frères OUDEK, El Harrach 16200 Alger, Algérie.

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

**ÉCOLE NATIONALE POLYTECHNIQUE**

المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

**Département Automatique**

# Mémoire de projet de fin d'études

**Pour l'obtention du diplôme d'ingénieur d'état en Automatique**

---

## Modélisation et control nonlinéar de drone octorotor coaxial

---

**CHICHIOU Zakaria & ALLALOU Abdelghani**

Sous la direction de **Dr. BOUDANA Djamel , Pr. BOUCHHIDA Ouahid**

Présenté et soutenu publiquement le (23/06/2025)

**Composition du jury:**

| | | |
|---|---|---|
| Président: | Pr. BOUCHRIT Mohamed Sghir | ENP |
| Encadrant: | Pr. BOUDANA Djamel | ENP |
| Co-encadrant: | Pr. BOUCHHIDA Ouahid | Université de Médéa |
| Examinatrice: | Pr. TADJINE Mohamed | ENP |

ENP 2025

---

Département Automatique / ENP, 10 Rue des Frères OUDEK, El Harrach 16200 Alger Algérie.

# ملخص

يعرض هذا التقرير نمذجة والتحكم في طائرة بدون طيار من نوع "طائرة بدون طيار ثمانية المراوح بتكوين محوري مشترك"، وهي نوع من الطائرات متعددة المراوح تتميز باستقرار محسن وقدرة تحميل أعلى. أولاً، تم تطوير النموذج الديناميكي غير الخطي الكامل للطائرة، بحيث يشمل كلًّا من الحركات الانتقالية والدورانية. وبناءً على هذا النموذج، تم تصميم وتنفيذ عدة استراتيجيات تحكم متقدمة لضمان طيران مستقر وتتبع دقيق للمسار. والتحكم المباشر ، والتحكم بطريقة الانزلاق ، والتحكم بطريقة وتشمل هذه الاستراتيجيات المتحكم الكلاسيكي ولتحسين أداء هذه المتحكمات، تم ضبط مكاسبها ومعاملاتها باستخدام . التكيفي، والتحكم بالمنطق الضبابي توضح التحليلات المقارنة . والخوارزمية الجينية خوارزميتين مستوحتين من الطبيعة: خوارزمية سرب الجسيمات نقاط القوة والضعف لكل طريقة تحكم من حيث المتانة، والتقارب، ودقة التتبع. وتؤكد نتائج المحاكاة فعالية مخططات التحكم المقترحة وتُبرز أهمية التحسين الذكي في تعزيز أداء الطائرات بدون طيار.

# Résumé

Ce rapport présente la modélisation et le contrôle d'un drone octorotor coaxial, un type de véhicule aérien multirotor offrant une grande stabilité et une capacité de charge accrue. Un modèle dynamique non linéaire complet du drone est d'abord développé, intégrant les mouvements de translation et de rotation. À partir de ce modèle, plusieurs lois de commande avancées sont conçues et mises en œuvre pour assurer un vol stable et un suivi précis de trajectoire. Les méthodes de contrôle utilisées incluent le PID classique, le Backstepping, la commande par mode glissant (SMC), la commande adaptative directe, ainsi que la commande floue (fuzzy logic). Pour optimiser les performances de ces contrôleurs, leurs gains et paramètres sont ajustés à l'aide de deux algorithmes d'optimisation inspirés de la nature : l'algorithme PSO (Particle Swarm Optimization) et l'algorithme génétique (GA). Une analyse comparative est réalisée afin d'évaluer la robustesse, la précision et la rapidité de convergence de chaque méthode. Les résultats de simulation valident l'efficacité des lois de commande proposées et soulignent l'intérêt de l'optimisation intelligente pour améliorer les performances des UAV.

# Abstract

This report presents the modeling and control of a coaxial octorotor drone, a type of multirotor UAV with enhanced stability and payload capacity. First, the complete nonlinear dynamic model of the drone is developed, capturing both translational and rotational motions. Based on this model, several advanced control strategies are designed and implemented to ensure stable flight and accurate trajectory tracking. These include the classical PID controller, Backstepping, Sliding Mode Control (SMC), Adaptive Direct Control, and Fuzzy Logic Control (FLC). To optimize the performance of these controllers, their gains and parameters are tuned using two nature-inspired optimization algorithms: Particle Swarm Optimization (PSO) and the Genetic Algorithm (GA). The comparative analysis demonstrates the strengths and limitations

of each control method in terms of robustness, convergence, and tracking accuracy. Simulation results validate the effectiveness of the proposed control schemes and highlight the advantage of intelligent optimization in enhancing UAV performance.

# dedication

*I dedicate this work to my family especially my parents, for their unconditional support and encouragement throughout this journey.*
*To my friends, for their presence and motivation.*
*And to everyone who believes in me and inspires me every day to give my best.*

# Dedication

*To my guiding light who inspires me to become the best version of myself,my beloved father.*

*To my heart's compass, my source of strength, and the smile that illuminates my path,my dearest mother.*

*To my sisters and brothers who stand with me.*

*To my beloved nieces and my beloved nephews, Kosay, Ghaith, and Mehdi, You are the pride of our family,*

*To my chosen family ,the friends who have been my unwavering support throughout this journey*

*To the ALALLOU and BITIT families.*

*To every classmate and to everyone who offered even a smallest encouragement.*

Abdelghani "Ayoub"

# Acknowledgments

*First and foremost, all praises and thanks are to Allah,*
*the Most Gracious, the Most Merciful,*
*for guiding us through this journey of knowledge.*

We express our sincere gratitude to:

- **Our supervisors** BOUDANA Djamel, BOUCHHIDA Ouahid, and CHERIFI Abderrezzak
  for their expert guidance, valuable advice, and constant support
  throughout our research project.
- **The jury members** [BOUCHRIT Mohamed Sghir] and [TADJINE Mohamed]
  for their time, careful evaluation, and constructive feedback
  that helped improve our work.
- **Our friends and colleagues**
  for their encouragement and shared moments during this journey.
- **Our families**
  for their endless love, support, and prayers that gave us strength.

We thank everyone who contributed to our project,
directly or indirectly.
May Allah reward you all for your kindness.

*Zakaria, Abdelghani "Ayoub"*

# Contents

**List of Tables**

**List of Figures**

**1 State Of Art** **19**

1.1 Current Applications . . . . . 19

1.1.1 Aerial photography and cinematography: . . . . . 19

1.1.2 Delivery services: . . . . . 20

1.1.3 Industrial inspections: . . . . . 20

1.1.4 Agriculture . . . . . 21

1.1.5 Security and surveillance: . . . . . 21

1.2 Drone ComponentS: . . . . . 22

1.3 Technological Advancements . . . . . 23

1.4 Future Perspectives . . . . . 23

1.5 Conclusion . . . . . 24

**2 DYNAMICS MODELLING OF OCTOROTOR COAXIAL** **26**

2.1 The possible movements of a octorotor coaxial: . . . . . 27

2.2 The mathematical model based on euler angles . . . . . 27

2.2.1 Step1:the rotation matrix . . . . . 27

2.2.2 Step2:the translation dynamic . . . . . 28

2.2.2.1 External Forces . . . . . 28

2.2.2.2 Final translational Model in Inertial frame: . . . . . 29

2.2.3 Step3:the rotation dynamic . . . . . 29

2.2.3.1 Newton's Second Law in Rotational Form . . . . . 29

2.2.3.2 Expanding the Equation . . . . . 29

2.2.3.3 External Torques: . . . 29
2.2.3.4 Final equations in body frame: . . . 30
2.2.3.5 Final equation in Inertial Frame: . . . 30
2.2.4 the state space model of the Octorotor Coaxial: . . . 31
2.2.4.1 State-space Equations: . . . 31
2.2.4.2 State Equation . . . 31
2.2.5 Rotor Dynamics : . . . 32
2.3 System Parameters . . . 32
2.4 Consclusion: . . . 33

**3 CONTROL SYNTHESIS 34**

3.1 intelligent optimization algorithms . . . 34
3.1.1 Genetic Algorithms (GA) . . . 34
3.1.1.1 Genetic Algorithm Parameter Settings . . . 35
3.2 Adopted control strategy of Octorotor Coaxial . . . 36
3.3 PID Controller . . . 37
3.3.1 Application in Drones . . . 37
3.3.2 Why PID is Used in Drones . . . 37
3.3.3 application: . . . 38
3.3.4 PID Gains Tuning . . . 38
3.3.5 Results . . . 38
3.3.5.1 3d plot . . . 40
3.3.6 Comment: . . . 41
3.4 Second LYAPONOV (Backstepping Control) . . . 41
3.4.1 Overview of Backstepping Control . . . 41
3.4.2 Principles Of the Method . . . 42
3.4.2.1 Step by step design: . . . 42
3.4.2.2 Use LYAPONOV Theory . . . 42
3.4.2.3 Recursive Approach . . . 42
3.4.3 Summary of the objective: . . . 42
3.4.3.1 LYAPONOV function $V(x)$ : . . . 42

3.4.3.2 The derivative of the LYAPONOV function . . . . . 43
3.4.4 Application to Drone : . . . . . 43
3.4.4.1 The backstepping of the orientation subsystem . . . . . 43
3.4.4.2 The backstepping of the translation Subsystem . . . . . 44
3.4.4.2.1 The Altitude control: . . . . . 44
3.4.4.2.2 Linear $x$ and $y$ motion control: . . . . . 45
3.4.4.3 Optimized Gains Using Genetic Algorithm for Backstepping Control . . . . . 45
3.4.4.4 Results . . . . . 45
3.4.4.4.1 Control Signals: . . . . . 46
3.5 Sliding Mode Control (SMC) . . . . . 48
3.5.1 Road Map for Sliding Mode Control Design . . . . . 48
3.5.1.1 Definition of Sliding mode . . . . . 48
3.5.1.2 Definition of the Sliding Manifold . . . . . 48
3.5.1.3 Preconditions for Sliding Mode Control . . . . . 49
3.5.1.4 Control Action Before Sliding . . . . . 49
3.5.2 Application to Drone . . . . . 49
3.5.2.1 The Sliding Mode Control of the orientation subsystem . . . . . 49
3.5.2.2 The Sliding Mode Control of the Translation subsystem . . . . 50
3.5.2.2.1 The Altitude control . . . . . 50
3.5.2.2.2 Linear $x$ and $y$ motion control . . . . . 50
3.5.2.3 Optimized Gains Using Genetic Algorithm for Sliding Mode Control . . . . . 51
3.5.3 Results . . . . . 51
3.6 Fuzzy Logic Control . . . . . 53
3.6.1 Introduction to Fuzzy Logic: . . . . . 53
3.6.2 Design of the Fuzzy Controller . . . . . 53
3.6.2.1 Structure . . . . . 53

3.6.2.2 Global fuzzy logic control architecture . . . 54
3.6.2.3 Define Input Variables . . . 54
3.6.3 Definition of Membership Functions (fuzzification): . . . 55
3.6.3.1 Input Membership Functions . . . 55
3.6.3.2 Output Membership Function : . . . 56
3.6.4 Rule Base Development: . . . 57
3.6.5 Defuzzification . . . 57
3.6.6 Optimization Gains using Genetic Algorithm of Fuzzy Logic Control (FLC) . . . 57
3.6.7 Results . . . 59
3.7 Conclusion . . . 60

**4 Comparative Analysis of Control Strategies for Octorotor coaxiale Drone 62**

4.1 Nominal Performance . . . 63
4.1.1 PID controller . . . 63
4.1.2 Second LYAPONOV . . . 64
4.1.3 Sliding Mode Control . . . 64
4.1.4 Fuzzy Logic Control . . . 65
4.2 Robustness tests . . . 67
4.2.1 Input perturbation . . . 67
4.2.1.1 PID Controller . . . 67
4.2.1.2 Second LYAPONOV . . . 68
4.2.1.3 Sliding Mode Control . . . 69
4.2.1.4 Fuzzy Logic Control . . . 70
4.2.2 System Parameters Variations(Parametric robustness) . . . 72
4.2.2.1 PID Controller . . . 72
4.2.2.2 backstepping method . . . 72
4.2.2.3 Sliding mode method . . . 73
4.2.2.4 Fuzzy Logic Control . . . 74
4.3 discussion and Comparison of results . . . 76
4.4 Conclusion: . . . 76

**5 Business Model Canvas (BMC) 79**

**Bibliography 81**

# List of Tables

1.1 Drone System Components Classification . . . . . 23

2.1 Possible Movements of a octocopter and Motor Control . . . . . 27

2.2 physical parameters used in the system . . . . . 32

3.1 Optimized PID Gains using GA . . . . . 38

3.2 Gains of Backstepping control . . . . . 45

3.3 Gains of Sliding mode control . . . . . 51

3.4 Rule Base Development . . . . . 57

3.5 Gains of Fuzzy Logic control . . . . . 58

4.1 Time Response Comparison for Different Controllers . . . . . 66

4.2 Comparison of Control Methods . . . . . 76

# List of Figures


1.1 Aerial photography and cinematography . . . . . 20
1.2 Delivery services . . . . . 20
1.3 Industrial inspections . . . . . 21
1.4 Agriculture . . . . . 21
1.5 Security and surveillance . . . . . 22
1.6 Drone Component: . . . . . 22

2.1 Frame of Octorotor Coaxial . . . . . 26

3.1 The Basic Program of Genetic Algorithm . . . . . 35
3.2 Synoptic scheme of the proposed control strategy . . . . . 36
3.3 System response (PID) . . . . . 39
3.4 Control signals (PID) . . . . . 40
3.5 Position Evolution of the Octorotor in Space (PID) . . . . . 40
3.6 Control signals for 3d plot . . . . . 41
3.7 System response (Backstepping) . . . . . 46
3.8 Control signals (Backstepping) . . . . . 46
3.9 Position Evolution of the Octorotor in Space (Backstepping) . . . . . 47
3.10 Sliding surface . . . . . 48
3.11 Position Evolution of the Octorotor Coaxial in Space(SMC) . . . . . 51
3.12 System response (Sliding Mode Control (SMC)) . . . . . 52
3.13 Control signals (Sliding Mode Control) . . . . . 52
3.14 Structure of individual fuzzy controller . . . . . 54
3.15 Global fuzzy logic controller . . . . . 54
3.16 Membership function of $e$ . . . . . 55

3.17 Membership function of $\dot{e}$ . . . . . 56
3.18 Membership function of input $U$ . . . . . 56
3.19 System response (Fuzzy Logic Control(FLC)) . . . . . 59
3.20 Control signals (FLC) . . . . . 59
3.21 Position Evolution of the Octorotor in Space (FLC) . . . . . 60

4.1 System response for performances evaluations (PID) . . . . . 63
4.2 System response for performances evaluations (Backstepping) . . . . . 64
4.3 System response for performances evaluations (SMC) . . . . . 65
4.4 System response for performances evaluations (FLC) . . . . . 66
4.5 System response with input perturbation (PID) . . . . . 68
4.6 System response with input perturbation (Backstepping) . . . . . 69
4.7 System response with input perturbation (SMC) . . . . . 70
4.8 System response with input perturbation (FLC) . . . . . 71
4.9 System response with parameter variations (PID) . . . . . 72
4.10 Réponse du système avec variations de paramètres (Backstepping) . . . . . 73
4.11 System response with parameter variations (Sliding Mode Control (SMC)) . . . 74
4.12 System response with parameter variations (Fuzzy Logic Control (FLC)) . . . . 75

# Introduction

Over the past decade, drones commonly referred to as Unmanned Aerial Vehicles (UAVs) have become integral to a wide range of applications. Their ability to perform tasks with high precision, reduce human effort, and minimize risk in hazardous environments has made them indispensable tools in numerous sectors. From aerial photography, surveillance, and infrastructure inspection to agriculture, logistics, and military operations, drones are increasingly relied upon to carry out missions that are either dangerous or labor-intensive for humans.

Unmanned Aerial Vehicles (UAVs), or drones, are aircraft that operate without a pilot onboard. They can be fully autonomous, remotely controlled, or human-supervised, depending on the mission (Gortney, 2016; Gertler, 2012). Their flexibility, low cost, and integration with advanced technologies make them suitable for a wide range of applications (Jara-Olmedo et al., 2018; Patel et al., 2022).[1]

UAVs are typically lighter and more compact than manned aircraft, with optimized structures for better performance and endurance. Their design reduces fuel consumption and allows for longer missions, unaffected by pilot fatigue (Hassanalian  Abdelkefi, 2017). Wing structure and material also influence noise and vibration during flight (Karpenko  Nugaras, 2022).[1]

The success of drones in performing diverse tasks more efficiently and safely than human operators has accelerated their technological evolution. In response to this growing demand, the design and capabilities of UAVs have diversified considerably. Today, various drone configurations exist ranging from quadcopters and hexacopters to octocopters each optimized for specific mission requirements in terms of stability, maneuverability, and thrust.

Among these configurations, the coaxial octocopter represents a promising advancement, especially in fields requiring high thrust and torque. This drone type employs a coaxial rotor setup, wherein two rotors are mounted on the same axis but rotate in opposite directions. Such a configuration not only increases thrust without expanding the drone's physical footprint but also enhances stability and redundancy. As a result, coaxial octocopters are particularly well-suited for demanding applications like precision agriculture, heavy payload delivery, and autonomous logistics.

Given these emerging needs, the present work is dedicated to the modeling and control of a coaxial octocopter. The primary objective is to develop a dynamic model of this multirotor system and to design an efficient control strategy that ensures precision, minimizes control effort, and optimizes energy consumption. By addressing these challenges, this project aims to contribute to the development of next-generation UAVs that are capable, reliable, and adaptable to complex operational environments.

## Problem Statement

The coaxial octocopter has become a preferred drone configuration for missions requiring high thrust and torque, such as precision agriculture, heavy payload delivery, and industrial logistics.

However, the increased number of DC motors, along with the additional structural weight, leads to significantly higher energy consumption compared to simpler UAV configurations like quadcopters.

Furthermore, applications such as delivery and agricultural spraying demand not only high power but also a smooth and reliable dynamic response. This introduces additional challenges in terms of control design, where the system must ensure a trade-off between energy efficiency, fast response time, and robustness to disturbances.

Therefore, it becomes essential to develop advanced control strategies that can meet these performance requirements. The control system should minimize energy usage, maintain an acceptable transient response, and effectively reject disturbances—ensuring safe and efficient operation of the coaxial octocopter in real-world environments.

## Objectives

The primary objective of this work is to develop a comprehensive model of a coaxial octocopter that accurately represents its dynamic behavior under real-world conditions. This includes identifying the key physical parameters and aerodynamic effects that influence the drone's motion, with the goal of establishing a realistic and reliable mathematical model.

Following the modeling phase, the focus shifts to designing control strategies capable of ensuring fast response times, effective disturbance rejection, and optimized energy consumption. Several control methods will be implemented and tested in simulation to evaluate their performance under various operating scenarios.

A comparative analysis will then be conducted between the different control approaches, with specific emphasis on criteria such as transient response, robustness to external disturbances, control signal smoothness, and overall energy efficiency. This analysis aims to identify the most suitable control strategy for high-performance applications such as delivery and precision agriculture.

## Methodology Overview

The methodology adopted in this work follows a progressive and comparative approach to the design and evaluation of control strategies for a coaxial octocopter. The goal is to identify the most effective control method that ensures fast response time, robustness to parameter variations, and a bounded, energy-efficient control signal.

The study begins with the implementation of a basic Proportional–Integral–Derivative (PID) controller, which serves as a benchmark. Its performance is evaluated through simulation tests focusing on tracking accuracy, response time, and sensitivity to disturbances.

Recognizing the limitations of PID control, particularly its lack of performances and robustness, the next phase involves the implementation of a backstepping control method. While backstepping offers improved performance in terms of stability and nonlinear system handling, it also demonstrates sensitivity to parameter changes and dynamic variations, such as variations in payload and structural weight.

To address these shortcomings, a more robust control approach Sliding Mode Control (SMC) is introduced. This method enhances system robustness, especially under model uncertainties

and external disturbances. However, during the analysis of control signals, SMC is observed to suffer from the well-known issue of chattering.

Finally, to enhance system performance, improve robustness, and optimize energy efficiency, we applied a fuzzy logic control strategy. This advanced control approach was selected for its ability to handle system nonlinearities and uncertainties while maintaining operational efficiency.

Throughout the development process, each control strategy is rigorously tuned to determine the optimal set of parameters. The tuning process prioritizes key performance criteria such as fast transient response, disturbance rejection, bounded control signals, and minimal energy consumption.

## Structure of the Report

This report is divided into four chapters, each covering a specific part of the project.

- **Introduction**
  Presents the general context, objectives, problem statement, scope, methodology, and structure of the report.
- **Chapter 1: State Of Art**
  Reviews existing research and current applications of multirotor drones, with a focus on coaxial octorotors and control methods.
- **Chapter 2: Dynamic Model Of Octorotor Coaxial**
  Explains the mathematical modeling of the drone using Newton-Euler equations, including forces, torques, and system dynamics.
- **Chapter 3:Control System Design**
  Describes the control strategy used to stabilize and guide the drone, including controller selection, design, and tuning.
- **Chapter 4: Simulation and comparison between results**
  Presents the simulation environment, test scenarios, and results. It includes performance analysis and discussion.
- **Conclusion And Futur work**
  Summarizes the main results of the project and suggests possible improvements or extensions for future research.

# Chapter 1

# State Of Art

Over the past decade, unmanned aerial vehicles (UAVs), commonly called drones, have become an essential technology in many industries. Among the different types of drones, octorotor coaxial drones have attracted special attention due to their unique design and high performance. An octorotor drone uses eight rotors to generate lift and maintain stability, and in a coaxial system, two rotors are mounted on the same axis but spin in opposite directions. This combination brings several advantages compared to traditional multirotor designs.
The main benefit of the coaxial configuration is the ability to produce more thrust without increasing the size of the drone. By stacking rotors vertically, the drone becomes more compact while still maintaining powerful lift capabilities. This is particularly useful for operations in confined spaces or where heavy payloads must be transported. In addition, the counter-rotation of the coaxial rotors cancels out torque effects, making the drone more stable and easier to control, even in strong winds.
Today, octorotor coaxial drones are widely used in professional fields such as aerial photography, emergency response, industrial inspection, agriculture, and logistics. Their high reliability, lifting power, and flight stability make them ideal for demanding missions where safety and performance are critical.
This state of the art will review the current applications, technological developments, and future trends of octorotor coaxial drones, showing how this advanced system is shaping the future of aerial robotics.

## 1.1 Current Applications

Today, octorotor coaxial drones are widely used in different industries. Some of the main applications include:

### 1.1.1 Aerial photography and cinematography:

Drones are widely used in aerial photography and cinematography to capture high-quality images and videos from the sky. They offer new perspectives that are difficult to achieve with traditional cameras. This technology is used in movies, advertisements, weddings, and real estate to provide impressive and dynamic shots,[2].

Figure 1.1: Aerial photography and cinematography

### 1.1.2 Delivery services:

Many companies are testing drones for fast and efficient delivery of small packages. Drones can reduce delivery time, especially in crowded cities or remote areas. They are also useful for urgent deliveries like medicines or documents. This application is still developing, but it has strong potential for the future,[2].

Figure 1.2: Delivery services

### 1.1.3 Industrial inspections:

Drones help in inspecting industrial sites such as power lines, pipelines, and wind turbines. They can reach high or dangerous places without putting workers at risk. Drones also save time and reduce the cost of inspections by collecting data quickly and accurately using cameras and sensors,[2].

Figure 1.3: Industrial inspections

### 1.1.4 Agriculture

In agriculture, drones are used to monitor crops, spray fertilizers or pesticides, and collect data about soil and plant health. They help farmers to improve productivity and make better decisions. This technology is especially useful in large farms where manual checking is difficult and time-consuming,[2].

Figure 1.4: Agriculture

### 1.1.5 Security and surveillance:

Drones are used for security and surveillance in public spaces, events, borders, and private properties. They can monitor large areas in real-time and detect suspicious activities. With thermal cameras and GPS, drones offer an effective solution for increasing safety and responding quickly to incidents,[2].

Figure 1.5: Security and surveillance

Thanks to their strong lift capacity and excellent stability, they are preferred for missions that require carrying heavy equipment or flying in difficult weather conditions.

## 1.2 Drone ComponentS:

The drone is part of a complex robotic system equipped with a greater number of sensors and high-performance control boards, particularly suited for tasks that require high precision.

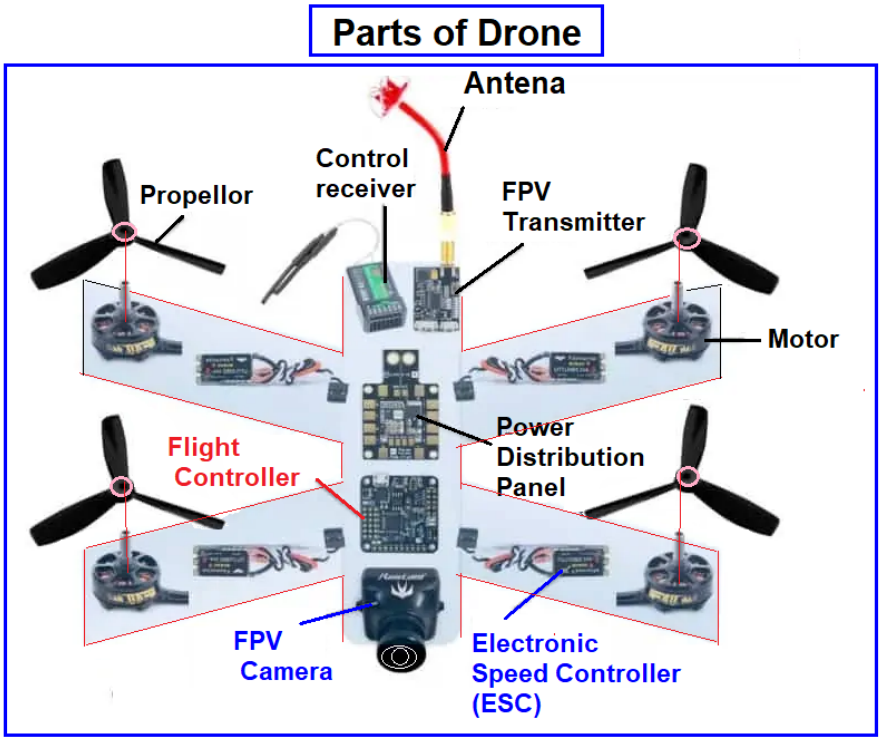



Figure 1.6: Drone Component:

| Category | Components | Function |
|---|---|---|
| Mechanical | Frame | Structural support and component housing |
| | Propellers | Thrust generation and lift |
| | Landing Gear | Safe takeoff/landing |
| Electrical | Battery | Power supply (LiPo typically) |
| | ESCs (Electronic Speed Controllers) | Motor speed regulation |
| | Brushless Motors | Propeller actuation |
| | Power Distribution Board | Current distribution |
| Electronic | Flight Controller | Brain of the drone |
| | GPS Module | Position tracking |
| | Telemetry System | Ground station communication |
| | Buzzer/LEDs | Status indication |
| | Voltage Regulator | Stable power supply |
| Sensors | IMU (Accel/Gyro) | Orientation and motion sensing |
| | Magnetometer | Heading determination |
| | Ultrasonic/Lidar | Obstacle detection |
| Payload | Camera/Gimbal | Aerial imaging |
| | Delivery Mechanism | Package transport |

Table 1.1: Drone System Components Classification

## 1.3 Technological Advancements

Recent developments have greatly improved octorotor coaxial drones. Some key improvements are:

- Use of lightweight and strong materials like carbon fiber.
- High-capacity and long-lasting batteries.
- Advanced GPS and smart flight controllers.
- Obstacle detection systems.
- Artificial intelligence for autonomous flight and decision-making.

Communication systems like 5G are also making drones faster and more reliable when exchanging data.

## 1.4 Future Perspectives

The future of octorotor coaxial drones looks very promising. Several important areas of development are expected:

- **Extended Flight Time:**
  - Development of better batteries with higher energy density.
  - Use of hybrid power systems combining electricity and fuel.
  - Lighter and stronger materials to reduce overall drone weight .
- **Improved Safety Systems:**
  - Automatic emergency landing protocols
  - Motor redundancy systems for failure protection
  - Increased resistance to extreme weather conditions
- **Greater Autonomy Through Artificial Intelligence:**
  - Real-time decision-making capabilities
  - Smart obstacle avoidance
  - Adaptation to changing weather and unexpected events during missions
- **Advanced Communication Technologies:**
  - Integration of 5G networks for faster data transmission
  - Better coordination between multiple drones and ground stations
  - Satellite communication for remote operations
- **New Applications in Various Sectors:**
  - **Medical services:** transport of urgent medical supplies
  - **Agriculture:** advanced crop monitoring and treatment
  - **Logistics:** rapid package delivery in urban areas
  - **Military:** reconnaissance and supply missions
  - **Urban Air Mobility:** potential use for passenger transport (drone taxis)
- **Challenges to Overcome:**
  - Regulation and legal restrictions
  - Privacy and security concerns
  - High development and operation costs

## 1.5 Conclusion

Octorotor coaxial drones represent an important step in the evolution of unmanned aerial vehicles. Their design, which combines strong stability, high lifting power, and compact size, allows them to complete missions that traditional drones cannot easily achieve. With the progress in materials, artificial intelligence, and communication technologies, these drones are becoming more autonomous, efficient, and reliable.

Their use in industries like emergency services, agriculture, logistics, and military operations shows their growing importance. In particular, their ability to carry heavy payloads and fly in difficult conditions gives them a major advantage.

In the future, octorotor coaxial drones are expected to play a key role in urban air transport, smart cities, and rapid delivery systems. They will not only improve existing services but also help create new solutions for modern society.

In conclusion, octorotor coaxial drones are a powerful technology that will continue to change and improve many sectors in the coming years.

# Chapter 2

# DYNAMICS MODELLING OF OCTOROTOR COAXIAL

Dynamics modelling of a system must be firstly performed before the controller design of the system,[3] Fundamentally, the Octorotor is a highly nonlinear, multivariable, strongly coupled, underactuated, and basically unstable system. It is a 6-DOFs rigid body consisting of eight brushless DC motors with attached propellers,[4].
Moreover, Figure 2.1 shows flight theory of coaxial Octorotor.

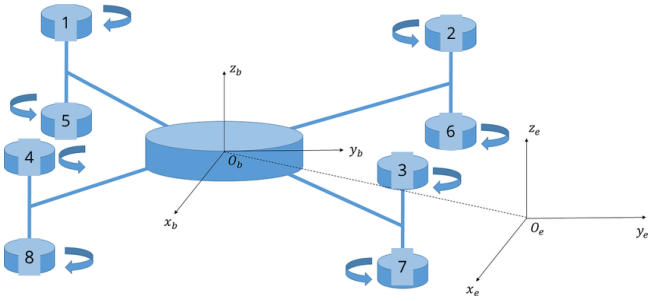



Figure 2.1: Frame of Octorotor Coaxial

These motors are fixed at the ends of a number of arms following a symmetric frame. Each motor can be controlled individually, thus modifying the attitude and the altitude of the vehicle and allowing the octorotor to move from one point to another, [4].
As shown in the Figure 2.1, $\omega_1$ , $\omega_2$ , $\omega_5$ and $\omega_6$ are rotational speed of forward direction propellers. $\omega_1$ , $\omega_5$ , $\omega_4$ and $\omega_8$ are rotational speed of right direction propellers. $\omega_3$ , $\omega_7$ , $\omega_4$ and $\omega_8$ are rotational speed of backward direction propellers, and $\omega_2$ , $\omega_6$ ,$\omega_3$ and $\omega_7$ are rotational speed of left direction propellers respectively , a coaxial octorotor presents advantages over a star-shaped one in terms of endurance and size . This configuration is adopted in this work, and will be controlled using an adapted quadrotor model,[4]. Also, the gravitational vector and movement state vector of the Octorotor can be calculated by considering the gyroscope effect,[5].
So, the dynamic equations of the Octorotor flying robot can be extracted. Indeed, dynamic equations of Octorotor can be calculated as the same with dynamic equations of Quadrotor by considering the flight concepts of Quadrotor[3]. Several methods have been proposed in the literature for the dynamic modeling of the quadrotor vehicle. the quadrotor is considered as a rigid body and its dynamics are described using the Euler-Lagrange formalism with small angles approximation. the model is derived using the Newton-Euler approach. The resultant model from both methods is the same, but written with different notations,[4].

## 2.1 The possible movements of a octorotor coaxial:

the quadrotore has four type of movements possible defined in this table:

| Movement | Description | Motor Control |
|---|---|---|
| **Lift** | The octocopter ascends vertically. All motors rotate at the same speed to generate lift. | All motors (M1, M2, M3, M4,M5 ,M6,M7,M8) increase speed equally. |
| **Descend** | The octocopter descends vertically. All motors decrease speed equally. | All motors (M1, M2, M3, M4,M5,M6,M7,M8) decrease speed equally. |
| **Yaw (Racet suivant Z)** | The octocopter rotates around its vertical axis.Motors on one diagonal rotate in the opposite direction to those on the other diagonal. | M2,M4,M5,M7 increase speed while M1,M3,M6,M8 decrease speed (or vice versa). |
| **Pitch (Tangage suivant Y)** | The octocopter tilts forward or backward. The front motors decrease speed while the rear motors increase speed (or vice versa). | M1,M2,M5,M6 increase speed M3,M4,M7,M8 decrease speed |
| **Roll (Roulis suivant X)** | The octocopter tilts to the left or right. The left motors decrease speed while the right motors increase speed (or vice versa). | M2,M3,M6,M7 increase speed , M1,M4,M5,M8 decrease speed . |

Table 2.1: Possible Movements of a octocopter and Motor Control

## 2.2 The mathematical model based on euler angles

to find the mpodel of quadrotor we devise the work in 3 steps:

### 2.2.1 Step1:the rotation matrix

We define two frames that we use to study the system's motion:

- a body-fixed reference frame $\{R_B\}$ $\{O, X, Y, Z\}$ originating at the center of mass of the the drone.
- An inartia frame $\{R_I\}$ $\{o, x, y, z\}$ fixed to the earth .

So the rotation Matrix $R_I^B$ defined the $\{R_B\}$ with respect to $\{R_I\}$ :

$$R_I^B = R_\psi \cdot R_\theta \cdot R_\phi = \begin{bmatrix} c_\theta c_\psi & -c_\phi s_\psi + s_\phi s_\theta c_\psi & s_\phi s_\psi + c_\phi s_\theta c_\psi \\ c_\theta s_\psi & c_\phi c_\psi + s_\phi s_\theta s_\psi & -s_\phi c_\psi + c_\phi s_\theta s_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix}$$

**Remarque:**

Where $\phi$ , $\theta$ and $\psi$ are the Euler angles , $S_{\phi,\theta,\psi}$ and $C_{\phi,\theta,\psi}$ refers to sinus and cosinus respectively of these angles.

### 2.2.2 Step2:the translation dynamic

According to Newton's second law of motion in the inertial frame:

$$m\ddot{X}|_I = \sum F_{\text{ext}}|_I \tag{2.1}$$

Where :

- m : dron mass
- $\ddot{X}|_I = [\ddot{x}, \ddot{y}, \ddot{z}]^t$ : acceleration in 3D space
- $\sum F_{ext}|_I$ : Total external force in inertial frame.

#### 2.2.2.1 External Forces

Two mains forces act on the drone:

1. **Thrust** $F$ Total upward force from all rotors:

$$F_p = R_I^B \cdot \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} \tag{2.2}$$

2. **Gravity** $P$ - always pulls downward :

$$P|_I = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \tag{2.3}$$

3. **disturbance** $D_z$**:**

$$D_z = \begin{bmatrix} 0 \\ 0 \\ D_z \end{bmatrix} \tag{2.4}$$

Where:

- $T = \sum_{i=1}^{8} f_i$ (Total thrust from eight (8) rotor in coaxial setup
- $f_i = b \cdot \omega_i^2$ (thrust from one rotor)

#### 2.2.2.2 Final translational Model in Inertial frame:

combining forces in the inertial frame :

$$\begin{cases} \ddot{x} &= \frac{T}{m}(c_\phi s_\theta c_\psi + s_\phi s_\psi) \\ \ddot{y} &= \frac{T}{m}(c_\phi s_\theta s_\psi - s_\phi c_\psi) \\ \ddot{z} &= \frac{T}{m}(c_\phi c_\theta) - g + dz \end{cases} \tag{2.5}$$

### 2.2.3 Step3:the rotation dynamic

#### 2.2.3.1 Newton's Second Law in Rotational Form

The time derivative of angular momentum in the inertial frame $I$ is equal to the sum of external torques:

$$[\frac{d(I\Omega)}{dt}]_I = \sum \Gamma_{\text{ext}} \tag{2.6}$$

Expressed in the body-fixed frame $B$, this becomes:

$$[\frac{d(I\Omega)}{dt}]_I = I \cdot \dot{\Omega}_B + \Omega_B \times I\Omega_B \tag{2.7}$$

Where:

- $I = diag(I_x, I_y, I_z)$ : Moment Of inertia Tensor (diagonal due to the symmetry)
- $\Omega_B = [p, q, r]^T$ : angular vilocity in the Body frame .
- $\Gamma_{ext}$ : external Torques

#### 2.2.3.2 Expanding the Equation

Assuming the symmetric inertia ,$I = diag(I_x, I_y, I_z)$, we get :

$$[\frac{Id\Omega}{dt}]_I = I \cdot \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_x \cdot p \\ I_y \cdot q \\ I_z \cdot r \end{bmatrix} \tag{2.8}$$

$$\frac{Id\Omega}{dt} = \begin{bmatrix} I_x \cdot \dot{p} \\ I_y \cdot \dot{q} \\ I_z \cdot \dot{r} \end{bmatrix} + \begin{bmatrix} (I_z - I_y) \cdot q \cdot r \\ -(I_z - I_x) \cdot p \cdot r \\ (I_y - I_x) \cdot p \cdot q \end{bmatrix} \tag{2.9}$$

#### 2.2.3.3 External Torques:

The Total torques has two parts:

1. **Control torques** $(\tau_\phi, \tau_\theta, \tau_\psi)$ : From rotor thrust difference

2. **Gyroscopic Effect**: caused by rotor spin.

$$\sum \Gamma_{ext} = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} + I_{tp} \cdot \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \cdot \omega_r \tag{2.10}$$

The equation becomes:

$$\sum \Gamma_{ext} = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} + \begin{bmatrix} I_{tp} \cdot q \cdot \omega_r \\ -I_{tp} \cdot p \cdot \omega_r \\ 0 \end{bmatrix} \tag{2.11}$$

#### 2.2.3.4 Final equations in body frame:

Combining everything , we get:

$$\begin{cases} \dot{p} = \left(\frac{I_y - I_z}{I_x}\right) q \cdot r + \frac{\tau_\phi}{I_x} + \frac{I_{tp}}{I_x}.\omega_r \cdot q, \\ \dot{q} = \left(\frac{I_z - I_x}{I_y}\right) p \cdot r + \frac{\tau_\theta}{I_y} - \frac{I_{tp}}{I_y} \cdot \omega_r \cdot p, \\ \dot{r} = \left(\frac{I_x - I_y}{I_z}\right) p \cdot q + \frac{\tau_\psi}{I_z}. \end{cases} \tag{2.12}$$

#### 2.2.3.5 Final equation in Inertial Frame:

The standard relationship between angular velocity in inertal frame and angular velocity in body frame is :

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \cdot \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.13}$$

this model can be also considered for the hover flight. Thus, $\Omega_I$ can be substituted with $\Omega_B$ in this case ($\Omega_I = \Omega_B$),[3].

$$\begin{cases} \ddot{\phi} = \left(\frac{I_y - I_z}{I_x}\right) \dot{\theta}\dot{\psi} + \frac{\tau_\phi}{I_x} + \frac{I_{tp}}{I_x}.\omega r.\dot{\theta}, \\ \ddot{\theta} = \left(\frac{I_z - I_x}{I_y}\right) \dot{\phi}\dot{\psi} + \frac{\tau_\theta}{I_y} + \frac{I_{tp}}{I_y}.\omega r.\dot{\phi}, \\ \ddot{\psi} = \left(\frac{I_x - I_y}{I_z}\right) \dot{\phi}\dot{\theta} + \frac{\tau_\psi}{I_z}. \end{cases} \tag{2.14}$$

where:
**The aerodynamic torques** The aerodynamic torques are produced by the drag and thrust forces created by the rotation of the eight (8) rotors,[4].
They are given by:

$$\begin{cases} \tau_\phi &= l \cdot b((\omega 2^2 + \omega 6^2 + \omega 3^2 + \omega 7^2) - (\omega 1^2 + \omega 5^2 + \omega 4^2 + \omega 8^2)) \\ \tau_\theta &= l \cdot b((\omega 1^2 + \omega 5^2 + \omega 2^2 + \omega 6^2) - (\omega 3^2 + \omega 7^2 + \omega 4^2 + \omega 8^2))) \\ \tau_\psi &= d \cdot (\omega 2^2 + \omega 4^2 + \omega 5^2 + \omega 7^2 - (\omega 1^2 + \omega 3^2 + \omega 6^2 + \omega 8^2)) \end{cases} \tag{2.15}$$

Where:

- $l \in \mathbb{R}^+$ is the distance between the center of gravity of the quadrotor and the axis of rotation of one of the rotors;

- $b \in \mathbb{R}^+$ is the lift coefficient;
- $d \in \mathbb{R}^+$ is the drag constant.

### 2.2.4 the state space model of the Octorotor Coaxial:

#### 2.2.4.1 State-space Equations:

The model has two parts:

1. **State Equation:** How states change over time ($\dot{X} = f(X, U)$)
2. **Output Equations:** What we measure ($Y = g(X)$)

#### 2.2.4.2 State Equation

The quadrotor robot is six degrees of freedom system defined with twelve states. The following state and control vectors are adopted:

$$X = [x, y, z, \dot{x}, \dot{y}, \dot{z}, \phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T \tag{2.16}$$

And the controls are defined as:

$$\begin{cases} U_1 & = T = b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2 + \omega_5^2 + \omega_6^2 + \omega_7^2 + \omega_8^2) \\ U_2 & = l \cdot b(\omega 2^2 + \omega 6^2 + \omega 3^2 + \omega 7^2 - (\omega 1^2 + \omega 5^2 + \omega 4^2 + \omega 8^2)) \\ U_3 & = l \cdot b(\omega 1^2 + \omega 5^2 + \omega 2^2 + \omega 6^2 - (\omega 3^2 + \omega 7^2 + \omega 4^2 + \omega 8^2)) \\ U_4 & = d \cdot (\omega 2^2 + \omega 4^2 + \omega 5^2 + \omega 7^2 - (\omega 1^2 + \omega 3^2 + \omega 6^2 + \omega 8^2)). \end{cases} \tag{2.17}$$

The state space of the drone will be :

$$\begin{cases} \dot{x}_1 & = x_4 \\ \dot{x}_2 & = x_5 \\ \dot{x}_3 & = x_6 \\ \dot{x}_4 & = \frac{U_1}{m} ux \\ \dot{x}_5 & = \frac{U_1}{m} uy \\ \dot{x}_6 & = \frac{U_1}{m}(cx_7 cx_8) - g + dz \\ \dot{x}_7 & = x_{10} \\ \dot{x}_8 & = x_{11} \\ \dot{x}_9 & = x_{12} \\ \dot{x}_{10} & = \left(\frac{I_y - I_z}{I_x}\right) x_{11} x_{12} + \frac{U2}{I_x} + I_{tp} \cdot \omega_r \cdot x_{11} \\ \dot{x}_{11} & = \left(\frac{I_z - I_x}{I_y}\right) x_{10} x_{12} + \frac{U3}{J_y} + I_{tp} \cdot \omega_r x_{10} \\ \dot{x}_{12} & = \left(\frac{I_x - I_y}{I_z}\right) x_{10} x_{11} + \frac{U4}{I_z} \end{cases} \tag{2.18}$$

with:

$$\omega_r = -\omega 1 + \omega 2 - \omega 3 + \omega 4 - \omega 5 + \omega 6 - \omega 7 + \omega 8 \tag{2.19}$$

And

$$\begin{cases} A = \frac{I_x - I_y}{I_z} \\ B = \frac{I_y - I_z}{I_x} \\ C = \frac{I_z - I_x}{I_y} \end{cases} \tag{2.20}$$

And:

$$\begin{cases} u_x = cx_7 sx_8 cx_9 + sx_7 sx_9 \\ u_y = cx_7 sx_8 sx_9 - sx_7 cx_9 \end{cases} \tag{2.21}$$

### 2.2.5 Rotor Dynamics :

The rotors are driven by DC-motors with the well known equations:

$$\begin{cases} L\frac{di}{dt} = u - Ri - K_e\omega_m \\ J\frac{d\omega_m}{dt} = \tau_m - \tau_d \end{cases} \tag{2.22}$$

As we use a small motor with a low inductance, the second order DC-motor dynamics may be approximated:

$$J\frac{d\omega_m}{dt} = -\frac{K_m^2}{R}\omega_m - \tau_d + \frac{K_m}{R}u \tag{2.23}$$

A standard DC motor is usually a 2nd order system. It is possible to model the dynamics of a DC motor system as a first order system .The transfer function of the motor dynamic is given by this 2.24,[6]:

$$G(s) = \frac{K}{\tau s + 1} \tag{2.24}$$

where $K$ and $\tau$ are the gain and the time constant of the motor, respectively.

## 2.3 System Parameters

the physical parameters used for octorotor coaxial simulation are summerized in the following table, 2.2 :

| Parameter | Symbole | Value | Unite |
|---|---|---|---|
| Octorotor coaxial masse | m | 19 | $kg$ |
| Distance between the center of a motor and the center of gravity | l | 0.4 | $m$ |
| Moment of inertia of octorotor coaxial with respect to its axis $X$ | $I_x$ | 3.04 | $kg.m^2$ |
| Moment of inertia of octorotor coaxial with respect to its axis $Y$ | $I_y$ | 3.04 | $kg.m^2$ |
| Moment of inertia of octorotor coaxial with respect to its axis $Z$ | $I_z$ | 1.52 | $kg.m^2$ |
| Moment of inertia of rotor with respect to its axis $Z$ | $I_{tp}$ | $10^{-3}$ | $kg.m^2$ |
| Lift coefficient | $b$ | $3.13 \cdot 10^{-5}$ | $N.s^2$ |
| Drag coefficient | $k$ | $7.5 \cdot 10^{-5}$ | $N.m.s^2$ |
| Gravitational constant | $g$ | 9.89 | $m \cdot s^{-2}$ |

Table 2.2: physical parameters used in the system

## 2.4 Consclusion:

The coaxial octorotor has a nonlinear and coupled dynamic model. It is affected by errors in the desired angles along the X and Y axes, which increases the complexity of the control system. As a result, the outer control loop requires a faster response time compared to the inner loop due to these dependencies. This represents a key challenge in the control design. Furthermore, managing eight motors can introduce additional errors that may negatively impact the drone's performance and stability.

# Chapter 3

# CONTROL SYNTHESIS

Drones or unmanned aerial vehicles (UAVs) require advanced control systems to ensure stability, precision, and adaptability during flight operations. These systems are essential for regulating the drone's motion, maintaining altitude, and managing orientation, all while compensating for external disturbances such as wind gusts, changes in payload, and environmental variability.
Drone control can be categorized into three main aspects: attitude control, which governs the drone's orientation (roll, pitch, yaw); position control, which manages movement in 3D space; and trajectory tracking, which ensures the drone follows a predefined path with accuracy.
To achieve effective control, a variety of techniques have been developed and implemented. Classical methods such as PID (Proportional-Integral-Derivative) control offer simplicity and real-time performance, while more advanced methods like Second Lyapunov Theory, Sliding Mode Control (SMC), and Fuzzy Logic Control (FLC) provide increased robustness and adaptability in the face of nonlinearities and uncertainties.
However, the effectiveness of these controllers heavily depends on the proper tuning of their parameters or gains. In this context, intelligent optimization algorithms such as Genetic Algorithms (GA) is widely used to automatically determine optimal gain values.

## 3.1 intelligent optimization algorithms

The optimization of control systems is crucial for ensuring the efficiency , stability, and reliability of complex systems in various fields, such as robotics, smart grids, and industrial automation. These control systems often deal with highly dynamic and non-linear processes, and achieving optimal performance requires advanced algorithms that can balance multiple objectives. Modern trends in optimization have contributed to the creation of new hybrid algorithms, which utilize various method facets to help by neglecting traditional algorithms,[7].

### 3.1.1 Genetic Algorithms (GA)

A Genetic Algorithm (GA) is an optimization method inspired by the natural process of evolution. It is used to solve complex problems by imitating how living organisms evolve and adapt over time. The genetic algorithm achieves optimization through operations such as selection, crossover, and mutation, seeking the optimal solution with the Minimize the cost function (Maximize the Fitness function) value in population The parameter tuning process is shown in Figure 3.1 , [8]:

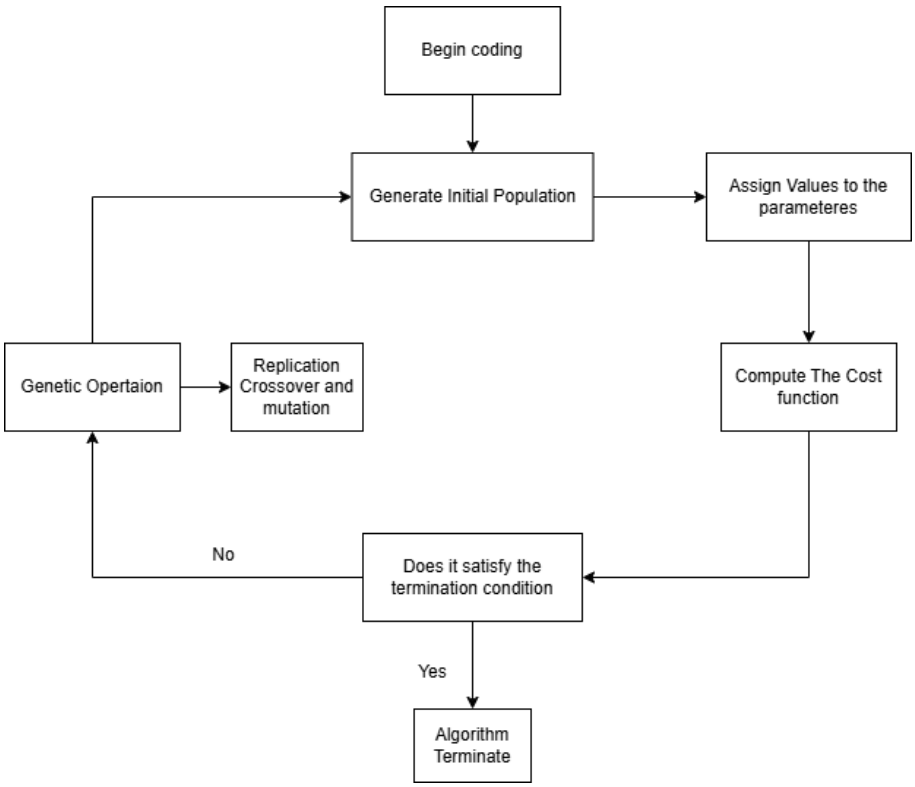



Figure 3.1: The Basic Program of Genetic Algorithm

#### 3.1.1.1 Genetic Algorithm Parameter Settings

To use a Genetic Algorithm effectively, several parameters must be carefully selected. These parameters control how the algorithm behaves and how fast it finds a good solution. In this project, the following GA settings were used,[8]:

- **Population Size:**
  This is the number of individuals (solutions) in each generation. A larger population provides more diversity but takes more time to process. In this project, the population size was set to 30, which gives a good balance between performance and speed.

- **Number of generations:**
  This defines how many times the algorithm will repeat the process of selection, crossover, and mutation. More generations give the algorithm more time to improve the solutions.

- **Selection Methode:**
  Selection is the process of choosing the best individuals from the current generation to create the next one. The Genetic Algorithm in MATLAB uses a ranking-based or stochastic universal sampling method by default, which ensures that better individuals have a higher chance of being selected.

- **Crossover:**
  Crossover mixes two parent solutions to create new offspring. This allows the algorithm to explore new combinations of parameters. MATLAB uses a default scattered crossover method, which combines elements from both parents at random points.

- **Mutation Rate:**
  Mutation introduces small random changes to the individuals. This helps the algorithm

avoid getting stuck in local minima and adds diversity to the population. In this project, the mutation rate was left at the MATLAB default value, which adjusts automatically based on the problem.

- **Cost function(Fitness function):**
The cost function measures the quality of each solution. In this project, the cost function calculates the sum of squared errors (SSE) over time, taken from the simulation of the drone system. The goal of the algorithm is to minimize this the Cost function $J$ as shown in equation 3.1 , maximize the Fitness function, which means the controller is performing well.

$$J = \int (e_x^2 + e_y^2 + e_z^2 + e_\phi^2 + e_\theta^2 + e_\psi^2) \cdot dt \tag{3.1}$$

### Remark:

We can adjust the gain of the cost function to improve the behavior, based on the characteristics of each variable.

The fitness function for this genetic algorithm optimization was set as the reciprocal of the objective function, as shown in 3.2,[8] :

$$f = \frac{1}{1+J} \tag{3.2}$$

- **Variable Bounds:**
Each parameter (for ex: $k_p$, $k_i$, $k_d$) has a minimum and maximum allowed value. These bounds prevent the algorithm from testing unrealistic or unsafe values. For example, $k_p$ values were limited between 0 and 10.

## 3.2 Adopted control strategy of Octorotor Coaxial

In the quadrotor system, it is important to note that the angles and their time derivatives are independent of the translation components. Conversely, the translations are influenced by the angles. Ideally, the system described by 2.18 can be viewed as consisting of two interconnected subsystems: angular rotations and linear translations.

$$\begin{cases} \phi_d = \arctan(u_x \cdot \sin\psi_d - u_y \cdot \cos\psi_d) \\ \theta_d = \arctan(\frac{u_x \cdot \cos\psi_d + u_y \cdot \sin\psi_d}{\cos\phi_d}) \end{cases} \tag{3.3}$$

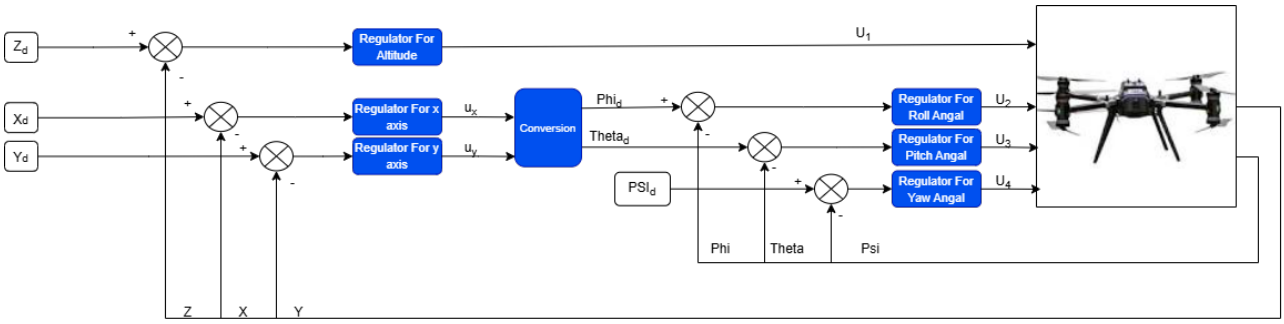



Figure 3.2: Synoptic scheme of the proposed control strategy

## 3.3 PID Controller

A **PID controller** is a widely used control system in many engineering applications, including drone flight control. The acronym **PID** stands for:

- **P – Proportional**
- **I – Integral**
- **D – Derivative**

Each component of the PID controller plays a specific role:

- **Proportional (P):** Reacts to the current error, which is the difference between the desired and actual state.
- **Integral (I):** Accounts for past errors by integrating them over time.
- **Derivative (D):** Predicts future errors by measuring how quickly the error is changing.

Together, these elements adjust the system output to minimize error and maintain stability.

### 3.3.1 Application in Drones

In drones, PID controllers are essential for ensuring **flight stability and navigation**. They are commonly used to control:

- **Altitude** – maintaining a specific height.
- **Pitch, Roll, and Yaw** – controlling the drone's orientation and balance.
- **Position Control** – maintaining or changing the drone's position accurately.

**Example:**
When a drone tilts due to wind or another disturbance, the PID controller detects the change using onboard sensors (such as gyroscopes and accelerometers) and adjusts the motor speeds to return the drone to stable flight.

### 3.3.2 Why PID is Used in Drones

- **Simple and Effective:** PID controllers are relatively easy to implement and tune.
- **Real-time Correction:** Drones require fast responses, which PID controllers provide.
- **Stability:** Helps avoid oscillations and unwanted movements during flight.
- **Adaptability:** Works well with various feedback systems and sensor inputs.

### 3.3.3 application:

In this section, we apply the PID controller to our drone system. To achieve this, we adopt a two-level control structure inspired by the same hierarchical design used in backstepping control [9]. This approach allows for modular and effective regulation of both the drone's position and orientation, 3.2.

### 3.3.4 PID Gains Tuning

Tuning the PID gains for a complex nonlinear system such as a drone is a challenging task. Due to the highly coupled dynamics, especially in the $x$ and $y$ directions, classical tuning methods are often ineffective.

For instance:

- The use of `pidtune` in MATLAB is limited because it assumes linear system models.
- Empirical or manual methods (e.g., trial and error) are time-consuming and may not yield optimal performance.
- Classical methods like the Ziegler–Nichols tuning rules often fail to provide satisfactory results in multivariable and nonlinear systems due to coupling effects.

To overcome these limitations and to efficiently search for optimal gain parameters, we used a **Genetic algorithm (GA)** algorithm.

The following table summarizes the optimized PID gains obtained for each control loop of the drone:

| **Control Loop** | **Kp** | **Ki** | **Kd** |
|---|---|---|---|
| Position X | 8.1721 | 0 | 4.89 |
| Position Y | 8.1 | 0 | 4.99 |
| Altitude (Z) | 100.8287 | 0 | 81 |
| Roll ($\phi$) | 80 | 8.4020 | 20.4165 |
| Pitch ($\theta$) | 80.0106 | 8.3765 | 20 |
| Yaw ($\psi$) | 33.1743 | 0.5 | 0.5 |

Table 3.1: Optimized PID Gains using GA

### 3.3.5 Results

The following figure present the simulation results of the coaxial octorotor. They show the drone's trajectory, orientation, and control signals under the proposed control strategy.

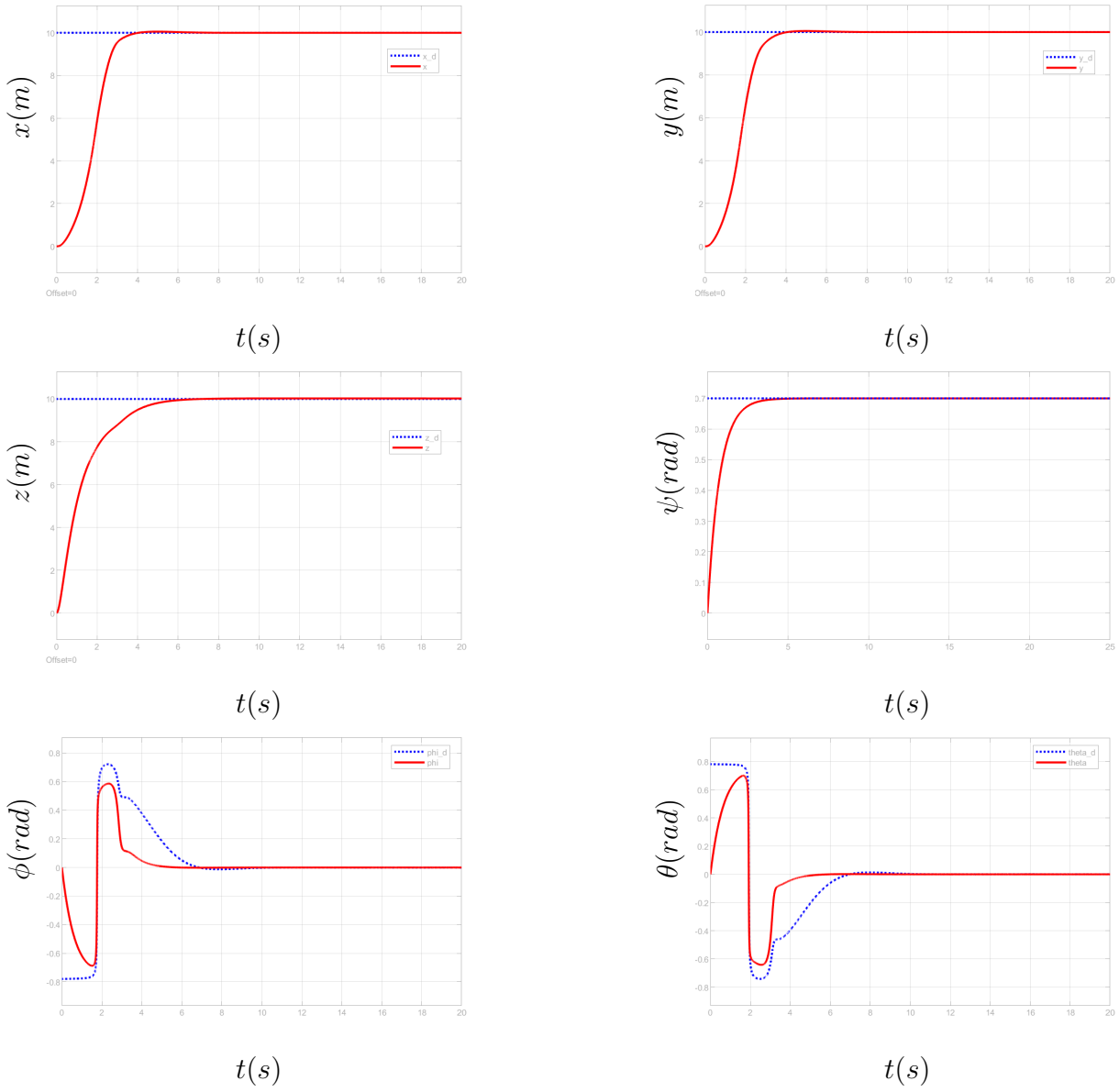



Figure 3.3: System response (PID)

**Control signals:** The following figure presents the simulation results of the control signals that produce the responses shown in Figure 3.3.

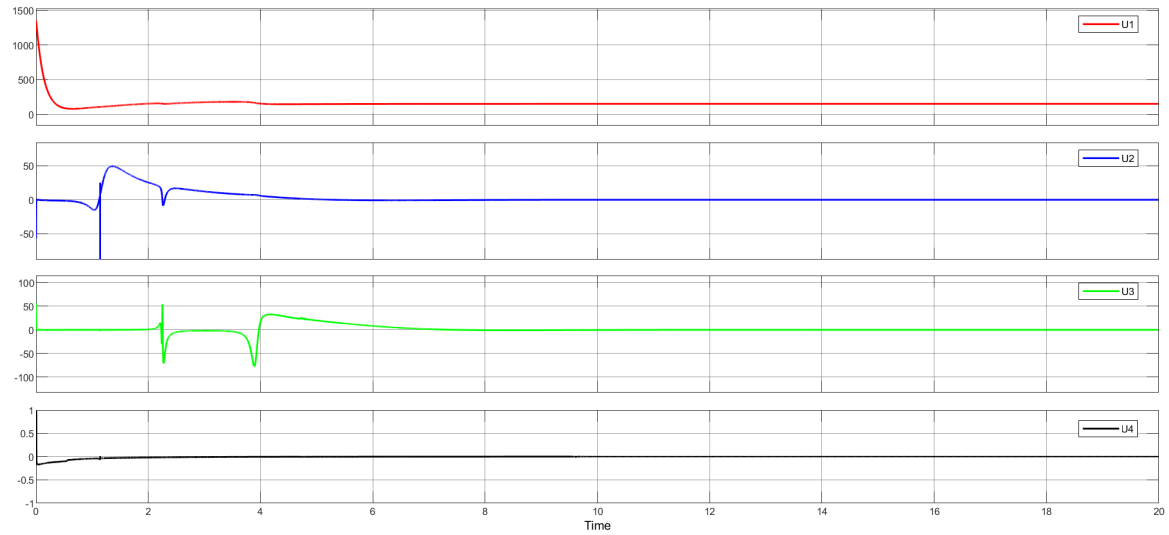



Figure 3.4: Control signals (PID)

#### 3.3.5.1 3d plot

the 3D trajectory and its desired trajectory

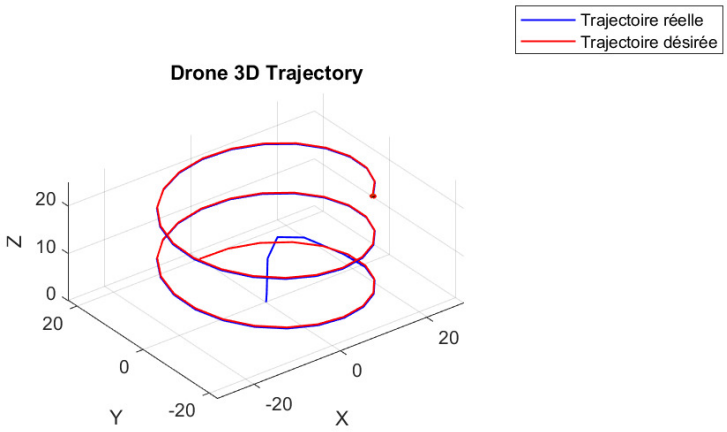



Figure 3.5: Position Evolution of the Octorotor in Space (PID)

**control signals:** The following figure presents the simulation results of the control signals that produce the responses shown in Figure 3.5.

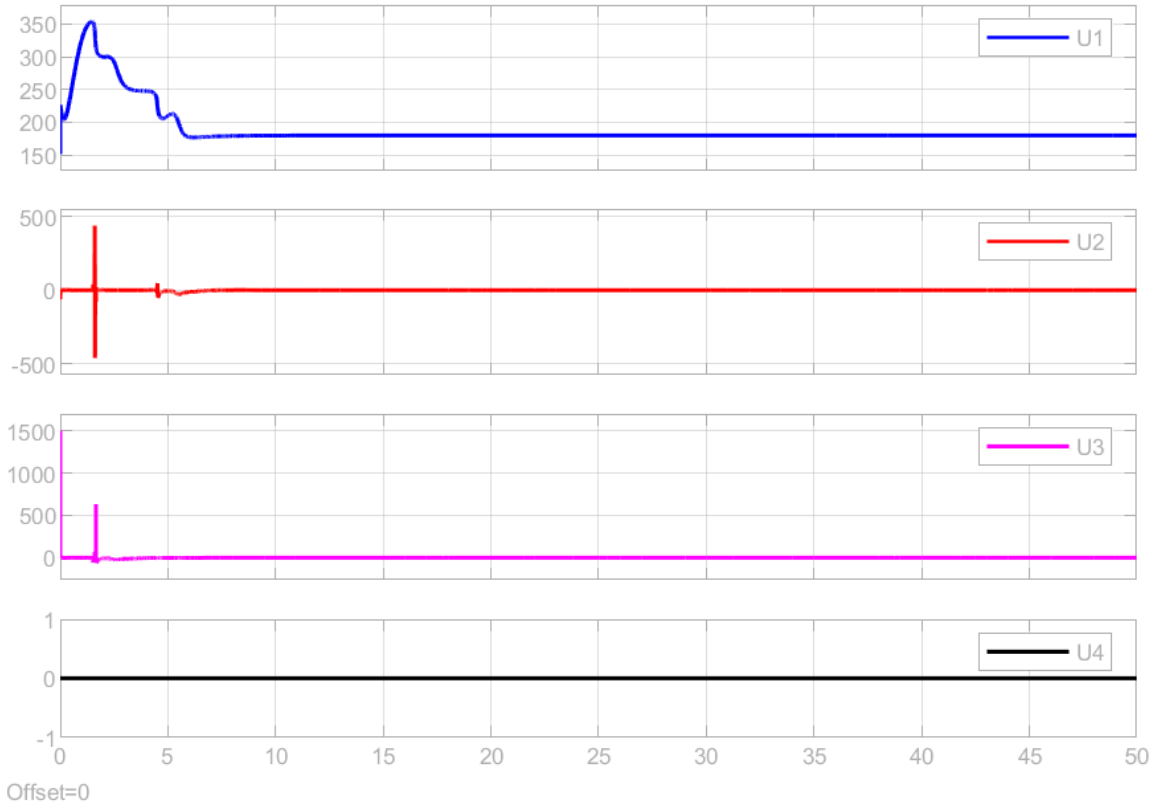



Figure 3.6: Control signals for 3d plot

### 3.3.6 Comment:

The PID controller, whose parameters were optimized using a genetic algorithm, demonstrated acceptable performance in simulation. It provided a smooth control signal and a satisfactory time response in accordance with the dynamic characteristics of the coaxial octocopter model.

However, as detailed in Chapter 4, the PID controller exhibited limited performance in terms of disturbance rejection and robustness, particularly under varying system parameters and external perturbations. These shortcomings highlight the inherent limitations of classical PID control in handling nonlinear dynamics and uncertain environments.

To enhance the system's robustness and improve its overall performance, a transition was made to LYAPONOV-based control strategies. In this context, the **backstepping control** method was selected due to its strong theoretical foundation and ability to handle nonlinear systems effectively. This method provides improved stability and resilience to disturbances, making it more suitable for the operational demands of a coaxial octocopter.

## 3.4 Second LYAPONOV (Backstepping Control)

### 3.4.1 Overview of Backstepping Control

Backsteping is a systematic control design method used for nonlinear systems . It is helpful when the system cannot be controlled using the classical linear methods due to its complexity , the main idea of Backstepping is to build the control law gradually , by dividing the system

into smaller subsystems and designing controllers for each part step-by-step,[10].
At each step , a virtual control is introduced to stable part of system , these virtual controls are then combined to design the final , real control input . this step-by-step approach ensure the ovrell stability of the system and makes it easier to manage complexe dynamices.
The methode based on **LYAPONOV Stability Theory** to prove that the system will remain stable over time . It is widely used in control engineering , including robotics , mechanical systems , aerospace applications and any system with interconnected nonlinear dynamics.

## 3.4.2 Principles Of the Method

The Methode backstepping is based on the idea of designing the control law step-by-step , starting from the part of the system that we want to control and moving backwards through the system's equations :

### 3.4.2.1 Step by step design:

- the system is divided into smaller subsystems , each with its own state variables.
- in each step , a virtual control is introduced to stabilize one subsystem .
- these virtual control act like temporary solutions until the final control is reached

### 3.4.2.2 Use LYAPONOV Theory

- At every step, a LYAPONOV function is used to check system stability.
- A control law is designed so that the derivative of the LYAPONOV function is negative, which means the system becomes more stable over time.

### 3.4.2.3 Recursive Approach

- The process continues recursively , where each control step depends on the previous one.
- At the final step , the actual control input is formed using all the previous results.

## 3.4.3 Summary of the objective:

To ensure the stability of a nonlinear system using LYAPONOV-based backstepping control, two key conditions must be satisfied:

### 3.4.3.1 LYAPONOV function $V(x)$ :

The Lyapunov function it must be Positive define function .

- $V(x) > 0$ for all $x \in \mathbb{R}^*$
- $V(0) = 0$

#### 3.4.3.2 The derivative of the LYAPONOV function

The derivative of the LYAPONOV function it must be Negative defined function .

$$\dot{V}(x) < 0$$

### 3.4.4 Application to Drone :

#### 3.4.4.1 The backstepping of the orientation subsystem

Using the backstepping approach, one can synthesize the control law forcing the system to follow the desired trajectory. Refer to 3.2 for more details. For the first step we consider the tracking-error , [6]:

$$e_{\phi 1} = x_{7d} - x_7 \tag{3.4}$$

We apply the LYAPONOV theorem by defining the LYAPONOV functione$_z$ as a positive definite function and ensuring that its time derivative is a negative definite function.

$$V_1 = \frac{1}{2} \cdot e_{\phi 1}^2 \tag{3.5}$$

The derivative of the LYAPONOV function is:

$$\dot{V_1} = (\dot{x_{7d}} - x_{10}) \cdot e_{\phi 1} \tag{3.6}$$

The stabilization of $e_\phi$ is ensured by incorporating a virtual control input $\Phi_\phi$:

$$\Phi_\phi = \dot{x_{7d}} + \lambda_\phi \cdot e_{\phi 1} \tag{3.7}$$

with $\lambda_\phi \geq 0$ then the equation becomes: :

$$\dot{V_1} = -e_{\phi 1}^2 \tag{3.8}$$

Let us perform a variable change by defining:

$$e_{\phi 2} = x_{10} - (\dot{x_{7d}} + \lambda_\phi \cdot e_{\phi 1}) \tag{3.9}$$

The augmented LLYAPONOV function:

$$V_2 = \frac{1}{2} \cdot e_{\phi 1}^2 + \frac{1}{2} \cdot e_{\phi 2}^2 \tag{3.10}$$

We introduce this substitution to eliminate the need for differentiating the desired trajectory:

$$\dot{e_{\phi 1}} = -(e_{\phi 2} + \lambda_\phi \cdot e_{\phi 1}) \tag{3.11}$$

And the time derivative is :

$$\dot{V_2} = (\dot{x_{7d}} - \Phi_\phi) \cdot e_{\phi 1} + (Bx_{11}x_{12} + \frac{I_{tp}}{I_x} x_{11}\omega_r + u_2/I_x + \lambda_\phi \cdot (e_{\phi 2} + \lambda_\phi \cdot e_{\phi 1}) - \ddot{x_{7d}} - e_{\phi 1})e_{\phi 2} \tag{3.12}$$

Since $\dot{V_2}$ is a Negative defined function($\ddot{x}_{7,8,9d} = 0$) , the control signal $u_2$ is , [6]:

$$u_2 = I_x(-Bx_{11}x_{12} - \frac{I_{tp}}{I_x} x_{11}\omega_r - \lambda_\phi \cdot (e_{\phi 2} + \lambda_\phi \cdot e_{\phi 1}) + e_{\phi 1} - \lambda_{\phi 2} e_{\phi 2}) \tag{3.13}$$

$$\dot{V}_2 = -e_1^2 - e_2^2 \tag{3.14}$$

with $\lambda_{\phi 2} \in \mathbb{R}^+$ The same procedure is applied to determine $u_3$ and $u_4$:

$$u_3 = I_y(-Cx_{10}x_{12} + \frac{I_{tp}}{I_y}x_{10}\omega_r - \lambda_\theta \cdot (e_{\theta 2} + \lambda_\theta \cdot e_{\theta 1}) + e_{\theta 1} - \lambda_{\theta 2}e_{\theta 2}) \tag{3.15}$$

$$u_4 = I_z(-Ax_{10}x_{11} - \lambda_\psi \cdot (e_{\psi 2} + \lambda_\psi \cdot e_{\psi 1}) + e_{\psi 1} - \lambda_{\psi 2}e_{\psi 2}) \tag{3.16}$$

with:

$$\begin{cases} e_\theta = x_{8d} - x_8 \\ e_{\theta 1} = x_{11} - (\dot{x}_{8d} + \lambda_\theta e_{\theta 1}) \\ e_\psi = x_{9d} - x_9 \\ e_{\psi 1} = x_{12} - (\dot{x}_{9d} + \lambda_\psi e_{\psi 1}) \end{cases} \tag{3.17}$$

#### 3.4.4.2 The backstepping of the translation Subsystem

This version clarifies that the backstepping technique is being applied as a control method to the translational part of a system

##### 3.4.4.2.1 The Altitude control:

$$e_z = x_{3d} - x_3 \tag{3.18}$$

We apply the LYAPONOV theorem by defining the LYAPONOV function $e_z$ as positive definite function and ensuring that its time derivative is negative definite function,[6].

$$V_1 = \frac{1}{2} \cdot e_z^2 \tag{3.19}$$

So the derivative of lyaponov function:

$$\dot{V_1} = (\dot{x_{3d}} - x_6) \cdot e_z \tag{3.20}$$

The stabilization of $e_z$ can be achieved by introducing a virtual control input $\Phi_z$:

$$\Phi_z = \dot{x_{3d}} + \lambda_z \cdot e_z \tag{3.21}$$

with $\lambda_z \geq 0$ then the equation will be :

$$\dot{V_1} = -e_z^2 \tag{3.22}$$

Let us perform a variable change by defining,[6]:

$$e_{z2} = x_6 - (\dot{x_{3d}} + \lambda_z \cdot e_z) \tag{3.23}$$

The augmented LYAPONOV function:

$$V_2 = \frac{1}{2} \cdot e_z^2 + \frac{1}{2} \cdot e_{z2}^2 \tag{3.24}$$

We apply this substitution to avoid the derivative of desired trajectory:

$$\dot{e_z} = -(e_{z2} + \lambda_z \cdot e_z) \tag{3.25}$$

And the time derivative is :

$$\dot{V_2} = (\dot{x_{3d}} - \Phi_z) \cdot e_z + (\frac{1}{m}(\cos x_8 \cos x_7) \cdot u_1 - g + D_z + \lambda_z \cdot (e_{z2} + \lambda_z \cdot e_z) - \ddot{x_{3d}} - e_z)e_{z2} \tag{3.26}$$

Since $\dot{V_2}$ is a Negative defined function($\ddot{x}_{1,2d} = 0$) , the control signal $u_1$ is:

$$u_1 = \frac{m}{\cos x_8 \cos x_7}(g - D_z - \lambda_z \cdot (e_{z2} + \lambda_z \cdot e_z) + e_z - \lambda_{z2} \cdot e_{z2}) \tag{3.27}$$

with $\lambda_{z2} \geq 0$

#### 3.4.4.2.2 Linear $x$ and $y$ motion control:

It can be observed that motion along the $x$ and $y$ axes depends on $u_1$ which represents the total thrust vector oriented to achieve the desired linear motion. By defining $u_x$ and $u_y$ as the component of $U_1$ responsible for motion along the $x$ and $y$ axes , respectively we can derive from equation 2.18 the necessary roll and pitch angles to compute the controls $u_x$ and $u_y$ , to ensuring that the derivative time of LYAPONOV functions are Negative defined function (NDF).

The same procedure is followed to determine $u_x$ et $u_y$ ($\lambda_x, \lambda_{x2}, \lambda_y, \lambda_{y2} \geq 0$):

$$\begin{cases} u_x = \frac{m}{u_1} \cdot (e_x - \lambda_x(e_{x2} + \lambda_x e_x) - \lambda_{x2} e_{x2}) \\ u_y = \frac{m}{u_1} \cdot (e_y - \lambda_y(e_{y2} + \lambda_y e_y) - \lambda_{y2} e_{y2}) \end{cases} \tag{3.28}$$

With :

$$\begin{cases} e_x = x_{1d} - x_1 \\ e_{x2} = x_4 - (\dot{x}_{1d} + \lambda_x e_x) \\ e_y = x_{2d} - x_2 \\ e_{y2} = x_5 - (\dot{x}_{2d} + \lambda_y e_y) \end{cases} \tag{3.29}$$

### 3.4.4.3 Optimized Gains Using Genetic Algorithm for Backstepping Control

In this Method , a Genetic Algorithm (GA) was applied to optimize the control gains used in a Backstepping controller for a Octorotor Coaxial drone,3.1.1. The Backstepping method was chosen for its ability to handle nonlinear systems and ensure system stability through recursive LYAPONOV-based design.

For each controlled variable($x$,$y$,$z$,$\phi$,$\theta$,$\psi$) , two tuning parameters (gains) were optimized $\lambda_1$ and $\lambda_2$.

The Genetic Algorithm was configured with a population size of 30 and a maximum of 50 generations. Upper and lower bounds were set for each parameter (gain) to restrict the search space. The cost function minimized by the algorithm was the total squared tracking error, 3.1.Calculated over the simulation time as follows:

$$J = \int_0^T (e_x^2(t) + e_y^2(t) + e_z^2(t) + e_\phi^2(t) + e_\theta^2(t) + e_\psi^2(t)) \cdot dt \tag{3.30}$$

After optimization, the following gains were obtained , as shown in 3.2 :

| Variable | $\lambda_1$ | $\lambda_2$ |
|---|---|---|
| $x$ | 1 | 10 |
| $y$ | 1 | 10 |
| $z$ | 0.1 | 5 |
| $\phi$ | 1 | 10 |
| $\theta$ | 1 | 10 |
| $\psi$ | 0 | 10 |

Table 3.2: Gains of Backstepping control

### 3.4.4.4 Results

The following figure present the simulation results of the coaxial octorotor. They show the drone's trajectory, orientation, and control signals under the proposed control strategy.

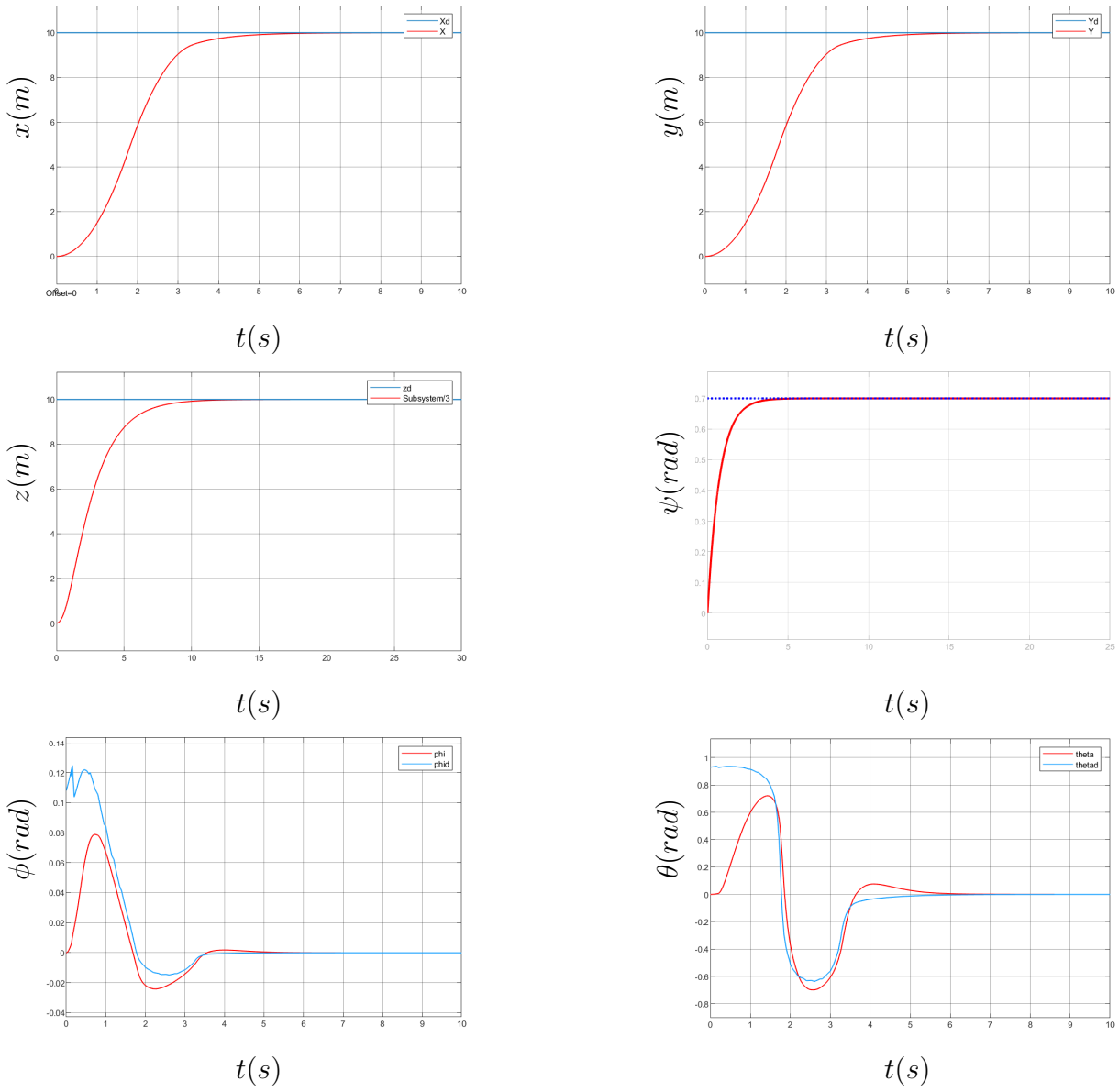



Figure 3.7: System response (Backstepping)

#### 3.4.4.4.1 Control Signals:

**Control signals:** The following figure presents the simulation results of the control signals that produce the responses shown in Figure 3.7.

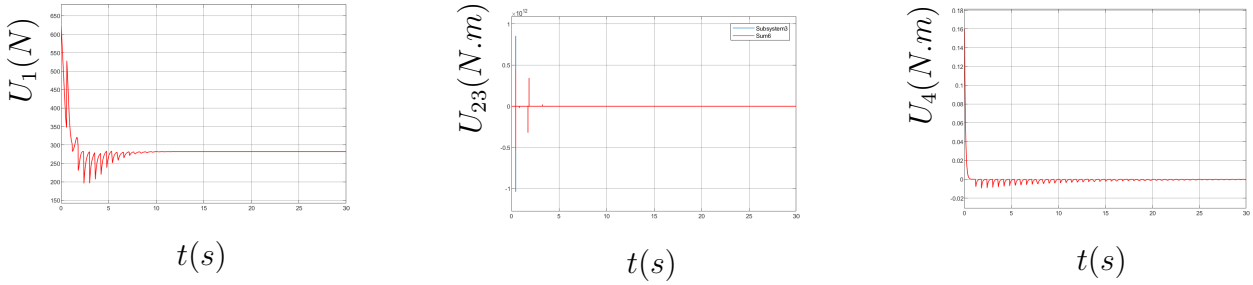



Figure 3.8: Control signals (Backstepping)

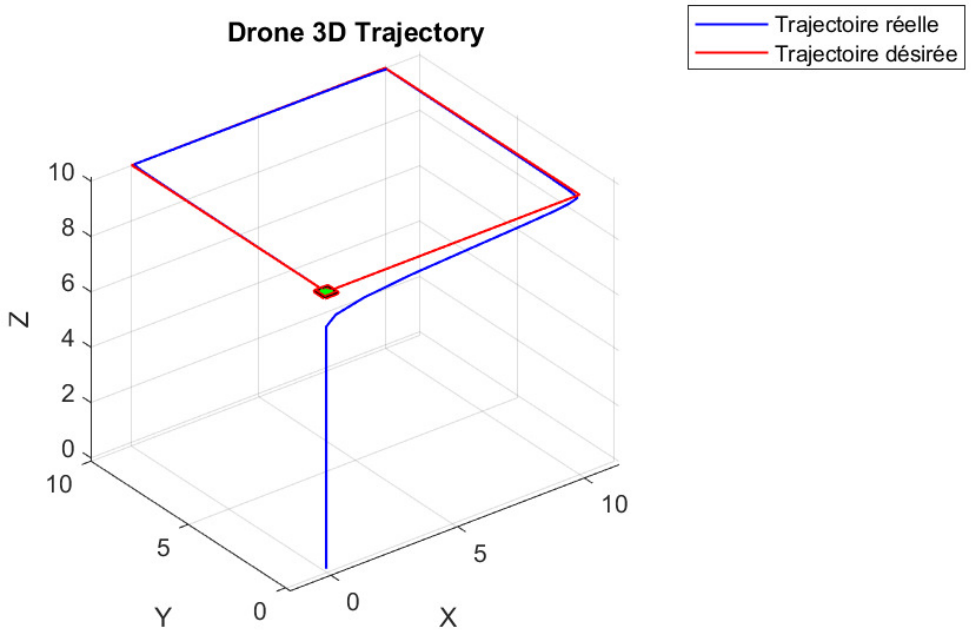



Figure 3.9: Position Evolution of the Octorotor in Space (Backstepping)

## Comment:

The backstepping controller demonstrated superior performance compared to the PID controller, particularly in terms of tracking accuracy and response time. However, during robustness testing especially under varying payload and structural mass conditions as detailed in Chapter 4,its performance declined. Additionally, the control signals generated by the backstepping method were observed to be unbounded in certain scenarios, which could lead to actuator saturation and instability in practical implementations.

Due to these limitations, particularly the lack of robustness against parameter variations and the issue of unbounded control effort, it became necessary to explore alternative control strategies. As a result, the focus shifted to **Sliding Mode Control (SMC)**, a method well-known for its robustness and its ability to maintain stability and performance in the presence of significant uncertainties and external disturbances.

## 3.5 Sliding Mode Control (SMC)

### 3.5.1 Road Map for Sliding Mode Control Design

In control systems, SMC is a nonlinear control method that alters the dynamics of a nonlinear system by application of a discontinuous control signal that forces the system to slide along surface with desired motions in the system state space. such a system can switch from one continuous structure to another based on the current position in the state space,[11].

#### 3.5.1.1 Definition of Sliding mode

The main idea of sliding mode control consists of enforcing the trajectories of closed-loop system to belong to a prescribed surface (sliding surface). The surface has to be selected in advance in order to fulfill the required performance of the closed-loop system. Next, the control law has to be designed to enforce the sliding mode.
To introduce a definition of sliding mode for a finite-dimensional model, we try to find a minimal set of properties characterizing it . The finite-time convergence to a sliding set is a specific feature of the finite-dimensional sliding mode systems.The discontinuity of the sliding mode control provides some robustness properties for the closed-loop system such as invariance with respect to the matched perturbations ,[11].

#### 3.5.1.2 Definition of the Sliding Manifold

The sliding surface is typically defined as a scalar function of the system state:

$$s(x) = 0 \tag{3.31}$$

This surface is designed so that when the system evolves on it, the dynamics of the system behave in a desired and stable way. The form of $s(x)$ depends on the control objective,[11] .

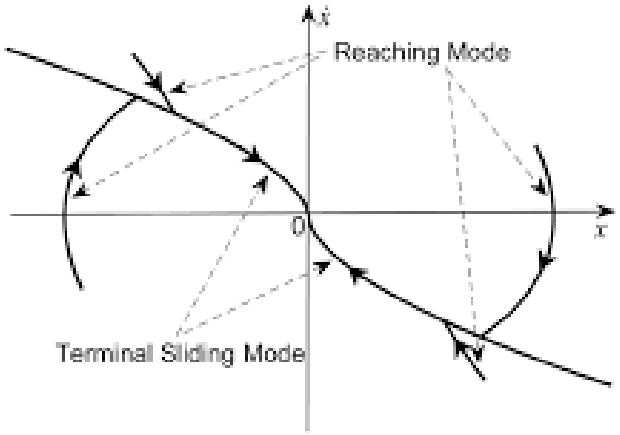



Figure 3.10: Sliding surface

#### 3.5.1.3 Preconditions for Sliding Mode Control

Before the system can enter sliding mode, it must first reach the sliding surface, which is defined by $s(x) = 0$ . To make sure the system moves toward this surface, we must satisfy an important condition,[11]:

$$\dot{s} \cdot s < 0 \tag{3.32}$$

This mean :

- If the state is above the surface ($s > 0$), then ($\dot{s} < 0$
- If the state is below the surface ($s < 0$), then ($\dot{s} > 0$

This guarantees that the trajectory will converge to $s(x) = 0$ in finite time.
This condition means that the direction of movement (represented by $\dot{s}$) must always push the system closer to the surface. One common way to make this happen is by using a special rule for the control called the reaching law:

$$\dot{s} = -k \cdot sign(s), \text{ with } k > 0 \tag{3.33}$$

where:

$$sign(s) = \begin{cases} +1, & s > 0 \\ 0, & s = 0 \\ -1, & s < 0 \end{cases} \tag{3.34}$$

#### 3.5.1.4 Control Action Before Sliding

During the reaching phase, the control law must ensure that this condition is met. To do that, a discontinuous control component is added, often involving a switching function like $sign(s)$ . A general control law can be written as:

$$u = u_{eq} - k \cdot sign(s) \tag{3.35}$$

where:

- $u_{eq}$ : The equivalent control, which maintains sliding motion once on the surface.

### 3.5.2 Application to Drone

By implementing Sliding Mode Control (SMC), the control law is formulated to guarantee that the system tracks the desired trajectory,[6].

#### 3.5.2.1 The Sliding Mode Control of the orientation subsystem

The first step is similar to the backstepping approach, except that $S_\phi$ is used in place of $e_{\phi\phi}$ 3.4:

$$s_\phi = x_{10} - (\dot{x}_{7d} + \lambda_\phi e_{\phi 1}) \tag{3.36}$$

The selected control signal for the attractive surface is the time derivative of $s_\phi$ ensuring that $s\dot{s} < 0$ .

$$\begin{cases} \dot{s}_\phi = -k_{1\phi} sign(s_\phi) - k_{2\phi} s_\phi \\ \dot{s}_\phi = B x_{11} x_{12} + \frac{I_{tp}}{I_x} x_{11} \omega_r + \frac{u_2}{I_x} + \lambda_\phi (s_\phi + \lambda_\phi e_\phi) \end{cases} \tag{3.37}$$

Thus, the control signal $u_2$ is

$$u_2 = -B x_{11} x_{12} - \frac{I_{tp}}{I_x} x_{11} \omega_r - \lambda_\phi e_\phi - k_{1\phi} sign(s_\phi) - k_{2\phi} s_\phi \tag{3.38}$$

The same steps are followed to derive the remaining control signals.

$$\begin{cases} u_3 = -C x_{10} x_{12} - \frac{I_{tp}}{I_y} x_{10} \omega_r - \lambda_\theta e_\theta - k_{1\theta} sign(s_\theta) - k_{2\theta} s_\theta \\ u_4 = -A x_{11} x_{10} - \lambda_\psi e_\psi - k_{1\psi} sign(s_\psi) - k_{2\psi} s_\psi \end{cases} \tag{3.39}$$

where $k_{1\phi,\theta,\psi} \in \mathbb{R}^+$ and $k_{2,\phi,\theta,\psi} \in \mathbb{R}^+$
with :

$$\begin{cases} s_\theta = x_{11} - (\dot{x}_{8d} + \lambda_\theta e_{\theta 1}) \\ s_\psi = x_{12} - (\dot{x}_{9d} + \lambda_\psi e_{\psi 1}) \end{cases} \tag{3.40}$$

#### 3.5.2.2 The Sliding Mode Control of the Translation subsystem

##### 3.5.2.2.1 The Altitude control

The first step is similar to the backstepping approach, except that $s_z$ is used instead of $e_{zz}$ as done in the orientation subsystem control.

$$s_z = x_6 - (\dot{x}_{3d} + \lambda_z e_{z1}) \tag{3.41}$$

Thus, we follow the same steps as in the orientation subsystem to determine the control signal.

$$\begin{cases} \dot{s}_z = -k_{1z} sign(s_z) - k_{2z} s_z \\ \dot{s}_z = \frac{1}{m} (\cos x_8 \cos x_7) \cdot u_1 - g + D_z + \lambda_z \cdot e_z \end{cases} \tag{3.42}$$

The control signal ($u_1$) is:

$$u_1 = \frac{m}{\cos x_8 \cos x_7} (g - D_z - \lambda_z e_e - k_{1z} sign(s_z) - k_{2z} s_z) \tag{3.43}$$

##### 3.5.2.2.2 Linear $x$ and $y$ motion control

The same explanation applies for deriving $u_x$ and $u_y$ as in the backstepping approach.

$$\begin{cases} u_x = \frac{m}{u_1} (-\lambda_x e_x - k_{1x} sign(s_x) - k_{2x} s_x) \\ u_y = \frac{m}{u_1} (-\lambda_y e_y - k_{1y} sign(s_y) - k_{2y} s_y) \end{cases} \tag{3.44}$$

with:

$$\begin{cases} s_x = x_4 - (\dot{x}_{1d} + \lambda_x e_{x1}) \\ s_y = x_5 - (\dot{y}_{2d} + \lambda_y e_{y1}) \end{cases} \tag{3.45}$$

#### 3.5.2.3 Optimized Gains Using Genetic Algorithm for Sliding Mode Control

In this project, a Genetic Algorithm (GA) ,3.1.1 was used to optimize the control gains of a Sliding Mode Controller (SMC) for a Octorotor coaxial drone. The controller was designed to track a desired trajectory in all six degrees of freedom: $x$ , $y$ , $z$ , $\phi$ , $\theta$ , $\psi$. To ensure robustness and stability .
Each controlled variable was associated with three tuning parameters (gains) :$\lambda$ , $k_1$ , $k_2$ .
The Genetic Algorithm was configured with a population size 30 , a maximum of 50 generation , and upper and lower bound of each parameter (gain) .
The cost function used by the Genetic Algorithm was the total tracking error computed as,3.1 :

$$J = \int_0^T (e_x^2(t) + e_y^2(t) + e_z^2(t) + e_\phi^2(t) + e_\theta^2(t) + e_\psi^2(t)) \cdot dt \tag{3.46}$$

after running The optimization process (GA) , the following control gains were obtined, as shown in 3.3 :

| Variable | $\lambda$ | $k_1$ | $k_2$ |
|---|---|---|---|
| $x$ | 1 | 0.12 | 0 |
| $y$ | 0.01 | 0.902 | 0 |
| $z$ | 1.009 | 0.763 | 0 |
| $\phi$ | 0.201 | 1.248 | 0 |
| $\theta$ | 1.193 | 0.709 | 0 |
| $\psi$ | 1.297 | 0.469 | 0.653 |

Table 3.3: Gains of Sliding mode control

### 3.5.3 Results

The following figures present the simulation results of the coaxial octorotor. They show the drone's trajectory, orientation, and control signals under the proposed control strategy.

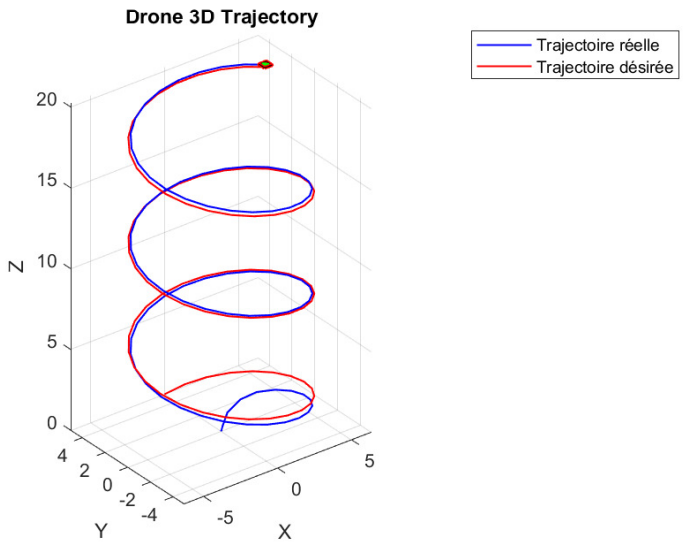



Figure 3.11: Position Evolution of the Octorotor Coaxial in Space(SMC)

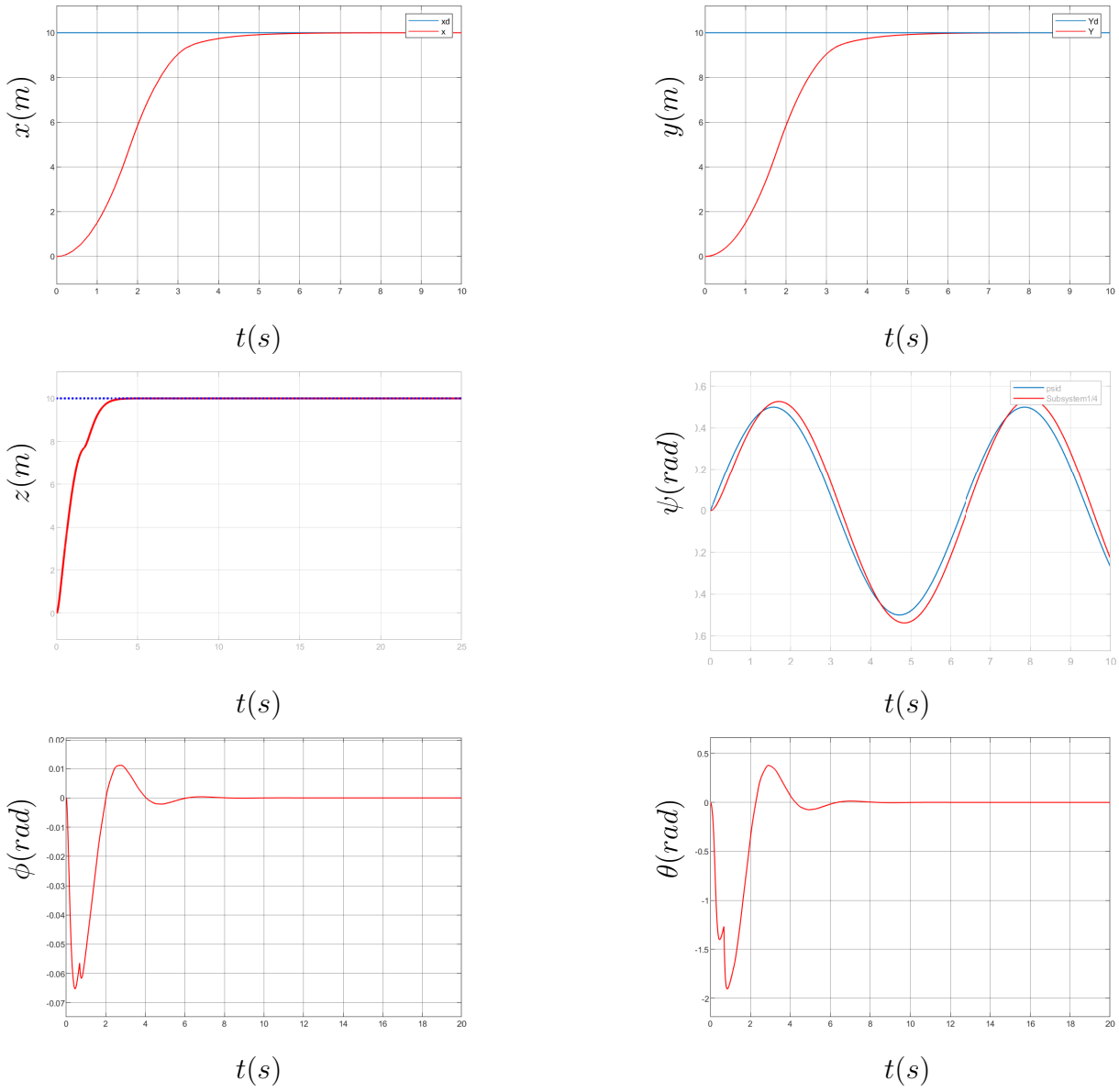



Figure 3.12: System response (Sliding Mode Control (SMC))

**Control signals:** The following figure presents the simulation results of the control signals that produce the responses shown in Figure 3.12.

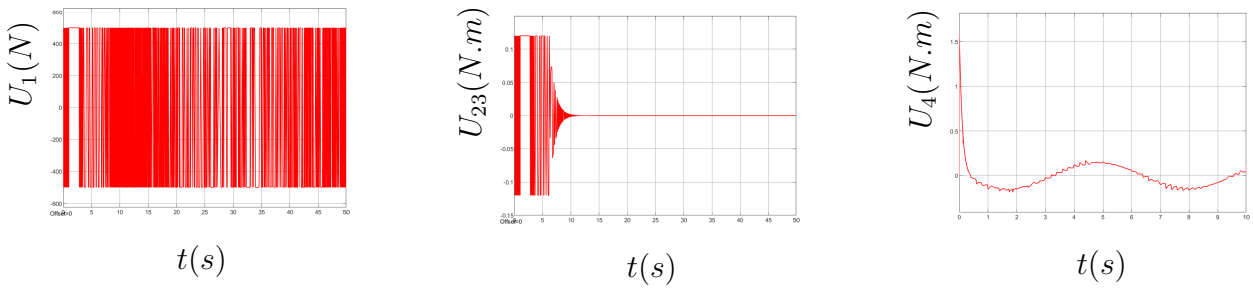



Figure 3.13: Control signals (Sliding Mode Control)

**Comment :**

Revised Paragraph: Sliding Mode Control (SMC) demonstrates high performance, acceptable time response, and strong robustness, as shown in Chapter 4. However, its control signals are not well-suited for practical applications such as DC motors and drones, as they suffer from the chattering phenomenon, which results in high energy consumption and possible hardware degradation. To address this issue, several techniques have been proposed to reduce or eliminate chattering. In this work, we propose the integration of a Fuzzy Logic Controller (FLC) with SMC to generate smoother control signals, thereby improving energy efficiency and ensuring better compatibility with sensitive actuators. We can use high-order sliding mode control to improve the control signals and minimize energy consumption.

## 3.6 Fuzzy Logic Control

### 3.6.1 Introduction to Fuzzy Logic:

Fuzzy logic is a type of logic that deals with partial truth. Unlike traditional logic, where something is either true or false, fuzzy logic allows values between 0 and 1. For example, instead of saying water is just "hot" or "cold," fuzzy logic can say it is 0.6 hot and 0.4 warm.

It was created by Lotfi Zadeh in 1965 to handle situations with uncertain or vague information. Fuzzy logic is used in systems like smart devices, AI, and control systems, where exact data isn't always available.

In short, fuzzy logic helps machines make better decisions in real-world situations that are not always black and white.

### 3.6.2 Design of the Fuzzy Controller

The fuzzy controller is designed to stabilize and control the position, altitude, and attitude of a quadrotor drone using a set of parallel fuzzy logic controllers (FLCs). This approach allows the drone to track a desired trajectory smoothly and effectively, even in the presence of uncertainties and disturbances,[12].

#### 3.6.2.1 Structure

The control system is divided into **two main subsystems** 3.2:

1. **Attitude control:** manages roll $\phi$ , pitch $\theta$ and pitch $\psi$ angles.
2. **Position and Altitude Control:** controls $x, y$ (horizontal movement) , and z (vertical altitude)

Each axis is controlled by an individual fuzzy controller, resulting in six FLCs:

- FLC$\phi$ (Roll) , FLC$\theta$ (Pitch), FLC $\psi$ (Yaw).

- FLC$x$ , FLC$y$ , FLC$z$ (Position and altitude)

These controllers are implemented using a mamdani fuzzy inference system, where the output of each rule is a fuzzy set, [12].

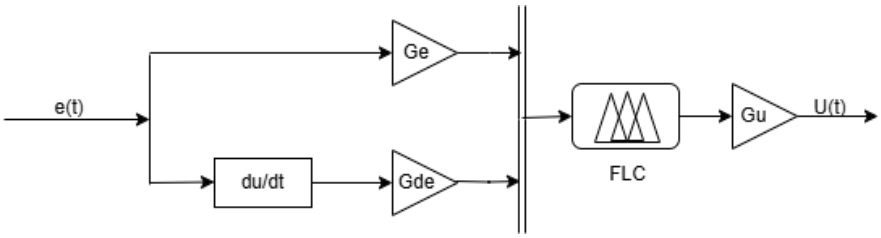



Figure 3.14: Structure of individual fuzzy controller

### 3.6.2.2 Global fuzzy logic control architecture

Each fuzzy logic controller (FLC1 to FLC6) follows the structure shown in Figure 3.14 . These controllers are integrated into a hierarchical architecture (figure 3.15), [13]

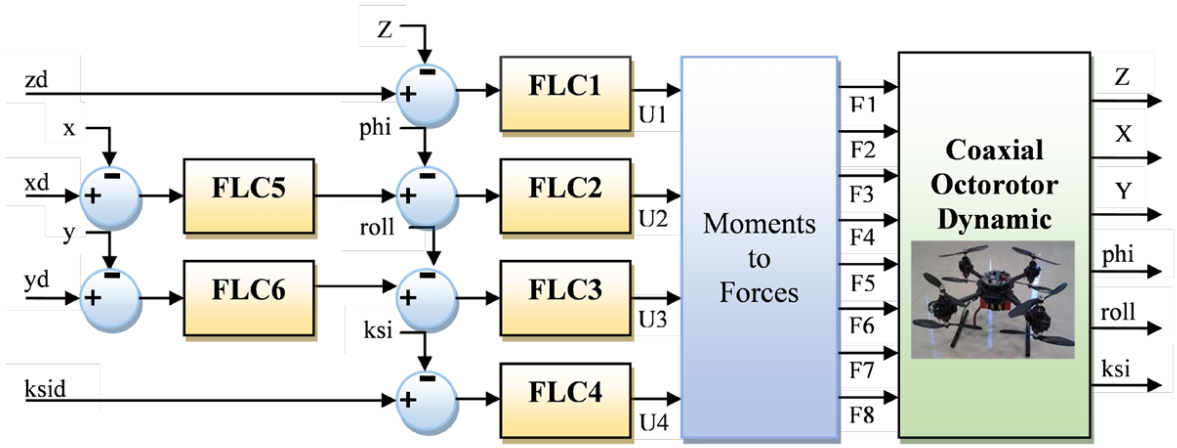



Figure 3.15: Global fuzzy logic controller

- **Outer Loop:** Controls the position (X, Y, Z) using FLC5 and FLC6 to generate orientation references ($\phi$ and $\theta$) , while FLC1 directly regulates altitude.
- **Inner Loop:** Stabilizes the drone's attitude using FLC2, FLC3, and FLC4 to control roll ($\phi$) , pitch ($\theta$) and yaw ($\psi$) , respectively.
- The outputs of the FLCs (U1 to U4) are converted into motor thrusts ($\omega_1$ to $\omega_8$) through the "Moments to Forces" block, adapted to the dynamics of the coaxial octorotor.

### 3.6.2.3 Define Input Variables

- **Inputs for each FLC:** [12] and [14]
  - **Error (e):** the difference between the desired and current value
  - **Derivative of error ($\dot{e}$):** the rate of change of the error :

* FLC$\phi$ :$e_\phi = \phi_d - \phi$ , $\dot{e_\phi} = \frac{de_\phi}{dt}$
* FLC$\theta$ :$e_\theta = \theta_d - \theta$ , $\dot{e_\theta} = \frac{de_\theta}{dt}$
* FLC$\psi$ :$e_\psi = \psi_d - \psi$ , $\dot{e_\psi} = \frac{de_\psi}{dt}$
* FLC$x$ :$e_x = x_d - x$ , $\dot{e_x} = \frac{de_x}{dt}$
* FLC$y$ :$e_y = y_d - y$ , $\dot{e_y} = \frac{de_y}{dt}$
* FLC$z$ :$e_z = z_d - z$ , $\dot{e_z} = \frac{de_z}{dt}$

- **Output:**
  The fuzzy controller generates several outputs to control the drone's motion. Each output corresponds to a specific part of the drone's behavior,[3]:

  - $U_1$ **Altitude control** $z$**:** Controls the vertical movement along the z-axis
  - $U_2$ **Roll control** $\phi$**:** Adjusts the left/right tilt to help control lateral movement.
  - $U_3$ **Pitch control** $\theta$ **:** Adjusts the forward/backward tilt to help control longitudinal movement.
  - $U_4$ **Yaw control** $\psi$**:** Controls rotation around the vertical axis (turning left/right).
  - $U_x$ **Virtual Control for x-axis Position:** Helps move the drone forward or backward by influencing pitch ($\theta$).
  - $U_y$ **Virtual control for y-axis Position:** Helps move the drone left or right by influencing roll ($\phi$).

### 3.6.3 Definition of Membership Functions (fuzzification):

#### 3.6.3.1 Input Membership Functions

In the fuzzy controller , both error ($e$) and the derivative of error ($\dot{e}$) are used as inputs , but each has its own membership function definition :

For **error (e) , [14] :**

- Negative : $\mu_N$ = trapezoidal{-5 -5 -2.5 0} .
- zero : $\mu_Z$ = triangle{-3 0 3}.
- Positive : $\mu_P$ = trapezoidal {0 2.5 5 5}

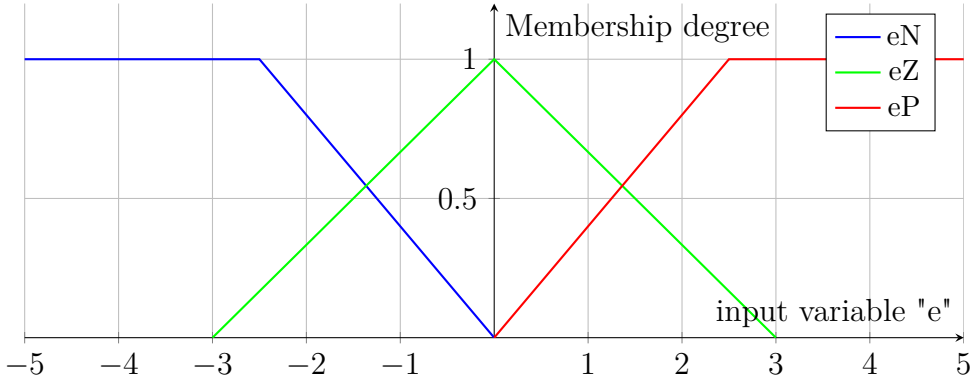



Figure 3.16: Membership function of $e$

For **derivative of error** ($\dot{e}$), **[14] :**

- Big Negative : $\mu_{BN}$ = gaussian{-6 -4 -4 -2 }
- Negative : $\mu_N$ = gaussian{-4 -2 -2 0}
- Zero : $\mu_Z$ = gaussian{-1 0 0 1}
- Positive : $\mu_P$ = gaussian{0 2 2 4}
- Big Positive : $\mu_{BP}$ = gaussian{2 4 4 6}

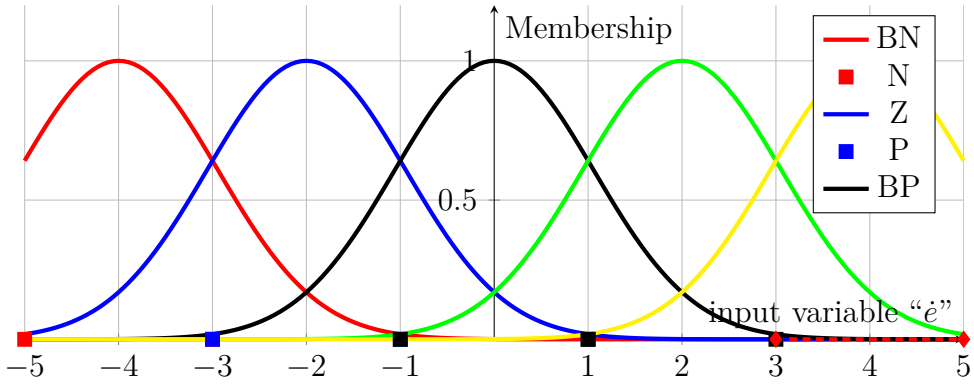



Figure 3.17: Membership function of $\dot{e}$

### 3.6.3.2 Output Membership Function :

The output membership functions represent the control actions chosen by the fuzzy rules, [14].

- Down Much : $\mu_{DM}$ = trapezoidal{-12.5 -12.5 -6 -3}
- Down : $\mu_D$ = triangle{-6 -3 0}
- Zero : $\mu_Z$ = triangle{-2 0 2}
- Up : $\mu_U$ = triangle{0 3 6}
- Up Much : $\mu_{UM}$ = trapezoidal{3 6 12.5 12.5}

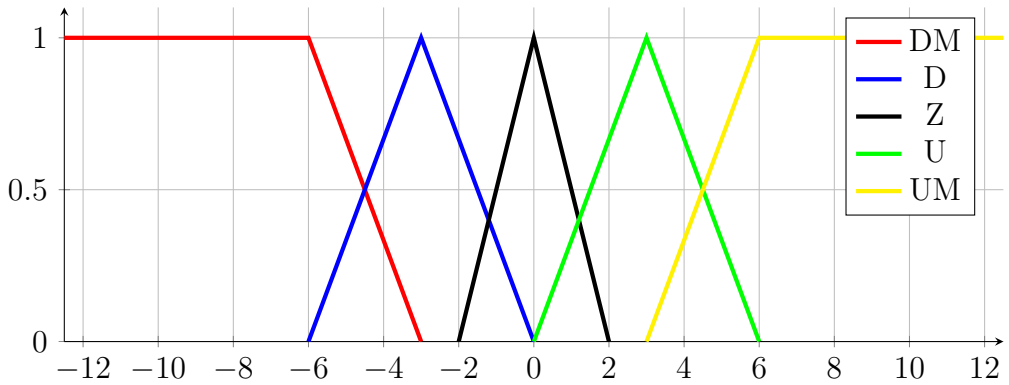



Figure 3.18: Membership function of input $U$

### 3.6.4 Rule Base Development:

The fuzzy rule base contains twelve rules that define how the controller reacts to different combinations of error and derivative of error, allowing the system to choose the correct control action based on both the current state and how it is changing,[14].

| **E / dE** | **BN** | **N** | **Z** | **P** | **BP** |
|---|---|---|---|---|---|
| **N** | DM | DM | D | Z | U |
| **Z** | DM | D | Z | U | UM |
| **P** | D | Z | U | UM | UM |

Table 3.4: Rule Base Development

### 3.6.5 Defuzzification

The defuzzification is performed using the weighted average method, producing a single control value based on the activated rules. This method ensures a fast and smooth output suitable for real-time control.

$$U = \frac{\int u_c(x)xdx}{\int u_c(x)dx} \tag{3.47}$$

### 3.6.6 Optimization Gains using Genetic Algorithm of Fuzzy Logic Control (FLC)

In this study, a Genetic Algorithm (GA) was used to optimize the scaling factors of a Fuzzy Logic Controller (FLC) for a quadrotor drone. The FLC was designed to control the drone's motion in all six degrees of freedom: linear positions ($x$,$y$,$z$) and angular oriantations ($\phi$,$\theta$,$\psi$). Fuzzy logic is especially useful in systems that are difficult to model precisely, as it allows control decisions based on linguistic rules and approximate reasoning.
Each fuzzy controller was adjusted using three gains:$K_e$,$K_{de}$ and $K_u$ . The gain $K_e$ scales the error input , $K_{de}$ scales the derivative of error input and $K_u$ scales the output of controllers .
To find the optimal values for these gains, a Genetic Algorithm was applied. The GA was configured with a population size of 30 and ran over 40 generations. Each individual in the population represented a complete set of 18 gains—three for each of the six control axes. The performance of each solution was evaluated using a cost function, which measured the total tracking error over time. The cost function was defined as,3.1:

$$J = \int_0^T (e_x^2(t) + e_y^2(t) + e_z^2(t) + e_\phi^2(t) + e_\theta^2(t) + e_\psi^2(t)) \cdot dt \tag{3.48}$$

After the optimization process, the best set of fuzzy control gains was selected , as shown in 3.5 :

| Variable | $k_e$ | $k_{de}$ | $k_u$ |
|---|---|---|---|
| $x$ | 1 | 1 | 50 |
| $y$ | 1 | 1 | 50 |
| $z$ | 50 | 60 | 90 |
| $\phi$ | 1 | 1 | 2 |
| $\theta$ | 1 | 1 | 1 |
| $\psi$ | 1 | 1 | 1 |

Table 3.5: Gains of Fuzzy Logic control

### 3.6.7 Results

The following figures present the simulation results of the coaxial octorotor. They show the drone's trajectory, orientation, and control signals under the proposed control strategy.

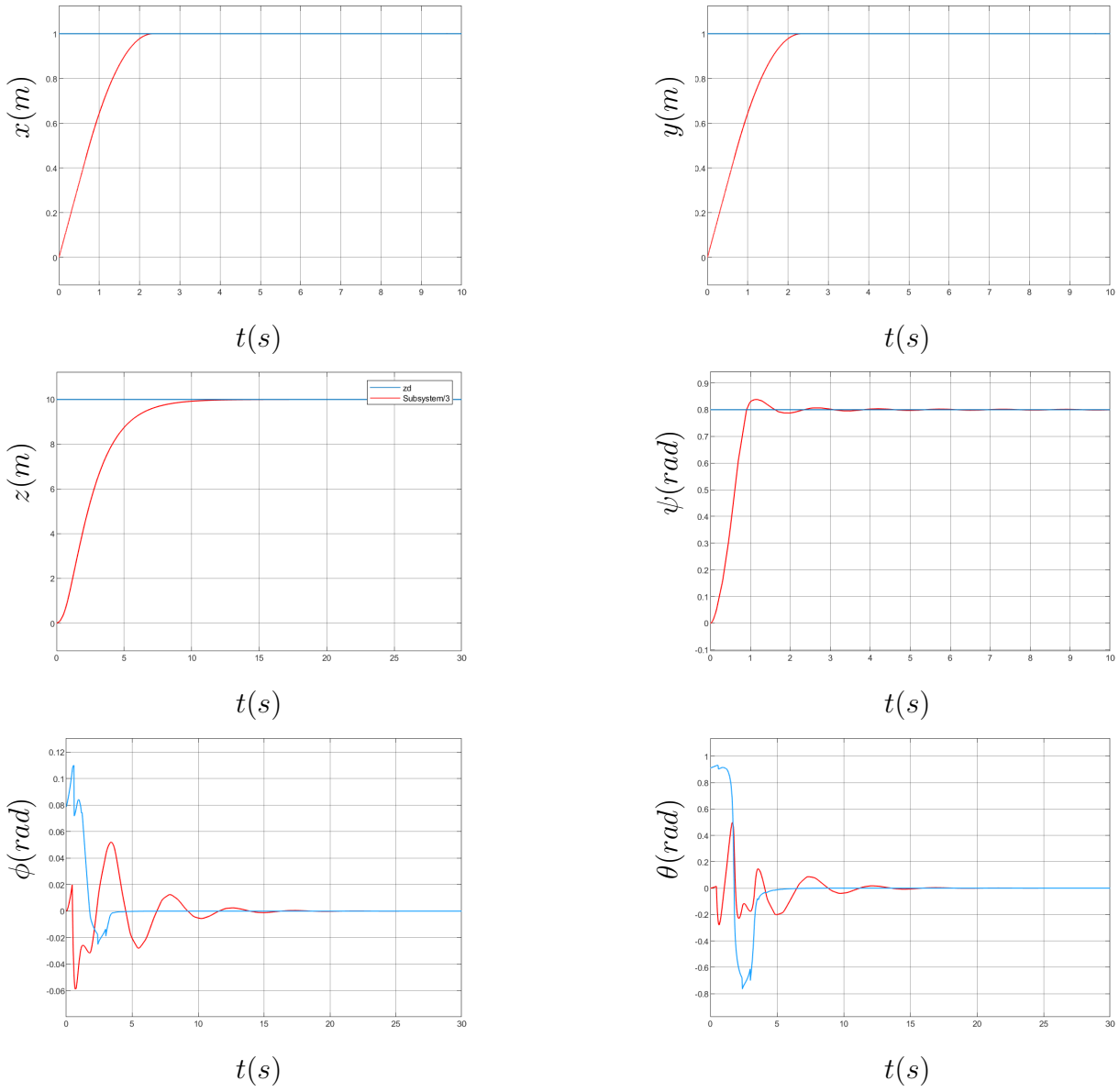



Figure 3.19: System response (Fuzzy Logic Control(FLC))

**Control signals:** The following figure presents the simulation results of the control signals that produce the responses shown in Figure 3.19.

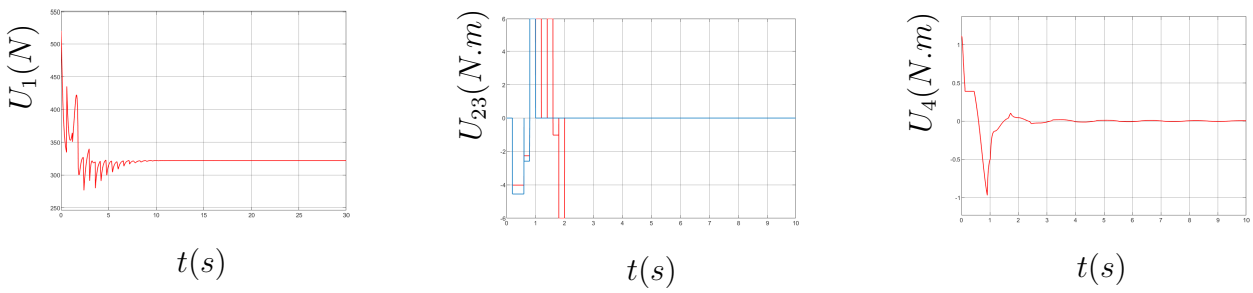



Figure 3.20: Control signals (FLC)

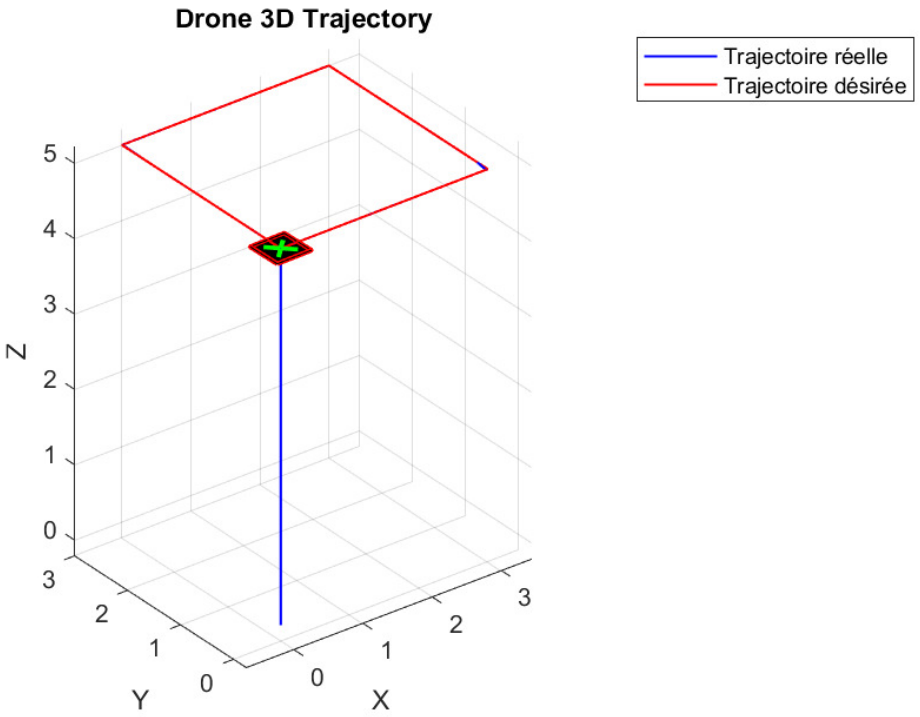



Figure 3.21: Position Evolution of the Octorotor in Space (FLC)

### Comment:

The integration of a Fuzzy Logic Controller (FLC) into the control system of a drone offers significant advantages, particularly in handling nonlinear dynamics and uncertainties commonly encountered during flight. Drones require smooth and precise control signals to maintain stability and ensure energy-efficient operation. Traditional Sliding Mode Control, while robust, can generate high-frequency oscillations (chattering) that negatively affect actuators and reduce overall system efficiency. By incorporating FLC, the control strategy becomes more adaptive and capable of generating smoother control actions. This hybrid approach enhances flight stability, reduces energy consumption, and increases the lifespan of the motors making it highly suitable for real-time drone applications.

## 3.7 Conclusion

In this chapter, we have presented a theoretical study of four control strategies applied to the coaxial octorotor drone: PID control, Backstepping, Sliding Mode Control (SMC), and Fuzzy Logic Control (FLC). Each technique was analyzed in terms of design principles, expected behavior, and suitability for the nonlinear and multivariable nature of the system.
The PID controller offers a simple and well-established solution, yet it may struggle with complex nonlinearities and external disturbances. The Backstepping method, based on a systematic nonlinear control approach, allows for improved stability and tracking performance. Sliding Mode Control is known for its robustness and effectiveness under uncertainties, though it may lead to chattering effects. Fuzzy Logic Control provides a model-free, flexible solution, capable of handling imprecision and adapting to variable conditions.
Although this work was conducted as part of a project and not validated through simulations or

experimental tests, the theoretical development offers a strong foundation for future implementation. Upcoming work will aim to simulate and compare these control strategies in practical scenarios to evaluate their performance under real-world constraints.

# Chapter 4

# Comparative Analysis of Control Strategies for Octorotor coaxiale Drone

In this chapter, we conduct a detailed comparative analysis of various control design strategies applied to the dynamic model of a coaxial octorotor drone. The objective is to evaluate the performance, robustness, and disturbance rejection capabilities of each control method under different operational conditions.

The control techniques investigated include Proportional,Integral,Derivative (PID) control, Backstepping control, Sliding Mode Control (SMC), Adaptive (Direct) control, and Fuzzy Logic control. Each method is designed based on the nonlinear dynamics of the drone, aiming to achieve accurate trajectory tracking, stability, and disturbance resistance.

To ensure a fair comparison, the control gains for each technique are optimized using heuristic optimization algorithms, specifically Particle Swarm Optimization (PSO) and Genetic Algorithm (GA). This allows each controller to operate under its best possible configuration.
The chapter is structured into three main sections:

1. **Nominal Performance Evaluation:** where we analyze the response of each controller under ideal flight conditions.
2. **Perturbation Analysis:** where specific changes such as wind gusts, mass variation, and sensor noise are applied to evaluate the resilience and adaptability of each method.
3. **Robustness Tests:** where random disturbances are introduced to test the controller's ability to maintain performance under model uncertainties and external disruptions.

Through simulations and graphical comparisons, this chapter highlights the strengths and limitations of each control approach, providing insights into their suitability for real-world drone applications.

## 4.1 Nominal Performance

This section presents the behavior of the octorotor drone under ideal, disturbance-free conditions using each of the designed control strategies. The objective is to assess the baseline performance of the controllers in terms of trajectory tracking, stability, and control smoothness. The responses of the drone in position (X, Y, Z) and attitude (roll, pitch, yaw) are recorded for each controller, allowing for a direct visual and quantitative comparison. This serves as a reference point for later robustness and perturbation analyses.

### 4.1.1 PID controller

The PID controller is a simple linear control method, commonly used in many industrial applications due to its ease of implementation.

#### Results

The following figures present the simulation results of the coaxial octorotor. They show the drone's trajectory, orientation, and control performances under the proposed control strategy.

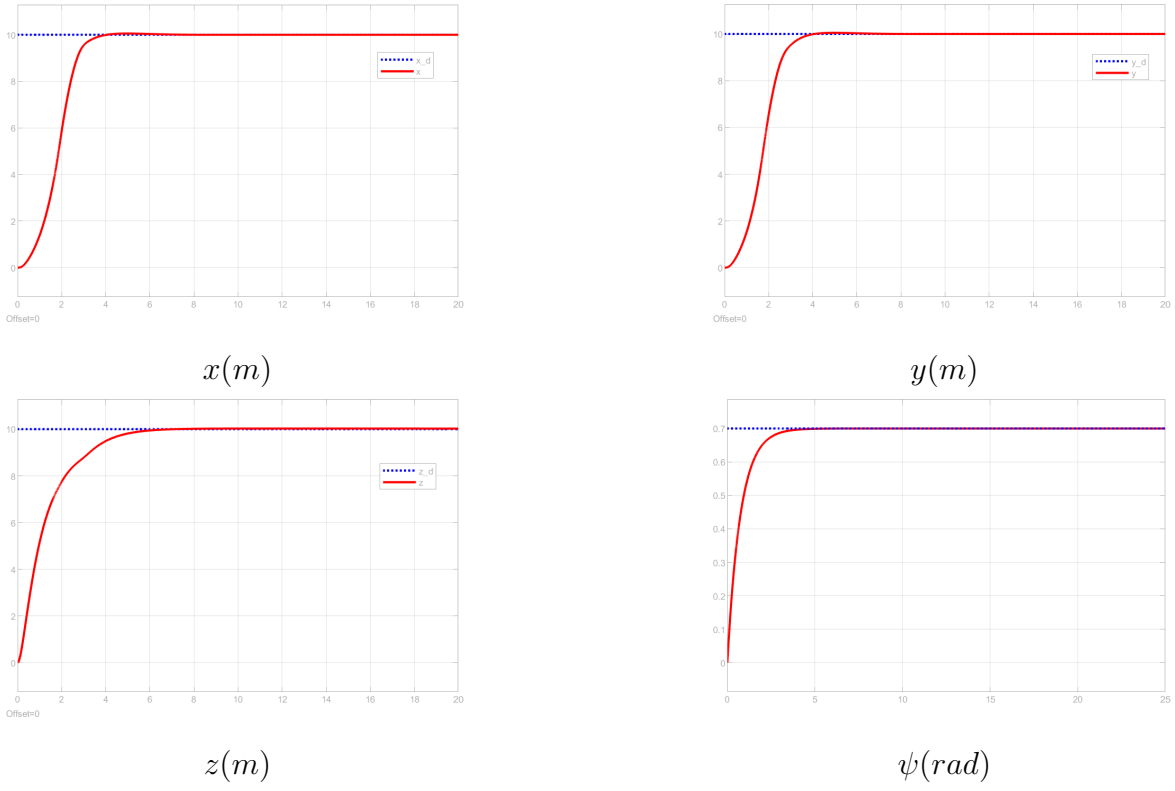


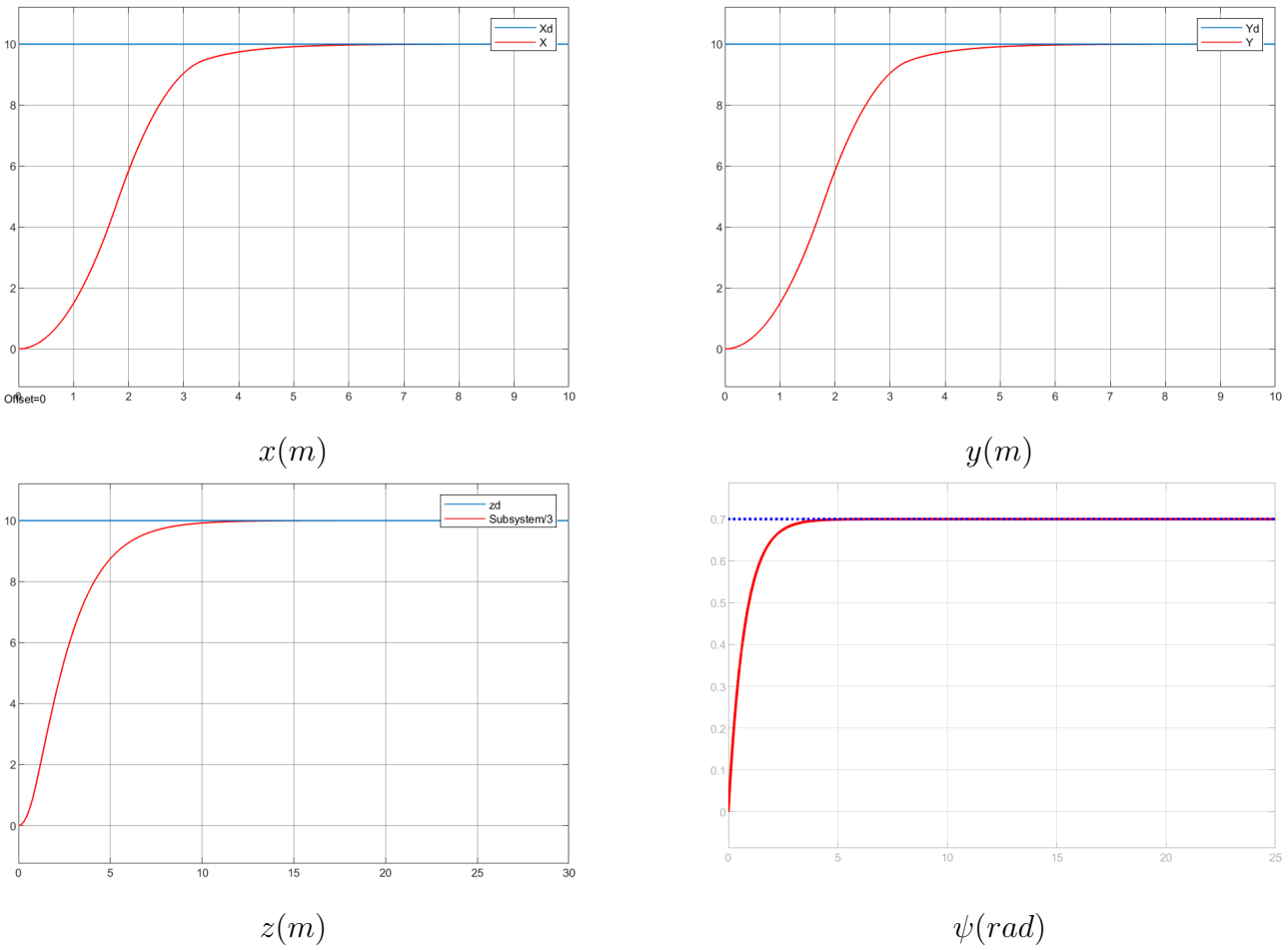

Figure 4.1: System response for performances evaluations (PID)

### 4.1.2 Second LYAPONOV

Backstepping is a nonlinear control approach that uses recursive design to stabilize each subsystem step by step.

#### Results

The following figures present the simulation results of the coaxial octorotor. They show the drone's trajectory, orientation, and control performances under the proposed control strategy.




$x(m)$ $y(m)$

$z(m)$ $\psi(rad)$

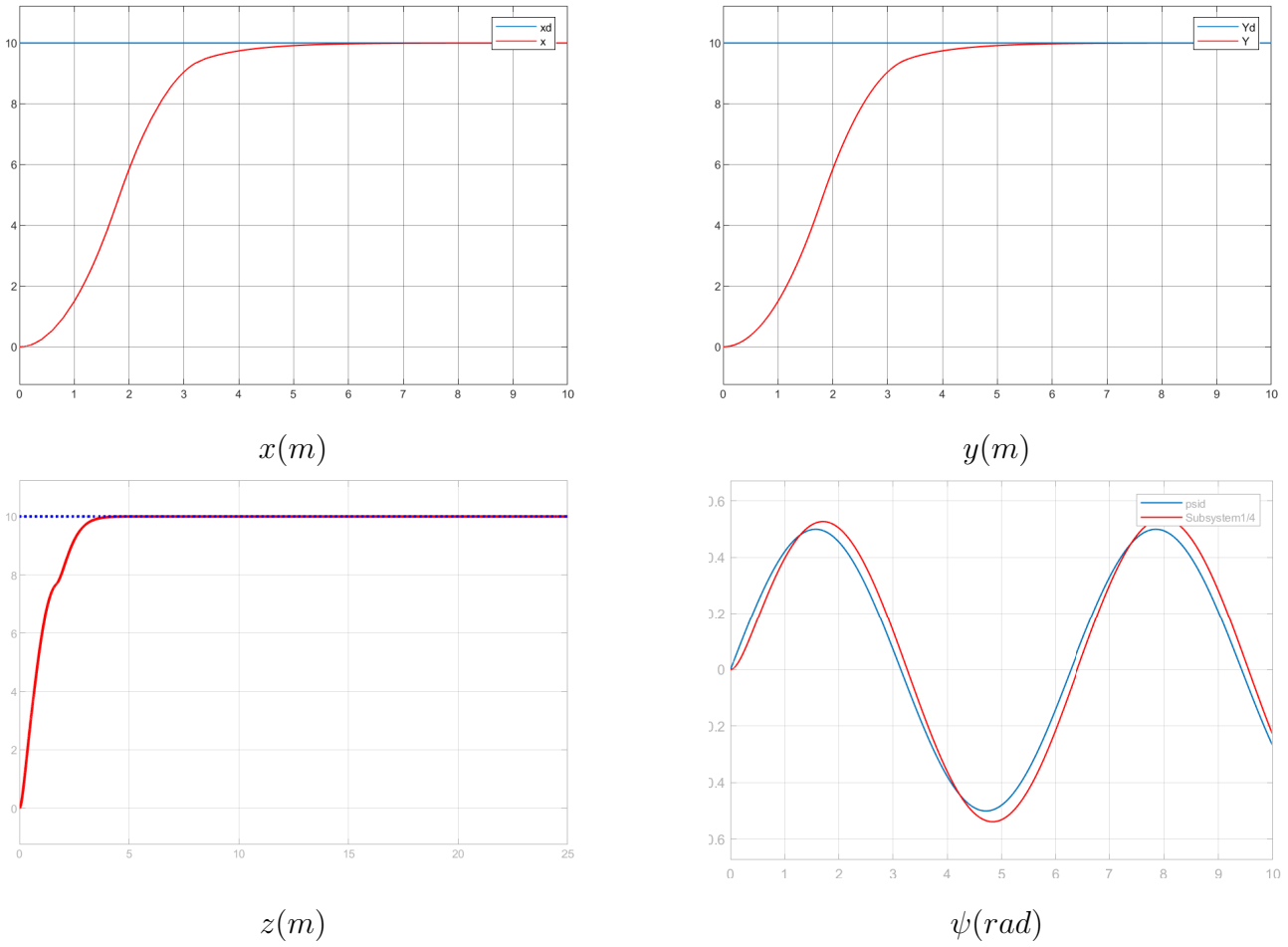
Figure 4.2: System response for performances evaluations (Backstepping)

### 4.1.3 Sliding Mode Control

SMC is a robust nonlinear method designed to force the system state onto a sliding surface for strong tracking and disturbance rejection.

#### Results

The following figures present the simulation results of the coaxial octorotor. They show the drone's trajectory, orientation, and control performances under the proposed control strategy.




$x(m)$ $y(m)$

$z(m)$ $\psi(rad)$

Figure 4.3: System response for performances evaluations (SMC)

### 4.1.4 Fuzzy Logic Control

The Fuzzy Logic Controller is not based on a mathematical model, which allows it to deliver interesting results. Below are the simulation results obtained using Simulink.

#### Results

The following figures present the simulation results of the coaxial octorotor. They show the drone's trajectory, orientation, and control performances under the proposed control strategy.

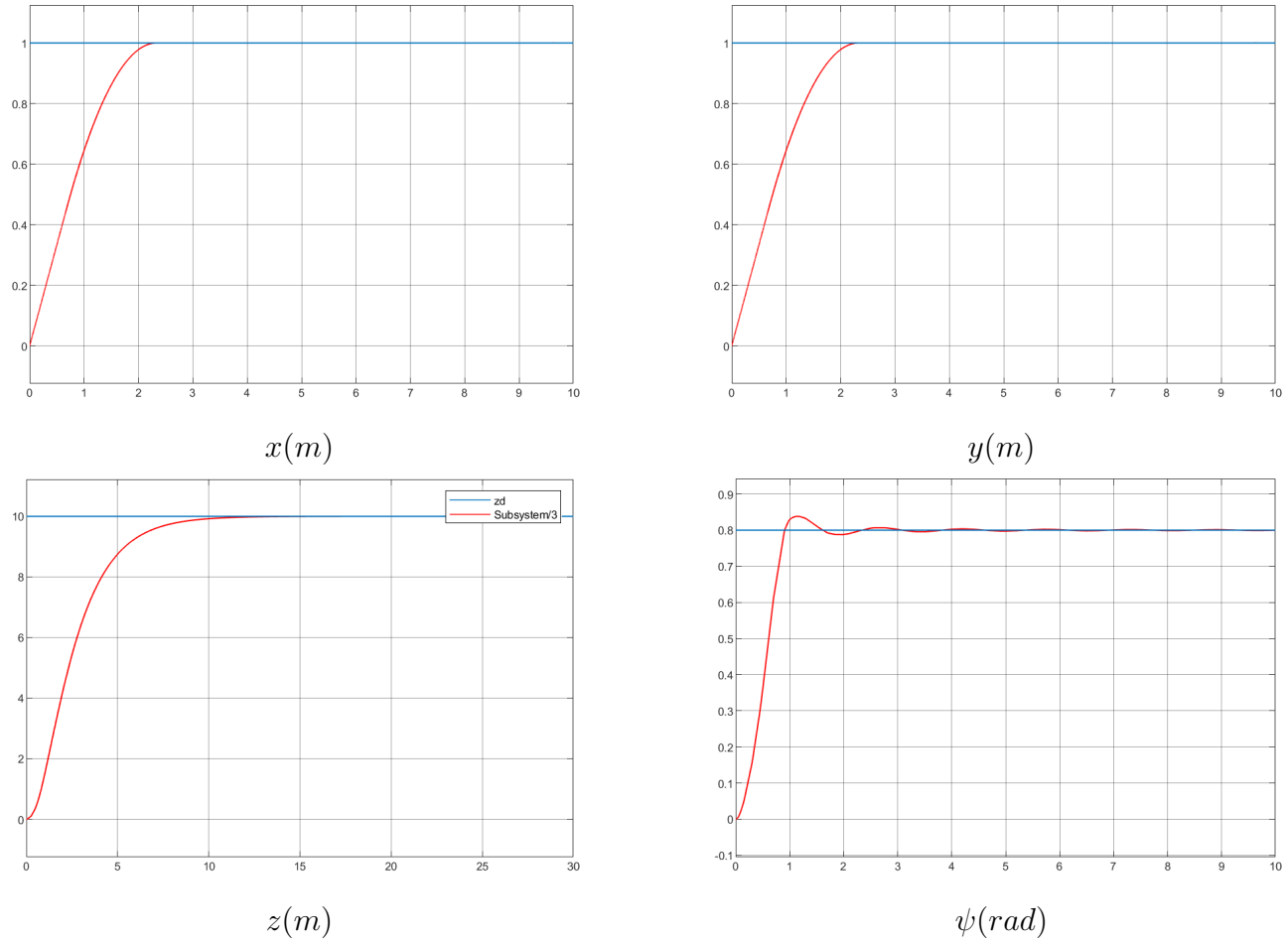



Figure 4.4: System response for performances evaluations (FLC)

## Tracking Accuracy

All the control approaches demonstrate high accuracy, with differences observed in their time response characteristics.

## Time Response

The time response refers to the duration a drone takes to reach and stabilize around its desired trajectory after a command is given. It is an important metric for evaluating the speed and efficiency of a control method. Faster time responses indicate quicker convergence, which is essential for dynamic tasks such as delivery or obstacle avoidance.

| **Control Method** | **Time Response (s) for 20%** |
|---|---|
| PID Controller | 2.5 |
| Backstepping | 2.4 |
| Sliding Mode Control | 2.6 |
| Fuzzy Logic Control | 1.35 |

Table 4.1: Time Response Comparison for Different Controllers

*Note: Time response includes the time to stabilize all main variables (x)*

### Conclusion

In an ideal environment, where external disturbances and parameter variations are minimal, all the tested control strategies deliver strong performance. The results show that each controller achieves accurate trajectory tracking, low overshoot, and acceptable time response. The PID controller offers a smooth control signal and is easy to tune, making it suitable for simple applications. The Fuzzy Logic Controller provides the fastest response and highest accuracy among the methods, along with a smooth control signal. Sliding Mode Control also ensures precise tracking but suffers from signal chattering. Backstepping performs well in terms of tracking but requires a more complex structure and produces less smooth control signals. Overall, under ideal conditions, each controller can achieve good performance, with differences mainly in response speed and signal quality.

## 4.2 Robustness tests

### 4.2.1 Input perturbation

This section is dedicated to evaluating the robustness of each control strategy when exposed to unexpected disturbances applied directly to the control signals of the octorotor drone. Specifically, a random signal is injected into the two control signals $U_2$ and $U_3$, which affect $\phi$ and $\theta$, and consequently impact the X and Y positions.

This scenario simulates real-world conditions such as actuator noise, environmental turbulence, and mechanical imperfections that can affect the effectiveness of the applied control signals. The objective is to assess each controller's ability to maintain stable and accurate tracking performance in the presence of such unpredictable perturbations.

Simulation results include time-domain plots of position, attitude, and tracking errors. These results highlight the resilience and fault tolerance of each method, offering insight into their suitability for practical applications where robustness is critical.

#### 4.2.1.1 PID Controller

The following figures present the simulation results of the coaxial octorotor. They show the drone's trajectory, orientation, control performances and the Robustenss under the proposed control strategy.

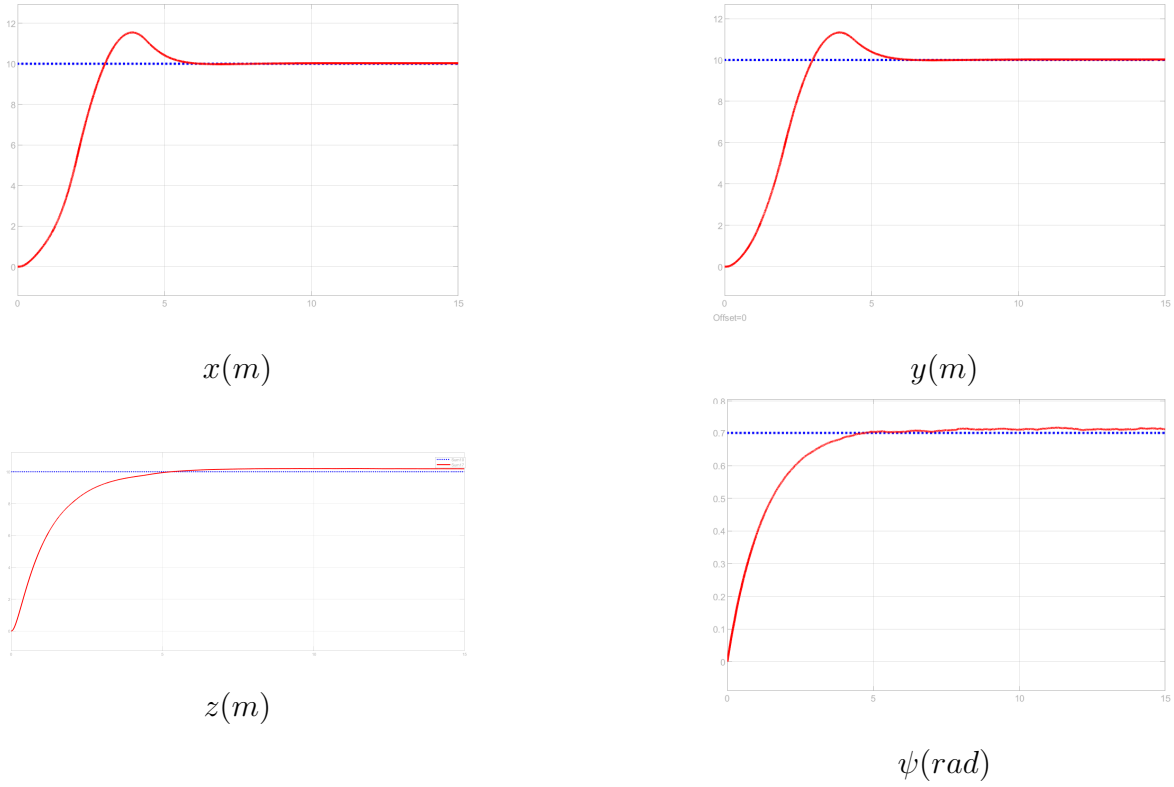



Figure 4.5: System response with input perturbation (PID)

## Comment:

We observe that the input perturbation may cause an overshoot of approximately 10%, which could adversely affect the drone's behavior. This overshoot may result in prolonged settling time in the step response, potentially compromising the system's responsiveness and stability.

#### 4.2.1.2 Second LYAPONOV

The following figures present the simulation results of the coaxial octorotor. They show the drone's trajectory, orientation, control performances and Robustness under the proposed control strategy.

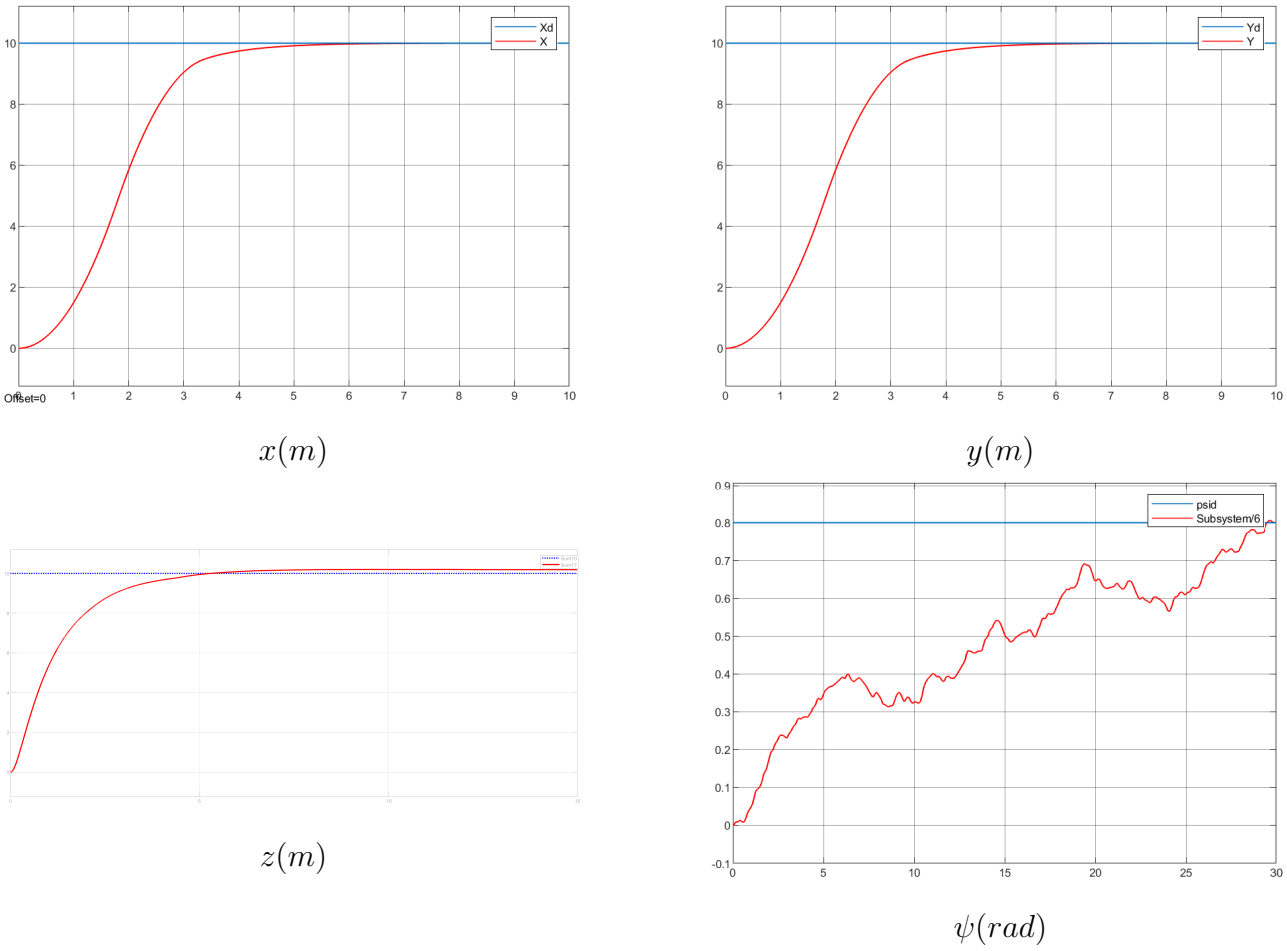



Figure 4.6: System response with input perturbation (Backstepping)

## Comment:

Backstepping maintains stability but begins to show performance degradation under control noise. Since the method does not include explicit robustness handling, noise propagates through the control law, causing oscillations.

#### 4.2.1.3 Sliding Mode Control

The following figures present the simulation results of the coaxial octorotor. They show the drone's trajectory, orientation, control performances and robustness under the proposed control strategy.

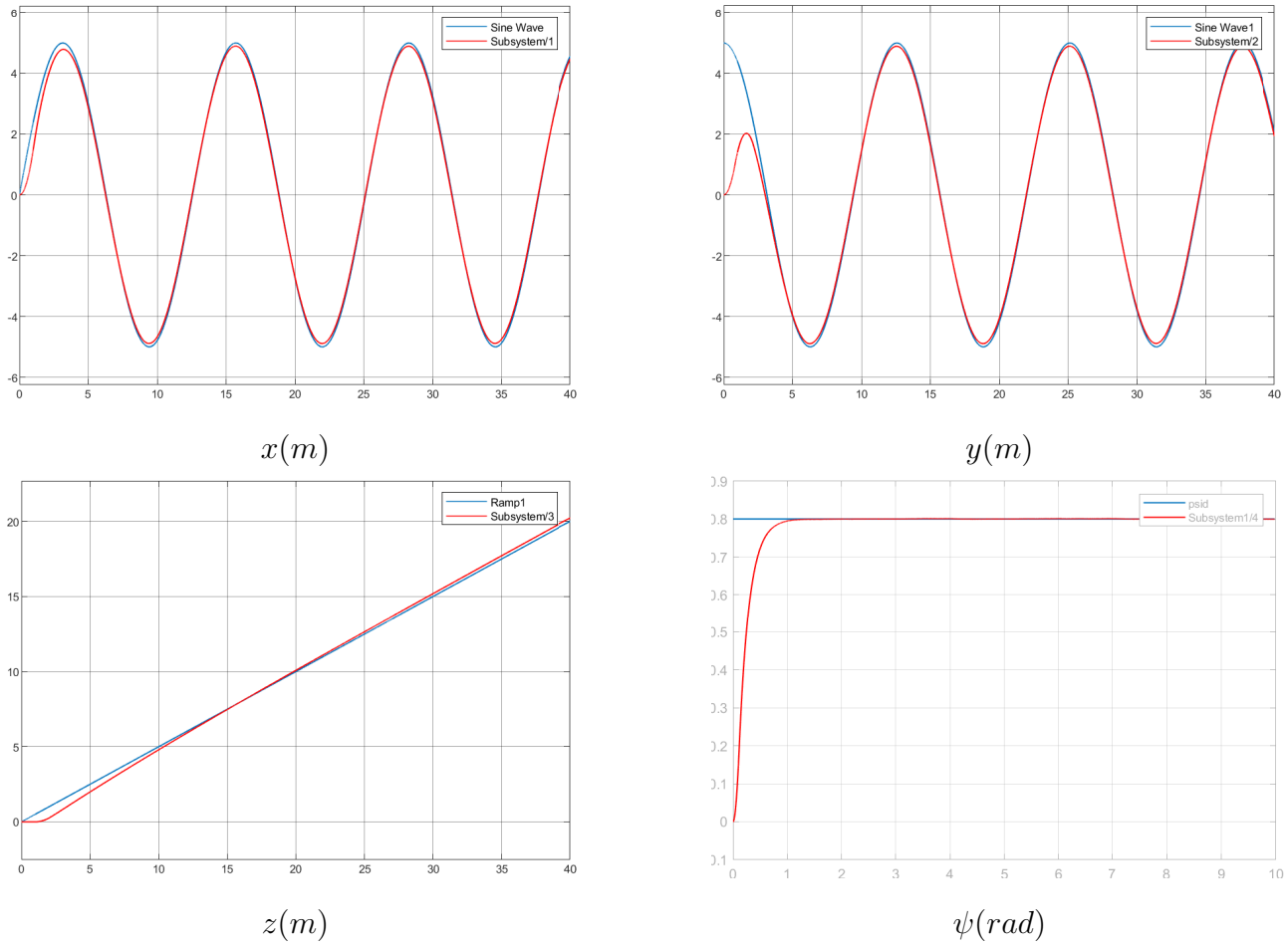



Figure 4.7: System response with input perturbation (SMC)

## Comment:

SMC shows strong robustness to random disturbances, as expected from its design. Its ability to drive system states to a sliding surface and maintain them there even in the presence of bounded uncertainties makes it highly effective. Chattering remains the main drawback.

#### 4.2.1.4 Fuzzy Logic Control

The following figures present the simulation results of the coaxial octorotor. They show the drone's trajectory, orientation,control performances and robustness under the proposed control strategy.

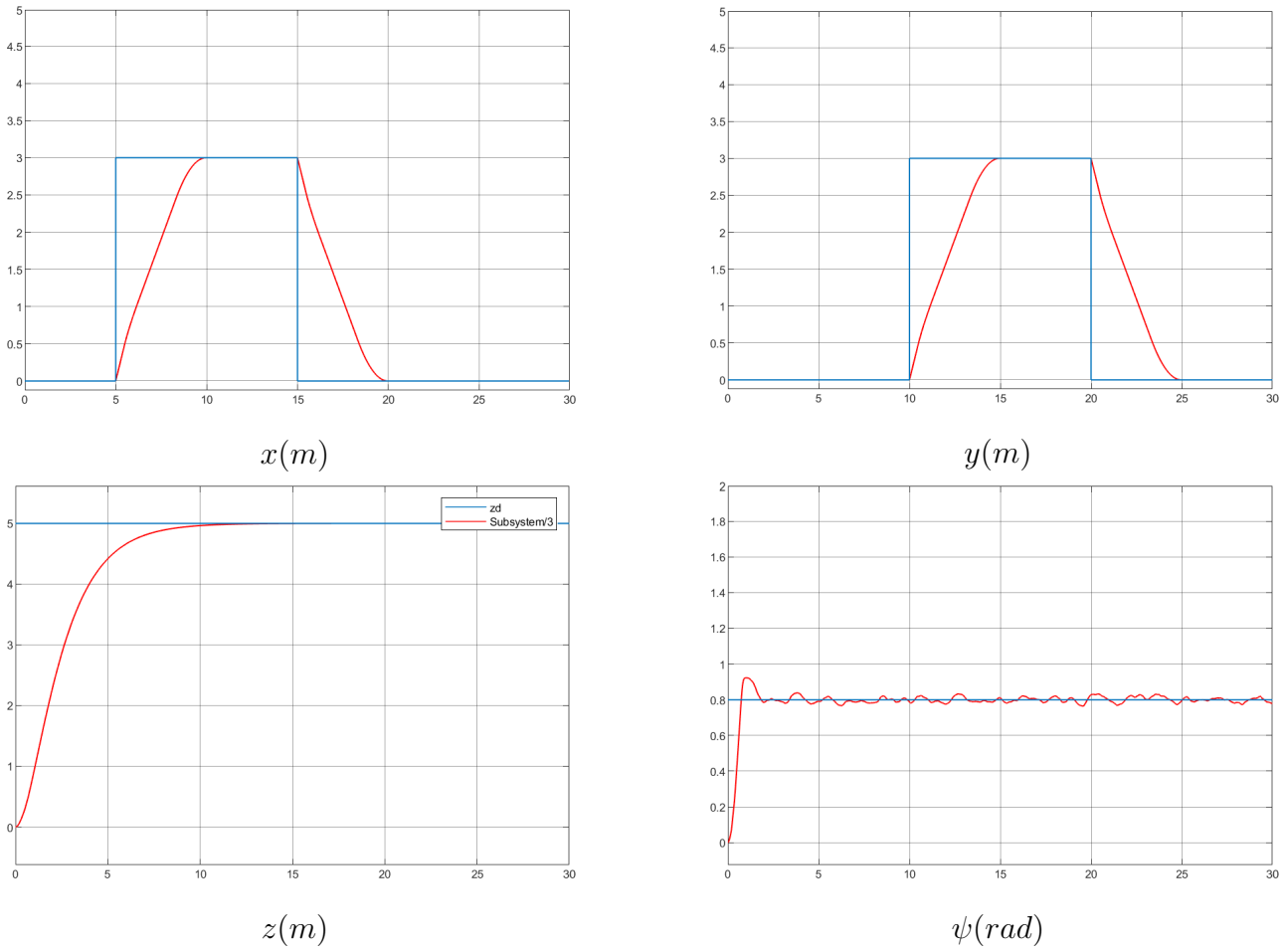



Figure 4.8: System response with input perturbation (FLC)

## Comment:

The Fuzzy Logic Controller (FLC) provides promising results, comparable to those achieved by Sliding Mode Control (SMC), due to its ability to adapt to different situations and its reliance on system error rather than a precise model. Most importantly, FLC generates smooth control signals, which is a critical advantage for practical implementation, as it significantly reduces energy consumption and extends the operational life of DC motors.

## Conclusion

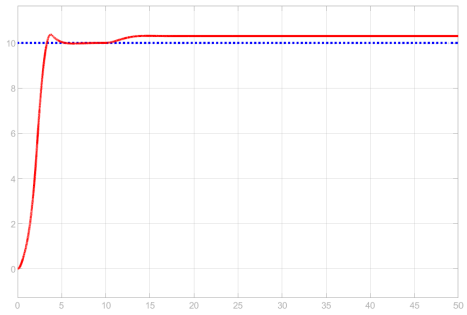
Under the influence of random disturbances injected into control signals, major differences between controllers became evident. The PID controller showed significant degradation, with poor resistance to control noise. Backstepping maintained stability but experienced oscillations. SMC demonstrated strong robustness, with SMC excelling in maintaining tight tracking despite disturbances. Fuzzy Logic handled noise moderately well. In summary, SMC stood out as the most robust approaches under uncertain and noisy environments.

### 4.2.2 System Parameters Variations(Parametric robustness)

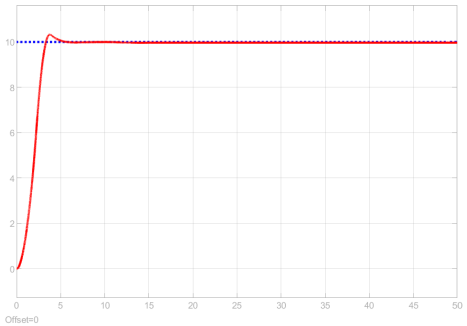
In this second part of the robustness test, we evaluate the performance of the controller under variations in the system parameters. This analysis helps assess the controller's ability to maintain stability and performance despite changes in the physical properties of the system, such as mass or inertia.

#### 4.2.2.1 PID Controller

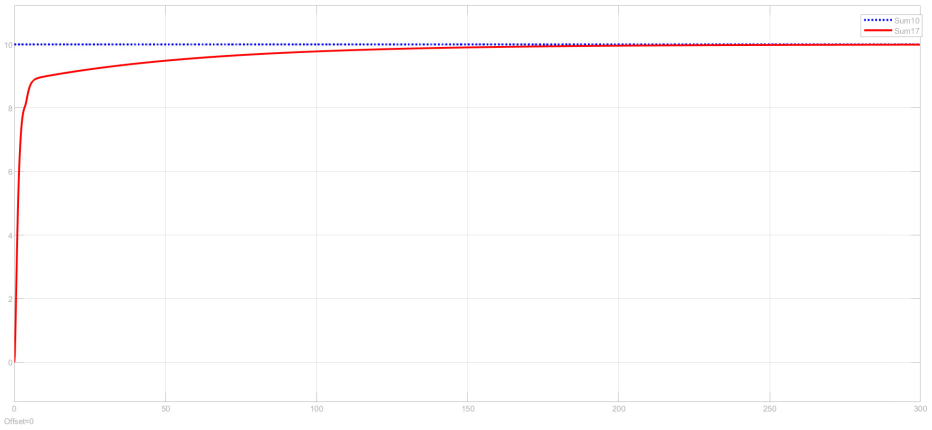
Here are the results produced by the PID controller when the weight changes to $m = 40\,\text{kg}$.



$x(m)$ en fonction de $t(s)$



$y(m)$ en fonction de $t(s)$



$z(m)$ en fonction de $t(s)$

Figure 4.9: System response with parameter variations (PID)

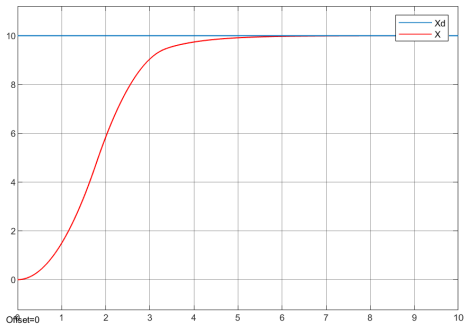
## Comment:

We can observe that the change in weight does not significantly affect the $x$ and $y$ axes, but it has a noticeable impact on the $z$-axis, and we see that :

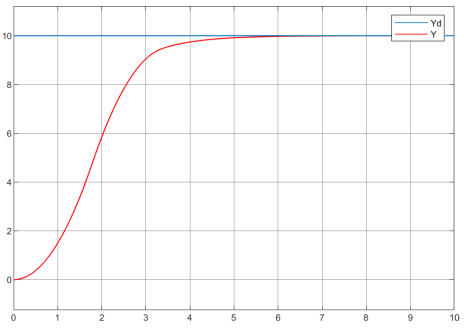
$$T_r > 100s \tag{4.1}$$

#### 4.2.2.2 backstepping method

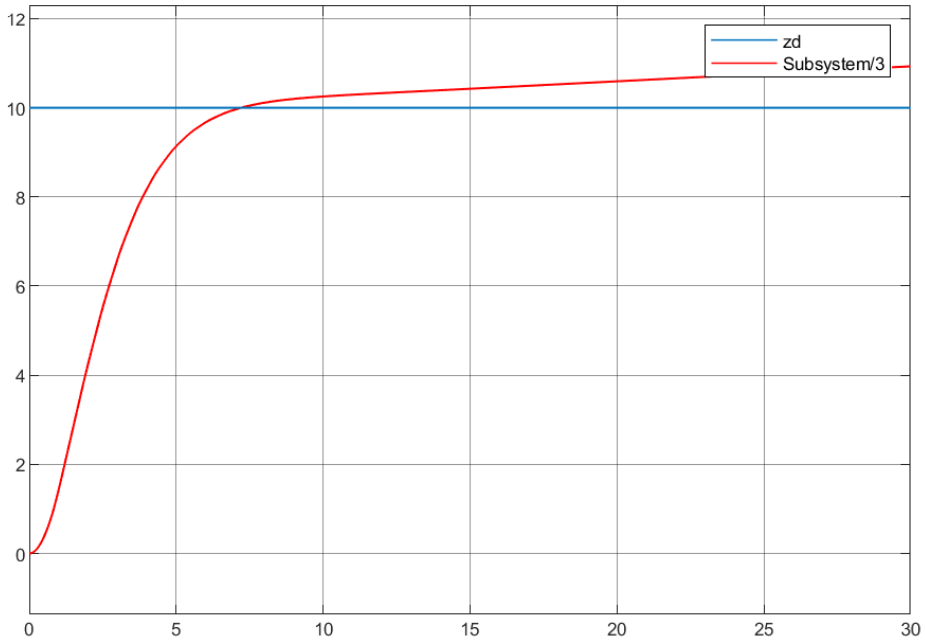
Here are the results produced by the 2nd method when the weight changes to $m = 40\,\text{kg}$.




(a) $x(m)$ en fonction de $t(s)$




(b) $y(m)$ en fonction de $t(s)$




(c) $z(m)$ en fonction de $t(s)$

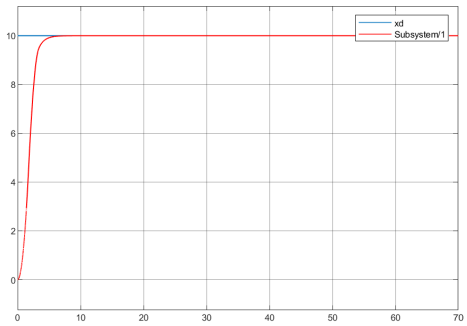
Figure 4.10: Réponse du système avec variations de paramètres (Backstepping)

## Comment:

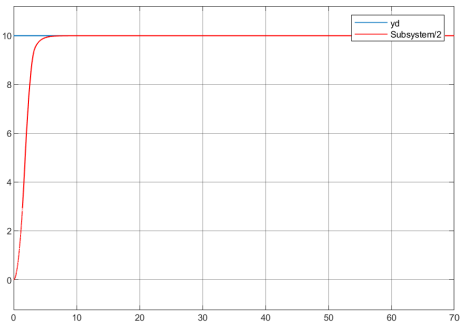
We observe that changes in weight have little effect on the $x$ and $y$ axes, but they significantly impact the $z$-axis. When the weight changes, the system becomes unstable and diverges. This indicates that the control strategy is not robust and is therefore unsuitable for applications like agriculture or delivery, where load variation is common.

#### 4.2.2.3 Sliding mode method

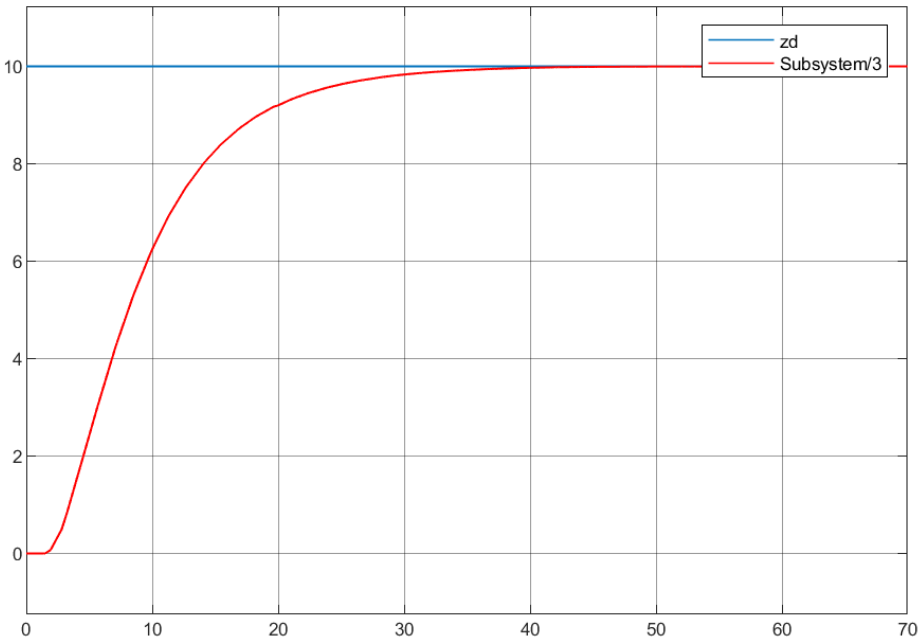
Here are the results produced by the Sliding Mode Control (SMC) controller when the weight changes to $m = 40\,\text{kg}$.




(a) $x(m)$ en fonction de $t(s)$




(b) $y(m)$ en fonction de $t(s)$




(c) $z(m)$ en fonction de $t(s)$

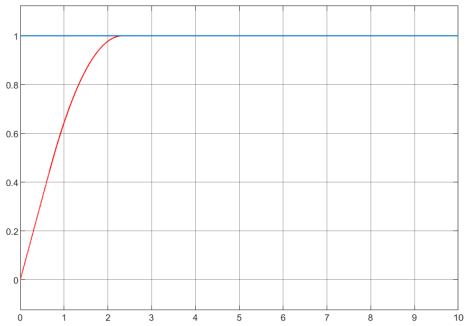
Figure 4.11: System response with parameter variations (Sliding Mode Control (SMC))

## Comment:

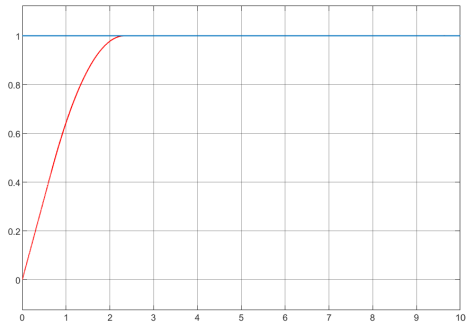
The Sliding Mode Controller (SMC), due to its robustness, can handle system parameter variations effectively and continue to deliver reliable performance. It maintains good control even when the system's weight changes, ensuring stability and accuracy, although it may exhibit a relatively high time response.

#### 4.2.2.4 Fuzzy Logic Control

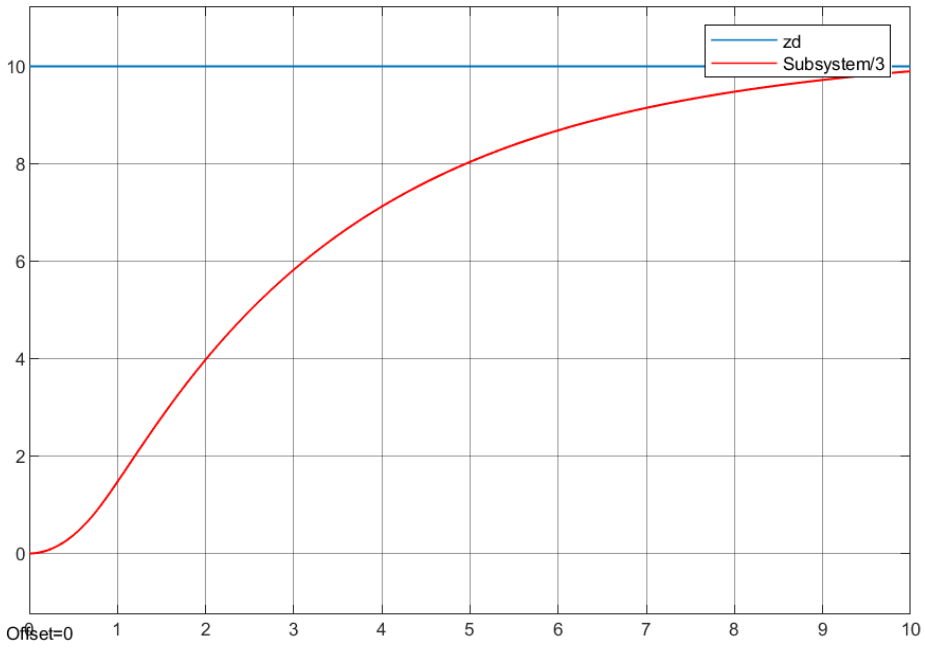
Here are the results produced by the Fuzzy Logic Control (FLC) controller when the weight changes to $m = 40\,\text{kg}$.



(a) $x(m)$ en fonction de $t(s)$



(b) $y(m)$ en fonction de $t(s)$




(c) $z(m)$ en fonction de $t(s)$

Figure 4.12: System response with parameter variations (Fuzzy Logic Control (FLC))

## Comment:

The Fuzzy Logic Controller (FLC), as it does not rely on an accurate system model or precise parameter values, is capable of maintaining consistent performance even when the system mass remains fixed. It provides acceptable time response compared to the previous controller.

## Conclusion

SMC delivers the best performance due to its high robustness and insensitivity to parameter variations. FLC also performs well but with some delay and a slower response time. Other control approaches did not yield satisfactory results

## 4.3 discussion and Comparison of results

This section presents a comprehensive synthesis of the performance of the four control strategies PID, Backstepping, Sliding Mode Control (SMC), and Fuzzy Logic Control as applied to an octorotor coaxial drone. Drawing from the results of the nominal performance, robustness tests (with random disturbances on control signals,variation of parametters), we aim to:

- Compare each controller's tracking accuracy, stability, and disturbance rejection ability.
- Highlight the trade-offs between simplicity, robustness, and adaptability.
- Provide a clear ranking or recommendation based on overall behavior across different test conditions.

The goal is not only to assess which controller performs best in ideal scenarios, but also to identify which ones are reliable and effective in real-world conditions where uncertainties and perturbations are inevitable. This comparative discussion will help guide future controller selection or improvements for drone systems operating in complex environments.

### Comparison

To compare the advantages and disadvantages of each control method, we present a summary table based on the results observed throughout this chapter. The comparison includes metrics such as RMSE, time response, overshoot, robustness, and control signal smoothness (related to energy and actuator efficiency).

| **Method** | **Tracking** | **Time Response** (20%) | **Overshoot** (with our gains) | **Robustness** | **Control Signal** | **Implement Complexity** |
|---|---|---|---|---|---|---|
| PID Controller | High | 2.5 | Low | Low | Smooth | Easy |
| Backstepping | High | 2.4 | Low | Medium | Less Smooth | Complex |
| Sliding Mode Control | High | 2.6 | Low | Very High | Chattering | Moderate |
| Fuzzy Logic Control | Very High | 1.35 | Low | Very High | Smooth | Moderate |

Table 4.2: Comparison of Control Methods

*Note: "Smooth" control signals reduce energy consumption and hardware stress, while "Chattering" indicates high-frequency oscillations that may affect actuator longevity.*

## 4.4 Conclusion:

Based on the results presented in Table 4.2, we can draw several conclusions regarding the four control methods studied. The Backstepping controller, while theoretically effective, does not produce significantly better results and is relatively complex to implement. Additionally, it requires higher control effort, making it less suitable for practical applications.

The Sliding Mode Control (SMC) shows strong robustness and satisfactory performance. However, the presence of chattering leads to increased energy consumption and may reduce the lifespan of motors due to high-frequency switching.

On the other hand, the PID controller and the Fuzzy Logic Controller (FLC) offer smoother control signals. The PID controller is very easy to implement and performs well in stable environments, such as indoor applications, despite its limited robustness. Therefore, it can be considered the most practical choice under favorable conditions.

The Fuzzy Logic Controller, while moderately complex to implement, provides very high robustness and fast response, as it does not rely on an accurate mathematical model. For outdoor applications or complex environments with variable payloads and external disturbances, the FLC is a more suitable solution due to its adaptability and reliability.

# General Conclusion

This project focused on the modeling and control of a coaxial octorotor drone. We began by developing a comprehensive mathematical model to describe the drone's dynamics, which served as the basis for simulating and evaluating various control strategies using MATLAB.

Four control methods were designed and simulated: the classical PID controller, the nonlinear Backstepping controller, Sliding Mode Control (SMC), and the Fuzzy Logic Controller (FLC). These methods were compared based on tracking accuracy, response time, overshoot, and the quality of the control signal. The simulation results showed that all four controllers provided satisfactory performance under ideal conditions, with each method exhibiting its own advantages and limitations.

We also explored more advanced approaches such as adaptive PID, adaptive SMC, adaptive Backstepping, and neural network–based controllers. While these strategies offer better adaptability and precision, they introduce higher complexity and computational cost, which can be a barrier to real-time implementation.

Although the scope of this work was limited to simulation, it lays a strong foundation for future practical implementation. Within the academic framework of our school, this project illustrates how theoretical knowledge in control systems can be applied to realistic and complex aerial platforms. It also opens promising avenues for future experimentation in the lab.

Our perspective is to achieve the physical realization of this promising drone within our school's laboratory, or at least to implement our work in a laboratory equipped with this type of drone. This will allow us to validate and compare the developed control methods in real-world conditions. Additionally, we aim to design more advanced and innovative control strategies that enhance performance and minimize energy consumption, first through simulations and then by enforcing their validation through real-world implementation.

# Chapter 5

# Business Model Canvas (BMC)

---

**Key Partners** **Key Activities** **Value Proposition**

---

- ROS2 & Gazebo open-source platforms
- Hardware suppliers (LiDAR, cameras)
- Research labs and universities
- Cloud service providers
- Robotics integrators • Develop and improve mapless navigation
- DRL-based control algorithm training
- Simulation and real-world testing
- Technical support and updates • Mapless navigation without SLAM
- Easy integration with ROS2 robots
- Smart and adaptive control using DRL
- Fast setup in new environments
- Plug-and-play solution with support

---

**Customer Relationships** **Customer Segments** **Channels**

---

- Personalized technical support
- Online documentation and training
- Co-development with pilot clients
- Community support (GitHub, forums) • Logistics warehouses
- Hospitals (delivery robots)
- Hotels (room service robots)
- Indoor/outdoor delivery startups • Website and online contact
- Robotics conferences and webinars
- GitHub and ROS community
- Integrator/distributor network

---

**Key Resources** **Cost Structure** **Revenue Streams**

---

- Robots, sensors, simulation tools
- Software and control algorithms
- Skilled team (robotics, AI, ROS)
- Training and support content • Development team salaries
- Hardware and testing costs
- Server infrastructure

- Marketing and outreach • Software license per robot/fleet
- SaaS subscription (updates, metrics)
- Paid add-ons (multi-robot, custom features)
- Training and support packages

---

# Bibliography


[1] Linker Criollo, Carlos Mena-Arciniega, and Shen Xing. Classification, military applications, and opportunities of unmanned aerial vehicles. *Aviation*, 2024.

[2] Wayne Ong, Sutthiphong Srigrarom, and Henrik Hesse. Design methodology for heavy-lift unmanned aerial vehicles with coaxial rotors. January 2019.

[3] Payam Zarafshan and Seyed Jamal Haddadi. Dynamics modelling and implementation of an attitude control on an octorotor. *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE).*, 2015.

[4] ZDaniel Alshamaa, Francis Clovis, Majd Saied, and Hassan Shraim. Model identification and validation for translational movements of an octorotor uav. *Education and Development of Unmanned Aerial Systems (RED-UAS)*, 2015.

[5] Payam Zarafshan and Seyed Jamal Haddadi. Attitude control of an autonomous octorotor. *IEEE International Conference on Robotics and Mechatronics (ICRoM 2014)*, 2015.

[6] Roland Siegwart and Samir Bouabdallah. Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. *2005 IEEE International Conference on Robotics and Automation*, 2005.

[7] Ahmad MohdAziz Hussein, Saleh Ali Alomari, Mohammad H. Almomani, Raed Abu Zitar, Hazem Migdady, Aseel Smerat, Vaclav Snasel, and Laith Abualigah. A hybrid pso-gcra framework for optimizing control systems performance. *International Journal of Robotics and Control Systems*, 2025.

[8] Shaobo Zhang, Mei Li, Jinsong Li, Jie Xu, Zelong Wang, and Shuaihang Liu. Research on ride comfort control of air suspension based on genetic algorithm optimized fuzzy pid. 2024.

[9] Samir Bouabdallah, Andre Noth, and Roland Siegwart. Pid vs lq control techniques applied to an indoor micro quadrotor. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.

[10] Nada Braiek, Lotfi Chaouech, and Imen Saidi. Fuzzy backstepping quadrotor control: balancing precision and energy efficiency. *International Journal of Dynamics and Contro*, 2025.

[11] Alexander Poznyak and Vadim Utkin. Road map for sliding mode control design. September 2020.

[12] Ali Medjghoua, Noureddine Slimaneb, and Kheireddine Chafaaa. Fuzzy logic controller using the nonholonomic constraints for quadrotor trajectory tracking. *Revue des sciences et Science de l'ingénieur*, 2018.

[13] Ali Medjghoua, Noureddine Slimaneb, and Kheireddine Chafaaa. Fuzzy logic controller using the nonholonomic constraints for quadrotor trajectory tracking. *Revue des sciences et Science de l'ingénieur*, 2018.

[14] Zamoum Yasmine, Baiche Karim, Boushaki Razika, and Benrabah Younes. Adaptive fuzzy logic control of quadrotor. *International Journal of Robotics and Control Systems (IJRCS)*, 2024.