الجمهورية الجزائرية الديمقراطية الشعبية
Peoples Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and scientific research
المدرسة الوطنية العليا للتكنولوجيا
National High School of Technology
Department of Industrial Engineering and Maintenance

Final Year Project to Obtain the Diploma of Engineering

Field:
Electromechanics

Specialty:
Mechatronics

Theme:

Contribution to quadrotor modeling and attitude stabilization based on an adaptive flight control strategy

Realized by
- CHITER Oussama
- TALI Nazim

Members of The Jury :

President	Dr. HOUARI Fouad
Supervisor	Dr. BOUADI Hakim (A)
Examinator	BOUCHACHI Islem

Algiers, the 25/06/2023

Academic year 2022 - 2023

Abstract :

A quadcopter drone is a complex mechatronic system. This study highlights the recent developments in UAVs, with a particular focus on quadruple-type multi-rotor UAVs. These UAVs have gained attention due to their suitable performance, wide applications, advantages in hover ability, maneuverability and low cost of implementation. However, designing an efficient controller to ensure the robust flight capabilities of the quadrotor remains a challenge. This work addresses these challenges by presenting a comprehensive approach to the dynamic modeling and control design of a quadrotor. First, an overview of the UAV configuration, a review of the literature on quadrotor control methods, and in particular adaptive control. Second, we explain the details of modeling the quadrotor drone, including the mathematical model derived using Euler-Newton formalism. Also, we design the control laws to stabilize and track using PID and Backstepping controllers. Simulations were performed using MATLAB Simulink for various controllers scenarios to validate the results. Finally, we make a scenario where a quadrotor has an asymmetric geometry and varying mass, using a control approach with an adaptive method to estimate unknown system parameters. The research concludes with an evaluation of the work and offers perspectives for future research.

Keywords: UAV, control, quadrotor, modeling, simulation, PID, Backstepping, Adaptive flight control.

Résumé :

Un drone Quadcopter est un système mécatronique complexe. Cette étude met en lumière les développements récents des drones, avec un accent particulier sur les drones multi-rotors de type quadruple. Ces drones ont attiré l'attention en raison de leurs performances appropriées, de leurs applications étendues, de leurs avantages en termes de capacité de vol stationnaire, de maniabilité et de leur faible coût de mise en œuvre. Cependant, concevoir un contrôleur efficace pour assurer les capacités de vol robustes du quadrirotor reste un défi. Ce travail aborde ces défis en présentant une approche globale de la modélisation dynamique et de la conception de contrôle d'un quadrirotor. Tout d'abord, un aperçu de la configuration du drone, une revue de la littérature sur les méthodes de contrôle du quadrirotor, et en particulier le contrôle adaptatif. Deuxièmement, nous expliquons les détails de la modélisation du drone quadrirotor, y compris le modèle mathématique dérivé à l'aide du formalisme d'Euler-Newton. En outre, nous concevons les lois de contrôle pour stabiliser et suivre à l'aide de contrôleurs PID et Backstepping. Des simulations ont été réalisées à l'aide de MATLAB Simulink pour différents scénarios de contrôleurs afin de valider les résultats. Enfin, nous créons un scénario où un quadrirotor a une géométrie asymétrique et une masse variable, en utilisant une approche de contrôle avec une méthode adaptative pour estimer des paramètres système inconnus. La recherche se conclut par une évaluation des travaux et offre des perspectives pour des recherches futures.

Mots clés: UAV, contrôle, quadrotor, modélisation, simulation, PID, Backstepping, Adaptive flight control.

ملخص :

الطائرة بدون طيار هي نظام ميكاترونيك معقد. تسلط هذه الدراسة الضوء على التطورات الأخيرة في الطائرات بدون طيار، مع التركيز بشكل خاص على الطائرات بدون طيار من النوع الرباعي. اكتسبت هذه الطائرات بدون طيار الاهتمام بسبب أدائها المميز، والتطبيقات الواسعة والمزايا في القدرة على التحليق والتكلفة المنخفضة للتنفيذ. ومع ذلك، فإن تصميم وحدة تحكم فعالة الطائرة بدون طيار رباعية المحرك لا يزال يمثل تحديًا. يعالج هذا العمل هذه التحديات من خلال تقديم نهج شامل للنمذجة الديناميكية وتصميم التحكم في رباعي المحركات. أولاً ، نظرة عامة على تكوين الطائرات بدون طيار واعطاء بعض الطرق للتحكم في الرباعي المحرك، بالاخص التحكم التكيفي. ثانيًا ، نشرح تفاصيل نمذجة الطائرة بدون طيار الرباعية المحرك، بما في ذلك النموذج الرياضي المشتق باستخدام شكلية أويلر ونيوتن. أيضًا ، قمنا بتصمم قوانين التحكم لتحقيق الاستقرار والتتبع باستخدام وحدات التحكم بالانحراف التناسبي والتكاملي والتفاضلي و التحكم بالتراجع. تم إجراء عمليات المحاكاة باستخدام برنامج ماطلاب لمختلف السيناريوهات لوحدات التحكم المختلفة للتحقق من صحة النتائج. أخيرًا، قمنا بعمل سيناريو حيث يكون للرباعي هندسة غير متماثلة وكتلة متفاوتة، باستخدام نهج تحكم مع طريقة تكيفية لتقدير معلومات النظام الغير المعروفة. ويختتم البحث بتقييم للعمل ويقدم منظورات للبحث في المستقبل.

كلمات مفتاحية : التحكم، طائرة رباعية المحركات، نمذجة، التحكم بالانحراف التناسبي و التكاملي و التفاضلي، التحكم بالتراجع، التحكم التكيفي.

Acknowledgments

First and foremost, we would like to thank the Almighty God, the All-Powerful and Merciful, who granted us the strength and patience to complete this humble work.

We would like to express our heartfelt gratitude to our supervisor, **Dr. Hakim BOUADI**, for his availability, encouragement, numerous beneficial interventions, and invaluable assistance that enabled us to complete this work.

We sincerely thank the president and members of the jury for their interest in this project and for honoring us by agreeing to judge and evaluate our work.

Our sincere thanks go to all our teachers and the administration of the National School of Technology for enabling us, in one way or another, to accomplish our work.

We would like to express our gratitude to our families and friends for their constant concern about the progress of our work.

A special thanks to those who have always believed in us and encouraged us to successfully finish this project.

Dedication

To my dear mother, father, siblings, and my young nephews,

Thank you for your boundless support and love throughout my educational journey. It is because of both of you, Mom and Dad, that I have been able to achieve my dreams and graduate today. You are the main source of strength and inspiration in my life. Let this graduation be a great dedication to both of you.

To my dear brothers and sisters, I hope you feel immense pride for being members of our family. We have faced numerous challenges and adventures together, but you have always been here to support and encourage me. I appreciate the love and cooperation that unites us, and I look forward to a bright future that brings us closer together.

To my young nephews, you are a true joy in my life. I have witnessed your growth and development, and you have always been an amazing source of laughter and fun. You are an integral part of my life, and I will always be here to support and encourage you.

I dedicate this graduation memoir to each one of you, as a memento of the love and appreciation I hold in my heart for my beloved family. Thank you for everything, and I wish us all a bright future filled with success and happiness.

With sincere love and gratitude,

OUSSAMA

Dedication

To my beloved father, mother, siblings, and the little children of my brother,

I dedicate this memoir to express my deepest gratitude and appreciation for your unwavering love, support, and encouragement throughout my educational journey. Your presence in my life has been a constant source of strength and inspiration.

To my father, thank you for your guidance, wisdom, and sacrifices. Your unwavering belief in me and your relentless pursuit of my dreams have shaped the person I am today. I am forever grateful for your unconditional love and the values you have instilled in me.

To my mother, your boundless love, nurturing care, and selflessness have been my guiding light. You have been my rock, my confidante, and my source of comfort. Your unwavering faith in me has given me the confidence to overcome challenges and reach for the stars.

To my dear siblings, you have been my companions, my best friends, and my pillars of support. We have shared countless memories, laughter, and tears. Your presence in my life has made it richer, and I am grateful for the bond we share. Together, we have weathered storms and celebrated victories, and I cherish each and every moment.

To the little children of my brother, you bring immense joy and laughter to our lives. Your innocence, curiosity, and pure hearts remind me of the beauty of life. As you embark on your own educational journey, know that I am here to support and guide you every step of the way. May you always dream big and achieve greatness.

This memoir is a testament to the love and gratitude I hold in my heart for each and every one of you. Thank you for being my constant source of love, strength, and inspiration. Our bond is unbreakable, and I am forever blessed to have such an incredible

NAZIM

Contents

General introduction **1**

1 State of the Art **3**

1.1 Introduction 4

1.2 Classification of drones 4

1.2.1 According to the aerodynamic shape 4

1.2.2 Depending on the landing method 4

1.2.3 Based on Weight and Range 5

1.3 Drone application area 5

1.4 Advantages and disadvantages of drones 8

1.4.1 Advantages 8

1.4.2 Disadvantages 8

1.5 Description and composition of the quadrotor 9

1.5.1 Frame 9

1.5.2 Motors 9

1.5.3 The propellers 11

1.5.4 Variable speed drives 11

1.5.5 Battery 12

1.5.6 Flight controller 12

1.5.7 The sensors 13

1.6 Quadrotor control 14

1.6.1 Linear controllers . . . 14
1.6.2 Non-linear controllers . . . 15
1.6.3 Intelligent controllers based on learning . . . 16
1.6.4 State of the art of the adaptive flight controller . . . 16
1.7 Conclusion . . . 24

2 Quadrotor Dynamics Modeling 25
2.1 Introduction . . . 26
2.2 Reference frames and passage matrices . . . 26
2.3 Kinematic modeling . . . 29
2.4 Dynamic modeling . . . 30
2.4.1 Translation dynamics . . . 30
2.4.2 Rotational dynamics . . . 31
2.5 Forces and moments acting on the quadrotor . . . 32
2.5.1 The forces . . . 32
2.5.2 The torques . . . 35
2.5.3 Gyroscopic effect . . . 37
2.6 Rotor dynamics . . . 38
2.7 Quadrotor model . . . 39
2.7.1 Modeling with the Newton-Euler formalism . . . 39
2.7.2 Modeling with the Euler-Lagrange formalism . . . 41
2.8 Conclusion . . . 44

3 Control design techniques 46
3.1 Introduction . . . 47
3.2 General command structure . . . 47
3.3 Command synthesis model . . . 48
3.4 PID controller . . . 50
3.4.1 Proportionnel . . . 50

3.4.2 Integrator . . . 51
3.4.3 Derivator . . . 51
3.4.4 Objective of the command . . . 52
3.4.5 Control laws . . . 53
3.5 PID control implementation . . . 54
3.5.1 Implementation . . . 55
3.5.2 Trajectory tracking . . . 56
3.6 Backsepping controller . . . 67
3.7 Backstepping control implementation . . . 73
3.7.1 Implementation . . . 73
3.7.2 Trajectory tracking . . . 74
3.7.3 Simulation results . . . 84
3.8 Conclusion . . . 86

4 Adaptive Flight Control **87**
4.1 Adaptive Flight Control . . . 88
4.1.1 Description of design steps for the quadrotor . . . 92
4.2 Adaptive Backstepping Flight Control implementation . . . 98
4.2.1 Implementation in the yaw angle . . . 99
4.2.2 Trajectory tracking and parameters estimation . . . 99
4.2.3 Results . . . 102
4.2.4 Implementation in the altitude . . . 103
4.2.5 Trajectory tracking and parameters estimation . . . 104
4.2.6 Results . . . 107
4.3 Conclusion . . . 108

General conclusion and perspectives **110**

Nomenclature

x, y, z = longitudinal, lateral and vertical motions, (m)

ϕ, θ, ψ = roll, pitch and yaw angles, (rad)

p, q, r = roll, pitch and yaw rates, (rad / s)

R_E = Assumed inertial terrestrial reference

R_B = reference related to quadrotor

R_{EB} = The homogeneous transformation matrix from R_E to R_B

I_x, I_y, I_z = roll, pitch and yaw inertia moments, (Kg.m^2)

C_L, C_D = lift and drag force coefficients

U_z = lift force, (N)

U_ϕ, U_θ, U_ψ = roll, pitch and yaw moments, (N.m)

m = mass, (Kg)

K_{fdx} = drag force coefficient w.r.t x

K_{fdy} = drag force coefficient w.r.t y

K_{fdz} = drag force coefficient w.r.t z

K_{fax} = aerodynamic friction coefficient w.r.t x

K_{fay} = aerodynamic friction coefficient w.r.t y

K_{faz} = aerodynamic friction coefficient w.r.t z

ω_i = rotor velocity, (rad) / s)

J_r = rotor inertia, (N.m / rad / s^2)

d = distance, (m)

g = gravity acceleration, (m / s^2)

List of Figures

1.1 Examples of the military drones 6
1.2 Phoenix 4 AG 10L [11] 7
1.3 Examples of the cinematograph drones 7
1.4 Examples of the Transport and logistics drones 8
1.5 The configuration and the frame of quadrotor 10
1.6 Brushless DC motor (1400kV) [10] 10
1.7 Propellers used for quadrotor [13] 11
1.8 ESC (Electronic speed controller) [6] 12
1.9 LiPo batterie (20000mAh) [12] 12
1.10 NAZA flight controller for quadrotor [5] 13
1.11 The sensors [1] 14
1.12 A MRAC system for the first-order plant 17
1.13 First ordre system, r = 4 19
1.14 Tracking performance, r = 4 19
1.15 Parameter estimation, r = 4 20
1.16 Tracking performance, r = 4 sin(3t) 20
1.17 Tracking performance, r = 4 sin(3t) 21
1.18 Parameter estimation, r = 4sin(3t) 21
1.19 The control law, r = 4 22
1.20 The control law, r = 4sin(3t) 22

2.1 EULER Angles 27
2.2 Roll and Pitch torques 36
2.3 Yaw torque 37

3.1 Quadrotor control structure flowchart 47
3.2 Closed loop PID control block diagram 52
3.3 Block diagram of the PID command applied to the quadrotor 53
3.4 The block diagram of the quadrotor simulation model 55
3.5 The block diagram of the PID control 55
3.6 Block diagram of dynamic inversion control 56
3.7 Longitudinal motion (x) PID controller 57
3.8 Generation of the desired trajectory by PID control 58
3.9 The attitude of the quadrotor achieving the desired trajectory 59
3.10 3D helical trajectory by the PID controller (where z = t) 60
3.11 The commands of the PID control technique 61
3.12 The rotational speeds of the four rotors 61
3.13 Generation of the desired trajectory by the PID controller 63
3.14 The attitude of the quadrotor achieving the desired trajectory 64
3.15 3D visualization of the trajectory by the PID controller 65
3.16 The commands of the PID controller technique 66
3.17 The rotational speeds of the four rotors 66
3.18 Block diagram of the backstepping controller applied to the quadrotor 72
3.19 The block diagram of the Backstepping controlller 73
3.20 The control input U2 of the Backstepping controller 74
3.21 Generation of the desired trajectory by the Backstepping controller 75
3.22 The attitude of the quadrotor achieving the desired trajectory 76
3.23 3D helical trajectory by the Backstepping controller (where z = t) 77
3.24 The control inputs of the Backstepping controller 78
3.25 The rotational speeds of the four rotors for $z = t$, $\psi = 1$ 78

3.26 Generation of the desired trajectory by the Backstepping controller 80
3.27 The attitude of the quadrotor achieving the desired trajectory 81
3.28 3D visualization of the trajectory by the Backstepping controller 82
3.29 The control inputs of the Backstepping controller technique 83
3.30 The rotational speeds of the four rotors 83
3.31 3D visualization of the trajectory by the Backstepping and the PID controllers . 84
3.32 3D helical trajectory by the Backstepping and PID controllers 84

4.1 Rotor with adaptive control 90
4.2 Rotor speed tracking 91
4.3 Parameters estimation 91
4.4 Block diagram of the Adaptive backstepping flight command applied to the quadrotor 98
4.5 The block diagram of Adaptive Backstepping flight control applied to the yaw control 100
4.6 The estimator equation 100
4.7 The U_4 command equation 101
4.8 The heading angle ψ_d and ψ outputs 102
4.9 The ϕ, θ angles and altitude z outputs 102
4.10 The λ_8, λ_9 and h_3 estimated outputs 103
4.11 The block diagram of Adaptive Backstepping flight control applied to altitude control 104
4.12 The estimator equation 105
4.13 The U_1 command equation 105
4.14 The altitude z_d and z outputs 106
4.15 The ϕ, θ and ψ angles outputs 107
4.16 The λ_{11} and the mass m estimated outputs 107

List of Tables

1.1 Unmanned aerial Vehicles classification based on weight and range 5

3.1 The physical parameters of the quadrotor 54

3.2 The rotor parameters 56

3.3 The Parameters of the PID controller 57

General introduction

In recent years, there has been significant development in Unmanned Aerial Vehicles (UAVs). Various types of drones with different structures, favorable performance, and wide applications have emerged. Among them, multi-rotors have gained attention due to their advantages in hovering ability, maneuverability, simple structure, and cost-effectiveness [37]-[27]. The quadcopter, in particular, is widely used in aerial photography, inspection, surveillance, cargo transportation, precision agriculture, and more [19]. Furthermore, the design of multi-rotors with more rotors, such as hexacopters and octocopters, is rapidly evolving, providing increased lift and flight duration [28]-[15]. However, additional rotors result in a larger size, limiting their use in certain areas. Despite their advantages, designing an effective controller to ensure robust flight capabilities for multi-rotor drones remains challenging [35].

The following main reasons contribute to the difficulty [34]-[39]:

1-Multi-rotor drones are under-actuated systems with six degrees of freedom but only four control inputs.

2-The system is complex, nonlinear, and strongly coupled, with the dynamics of translation and rotation interacting with each other.

3-Uncertainties, unmodeled dynamics, and external disturbances always affect system stability and control performance.

These challenges have captured the attention of many researchers, leading to the introduction of various control methods aimed at addressing these issues.

The objectives of this work are:

1-Dynamic modeling of a quadrotor with vertical takeoff and landing.

2-Designing a controller for the stabilization and control of this type of multi-rotor.

The first chapter introduces the classification of UAVs and the composition of the quadrotor. It also provides a review of recent literature on the different approaches used for quadrotor control, with a focus on adaptive control.

In the second chapter, the modeling of a quadrotor drone is detailed to design the control laws and validate them. The Euler-Newton formalism is applied to derive the mathematical model. The low-level modeling of the rotors is also presented.

The third chapter focuses on the design of control laws for stabilizing and tracking the trajectory with the PID and the Backstepping controllers. In this study, simulations were conducted using MATLAB Simulink to express the results. A hierarchical control approach was proposed, employing a cascade structure that decomposes the system dynamics into two subsystems: translation and rotation. Dynamic inversion was employed to control the rotor speeds.

In the fourth chapter, we consider the scenario where the quadrotor has non-symmetric geometry and the mass varies due to energy consumption from the battery, leading to unknown system parameters. To address this, we employed the backstepping control approach with an adaptive method. In MATLAB/Simulink, we estimated the new parameters of the system to account for these variations.

Finally, the work concludes with an assessment and perspectives of our research.

Chapter 1

State of the Art

1.1 Introduction

Pilot autonomous aerial vehicles or Unmanned Aerial Vehicles (UAVs) have become a real center of interest. In recent years, their use has increased considerably. Today, they are used for civil and military applications, for multiple tasks such as aerial photography, videography, topography, cartography, payload transport, agriculture, rescue, surveillance, etc.

In this chapter, we present the classification of drones, the different components of the quadrotor, the control techniques applied to the quadcopter and the adaptive control.

1.2 Classification of drones

The drones are classified on several criteria :

1.2.1 According to the aerodynamic shape

Nowadays, aeronautics engineers design and manufacture different aerodynamic shapes of drones. They can be classified into three types, fixed wing, flapping wing, and rotary wing (multi-rotor). Multirotors are classified according to the number of rotors and their positioning. The quadrotor studied in this work is a multi-rotor of four rotors.

1.2.2 Depending on the landing method

There are two types, HTOL and VTOL :

HTOL: is an extension of fixed-wing drones, they have a high cruising speed and a classic horizontal landing.

VTOL: they often use a vertical propulsion system. This type of drone can take off and land vertically and does not need a runway to take off.

1.2.3 Based on Weight and Range

Some researchers and organizations have classified drones based on weight and range. Table 1.1 presents the list of unmanned aerial vehicles based on weight and range [32].

Table 1.1: Unmanned aerial Vehicles classification based on weight and range

Type	Maximum Weight	Maximum Range	Category
Nano	200 g	5 km	Fixed wing,multirotor
Micro	2 kg	25 km	Fixed wing,multirotor
Mini	20 kg	40 km	Fixed wing,multirotor
Light	50 kg	70 km	Fixed wing,multirotor
Small	150 kg	150 km	Fixed wing
Tactical	600 kg	150 km	Fixed wing
MALE	1000 kg	200 km	Fixed wing
HALE	1000 kg	250 km	Fixed wing
Heavy	2000 kg	1000 km	Fixed wing
Super Heavy	2500 kg	1500 km	Fixed wing

1.3 Drone application area

Drones, also known as unmanned aerial vehicles (UAVs), have emerged as versatile tools with diverse applications across various industries. These remotely piloted or autonomous aircraft have revolutionized numerous sectors by providing innovative solutions and enhancing efficiency in a wide range of tasks. Let's explore some of the key application areas of drones.

Military domain

Military drones are employed in contemporary warfare and serve to eliminate strategic targets without endangering the pilots. A use that is still disputed in the army because it puts a cold distance between the operator and his target, between the control screen and the eliminated target. The United States is the largest user of combat drones, which entered their armies in the 1990s. It was for example in 1994 that the Predators made their first test flights and were launched the following year in the Balkan. These devices which have a range of 1250 km can

be launched in reconnaissance without weapons or be equipped with weapons. In general, the drone has a triple use in contemporary armies: surveillance (inside and outside), reconnaissance of the terrain for ground, and air [26].

(a) Predator Drone [20]

(b) Black Hornet mini drone [7]

Figure 1.1: Examples of the military drones

Civil domain

- **Agriculture:** agriculture is experiencing a new mutation that is characterized by IT, communications, robotics, or even data collection. agriculture uses drones to map, analyze and process crops. This is not just the application of new technologies, but an information revolution, which can lead to more accurate and efficient farm management systems [21].

- **Cinematograph:** In recent years, drones for filming have come a long way with improved mechanisms and high-resolution cameras. Drones have become an integral part of filmmaking. In the past, filmmakers were forced to use helicopters and cranes to get that aerial shot that adds more excitement to a movie. The video quality of these small cameras, attached to the drones themselves, has improved dramatically over the years [23].

- **Transport and logistics:** Drones have appeared recently in the field of logistics, and many companies have been interested in them for several years. Right now they are in the

spotlight because we are going through a period of rapid market growth for this technology. Logistics companies are developing them with the aim of making them operational in the next few years. For example, Amazon first became interested in drones when it launched its Prime Air program in 2013. The project's goal is to deliver packages within 30 minutes of ordering to an address 10 miles away. Over the past few years, Amazon has conducted test flights in multiple locations around the world to operate drones as soon as the law permits. Over the next few years, drones will be in everyday use just like trucks are today [30].

Figure 1.2: Phoenix 4 AG 10L [11]

(a) DJI Mini 2 [4]

(b) K60 Pro [9]

Figure 1.3: Examples of the cinematograph drones

(a) DHL delivery [2] (b) Mini drone for delivery [3]

Figure 1.4: Examples of the Transport and logistics drones

1.4 Advantages and disadvantages of drones

1.4.1 Advantages

- Maintain an environment.
- Cost saving technology.
- Aerial imagery quality.
- Easy to control.
- Security.
- Minimize obvious hazards and health risks.
- Areas of action that change depending on the scenario.
- Remote access to dangerous or impassable areas.

1.4.2 Disadvantages

- The weather conditions, and more specifically the wind, must be within the limits of the drone to guarantee a good result for your sequences.

- drones really don't like rain. On the other hand, they tolerate snow, cold and very high altitudes (under certain conditions).
- Drones are easy to hack.
- Slow data transfer.
- Night flights are limited depending on conditions. (Night flights are defined as taking place 30 minutes before or after sunrise or sunset.
- overflights of major cities are subject to local restrictions and the necessary prior authorizations. (especially near airports).

1.5 Description and composition of the quadrotor

The quadrotor is a multi-rotor of four rotors attached to a rigid frame. it has several characteristics (mechanically simple, vertical take-off and landing, hovering, more payload, longer flight time, high maneuverability, faster). it consists of:

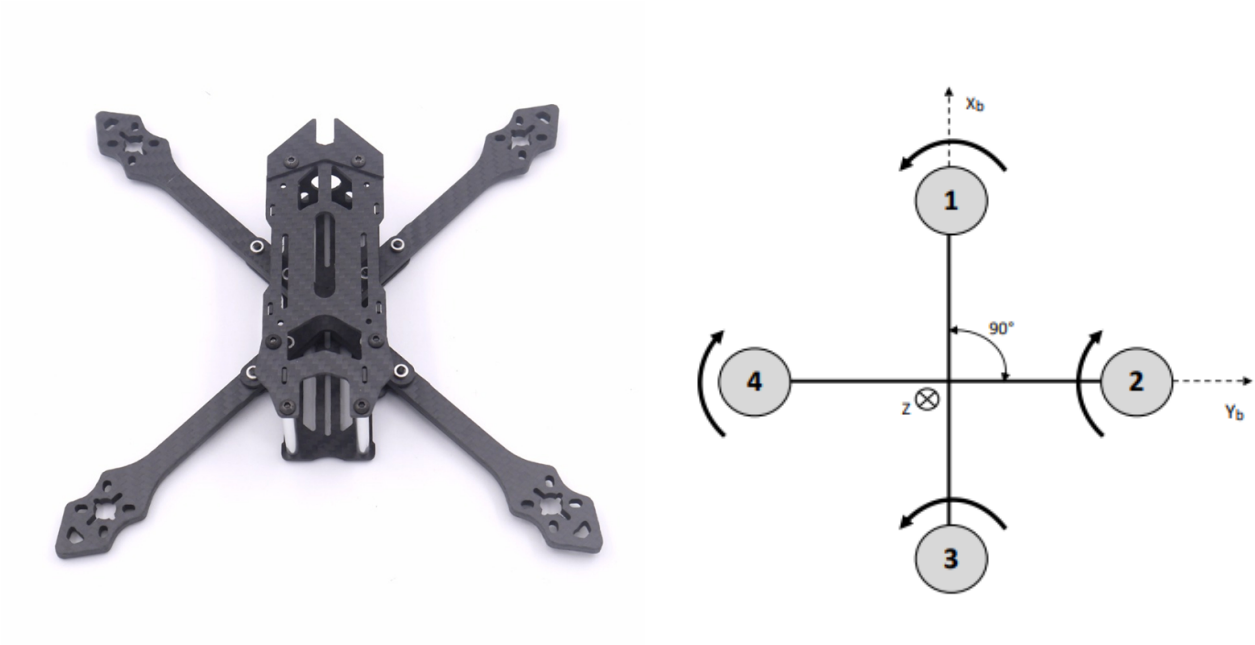
1.5.1 Frame

The frame is designed to support the weight of other components. it gives the aerodynamic shape of the drone to ensure its stability. it must be strong and light. the materials used for its manufacture are generally carbon fibers and aluminum.

1.5.2 Motors

A brushless DC motor (BLDC) is a synchronous electric motor that is powered by direct current and has an electronically controlled commutating system instead of a brush-based mechanical commutating system. In this type of motors, the current and the torque, the voltage, and the speed of rotation are linearly related. Each motor drives a propeller. brushless motors are listed according to their KV(Figure 1.6), which corresponds to the number of revolutions

carried out per minute and per volt. Without them, there would be no multi-rotor drones. They do not use brushes to distribute energy to the rotor. The magnets spin, and the coils are static, causing less friction and wearing parts except for the shaft. The brushless motors have an efficiency of close to 90% and do not require maintenance. there are two kinds: rotating squirrel cage motors, the most common because they offer good torque and fixed squirrel cage motors that can be embedded in a fuselage [22].



(a) Frame [8] (b) Quadrotor configuration +

Figure 1.5: The configuration and the frame of quadrotor

Figure 1.6: Brushless DC motor (1400kV) [10]

1.5.3 The propellers

The quadrotor operates with two normal-pitch propellers (clockwise, CW) and two reverse-pitch propellers (counterclockwise, CCW). The materials used, in increasing order of price and performance, are nylon (plastic), wood (typically beech), and carbon fiber. Wooden propellers have the advantage of being less dangerous in case of impact. Metal propellers, reserved for general aviation, are used for large drones. The propeller can be of fixed size or foldable, which makes it less fragile and allows for space-saving during transport. They are classified based on the product size × step, where the size is the length in inches of the propeller and the pitch is the horizontal distance traveled in inches by the propeller in each revolution.

Figure 1.7: Propellers used for quadrotor [13]

1.5.4 Variable speed drives

The speed variators on ESC (Electronic speed controller) are electronic circuits, they transform the direct current of the battery to an alternating current in precise impulses to operate the brushless motors. They are characterized by a value in amperes (ex: 30A) which indicates the maximum intensity that they can withstand during the flight. It is modeled as a three-phase DC-AC inverter which transforms the DC voltage (PWM signal) in its input to a three-phase voltage in its output.

Figure 1.8: ESC (Electronic speed controller) [6]

1.5.5 Battery

The batteries are the source of energy for the various components of the quadrotor. If circuits of different voltages coexist on board, it is always lighter and more practical to power them by the flight battery alone (possibly adding transformers), rather than by other batteries. the battery must have an optimal weight/power ratio. The "Lithium Polymer" category meets this requirement. A LiPo cell provides a voltage of 3.7V. On a drone, batteries with 3 or 4 cells are generally used. The capacity, voltage and discharge coefficient are indicated in their box.

Figure 1.9: LiPo batterie (20000mAh) [12]

1.5.6 Flight controller

It is an electronic card, it receives the information measured by the sensors and transmits electrical impulses to the speed variators of the motors. It contains a microprocessor, memory,

interfaces I/O and sometimes integrated sensors. It is programmed through algorithms to manage the flight in an autonomous or semi-autonomous way.

Figure 1.10: NAZA flight controller for quadrotor [5]

1.5.7 The sensors

Different sensors are used to measure acceleration, speed, altitude, attitude and translate them into electrical current sent to the flight controller.

Example

— **The accelerometers:** three sensors (one per axis) to measure the accelerations of the translation movement.

— **Gyroscopes:** three gyros (one per axis) to measure the angular rotation speed on each axis.

— **The compass:** to measure the intensity of the earth's magnetic field and determine north.

— **The GPS module:** it allows geolocation of the drone in longitude, attitude and altitude.

(a) Gyroscope (b) GPS module (c) Accelerometer

Figure 1.11: The sensors [1]

1.6 Quadrotor control

The design of a flight controller consists in synthesizing control laws which calculate the inputs of the quadrotor actuators to produce the forces and torques which act and make it possible to obtain a desired behavior [16].

The flight control approaches proposed in the literature for quadrotor control can be classified into three categories: linear controllers, non-linear controllers and intelligent controllers.

1.6.1 Linear controllers

In this type, the command synthesis is based on a linear approximation of the quadrotor's dynamic model.

For quadrotor control, the PID controller has been implemented either alone or as part of a hierarchical structure of a control algorithm. The quadrotor dynamics are decomposed into two subsystems: rotational dynamics and translational dynamics.

The control is then designed in two loops: an inner loop for attitude control and an outer loop for position control.

The cascade control strategy using PID controllers is the most commonly used in practice. It has been applied for quadrotor flight control. A combination of PID controllers has been used for attitude control of a quadrotor [14].

1.6.2 Non-linear controllers

Backstepping control

The backstepping technique is based on a recursive algorithm for controlling linear and nonlinear systems, It allows the synthesis of control laws by defining associated Lyapunov functions. A non-linear controller is derived using backstepping techniques and implemented on a quadrotor. Simulation results have shown good performance of the proposed control strategy [36].

One of the problems with the Backstepping controller is its sensitivity to model uncertainties, as complete information about the model is explicitly required for its implementation [25]. This modification is beneficial for both robustness and performance in the presence of unmodeled disturbances. Even under unknown degradation of the propulsion system during maneuvers, the proposed controller was able to maintain the stability of the quadrotor. However, the controller still relies on accurate knowledge of the model and quadrotor states [29].

Sliding mode control

Sliding mode control is a type of variable structure control that can be applied to both linear and nonlinear systems. The basic principle is to drive the system onto a stable hypersurface called the sliding surface, and then ensure convergence on the sliding surface towards the desired equilibrium point. A synthesis of sliding mode control laws was presented. The objective was to stabilize a quadrotor for trajectory tracking. The proposed control strategy was tested through simulations, and the results obtained were satisfactory [33].

The main drawback of this control method is the peculiar dynamic behavior around the sliding surface, known as the chattering phenomenon due to the discontinuous "sign" function in the control law. To address this issue, a saturation function is often employed to mitigate this inconvenience. However, the stability of the system depends on the choice of the control law gain. If the gain is not chosen correctly, the system may become unstable.

Predictive command

The basic principle of predictive control is to use an explicit model of the system in the control algorithm in order to predict the future behavior at the output. control laws are calculated by solving constrained optimization problems to minimize tracking error over a finite time interval called the prediction horizon.

The design of a model predictive controller for attitude control of a quadrotor is introduced. the indoor and outdoor experiments carried out show promising results [31].

1.6.3 Intelligent controllers based on learning

The controllers of this type are based on fuzzy logic and artificial intelligence. A fuzzy linear controller is proposed to improve the altitude flight performance of a quadrotor. The control strategy was tested by simulation. The quadrotor followed the desired altitude, but the system response time was not satisfactory. A neural network-based real-time controller is proposed for trajectory tracking while carrying loads. The results obtained are validated experimentally by implementing the controller on a DSP board. They are good and robust compared with a PID controller [18].

1.6.4 State of the art of the adaptive flight controller

Adaptive flight control is a type of control technology used in aviation to deal with uncertainties, variances, and changes in aircraft dynamics and ambient variables. It tries to change control rules and settings in real-time to maintain stability and performance in the face of uncertainty. The main concept underlying adaptive flight control is to include a parameter estimating mechanism that continually updates the aircraft's model parameters. This estimating process can be based on a variety of approaches, including adaptive filters, adaptive observers, and adaptive algorithms, which continually analyze the system's reaction and alter the control inputs appropriately.

Adaptive flight control allows the aircraft to retain stability, maneuverability, and track-

ing accuracy even when faced with unknown disturbances, changing flying circumstances, or fluctuations in the aircraft's characteristics by modifying the control rules depending on the estimated parameters.

Both manned and unmanned aircraft systems have routinely used adaptive flight control. It has increased resilience, fault tolerance, and the capacity to manage different flight regimes and operating situations. To maintain safe and dependable flight operations, the design and implementation of adaptive flight control systems must be carefully considered and validated [38].

Illustrative example

In this section, we explain how to simulate a model reference adaptive controller in MATLAB. The motivation for creating this post comes from the fact that MATLAB implementation of model reference adaptive controllers is usually omitted in control theory books. However, implementation is far from trivial and there are a number of pitfalls that can potentially lead us to misleading conclusions.

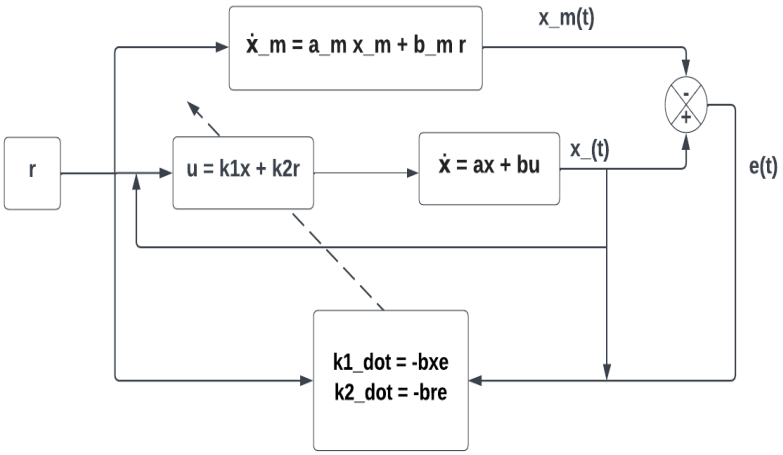


Figure 1.12: A MRAC system for the first-order plant

We focus on a simple first-order system, Consider the control of the unstable plant [24]:

$$\dot{x} = x + 3u \tag{1.1}$$

Where y is an output of the system that we want to control, u is an output of a controller (referred to as the control input), and a and b are constants determining the system dynamics. We assume that constants a and b are unknown during the control design procedure.

The goal of the Model Reference Adaptive Controller (MRAC) is to compute the control input such that the system output is as close as possible to an output of a reference model. The reference model is a model that specifies the desired system behavior. That is, the output of the reference model is the desired system output that we want to achieve in practice. During the design process of MRAC, it is assumed that the reference model is known. Accordingly, for the model (1.1), using the previously designed adaptive controller. The plant parameters $a = -1$,$b = 3$ are assumed to be unknown to the adaptive controller. The reference model is chosen to be:

$$\dot{x}_m = -4x_m + 4r \tag{1.2}$$

Where y_m is the output of the reference model, r is the input of the reference model. We assume that a_m and b_m the parameters of the reference model are known. The signal r is known. Since r is known, we can simulate the reference model (1.2) to generate the output of the reference model y_m. i.e., $a_m = 4$, $b_m = 4$. The adaptation gain γ is chosen to be equal to 2. The initial values of both parameters of the controller are chosen to be zero, indicating no a priori knowledge. The initial conditions of the plant and the model are both zero.

Since the plant parameters were known, the following values of control parameters:

$$a_r^* = \frac{b_m}{b} \qquad a_y^* = \frac{a - a_m}{b} \tag{1.3}$$

Two different reference signals are used in the simulation:

- $r(t) = 4$. It is seen from Figure 1.14 that the tracking error converges to zero but the parameter error does not Figure 1.15.

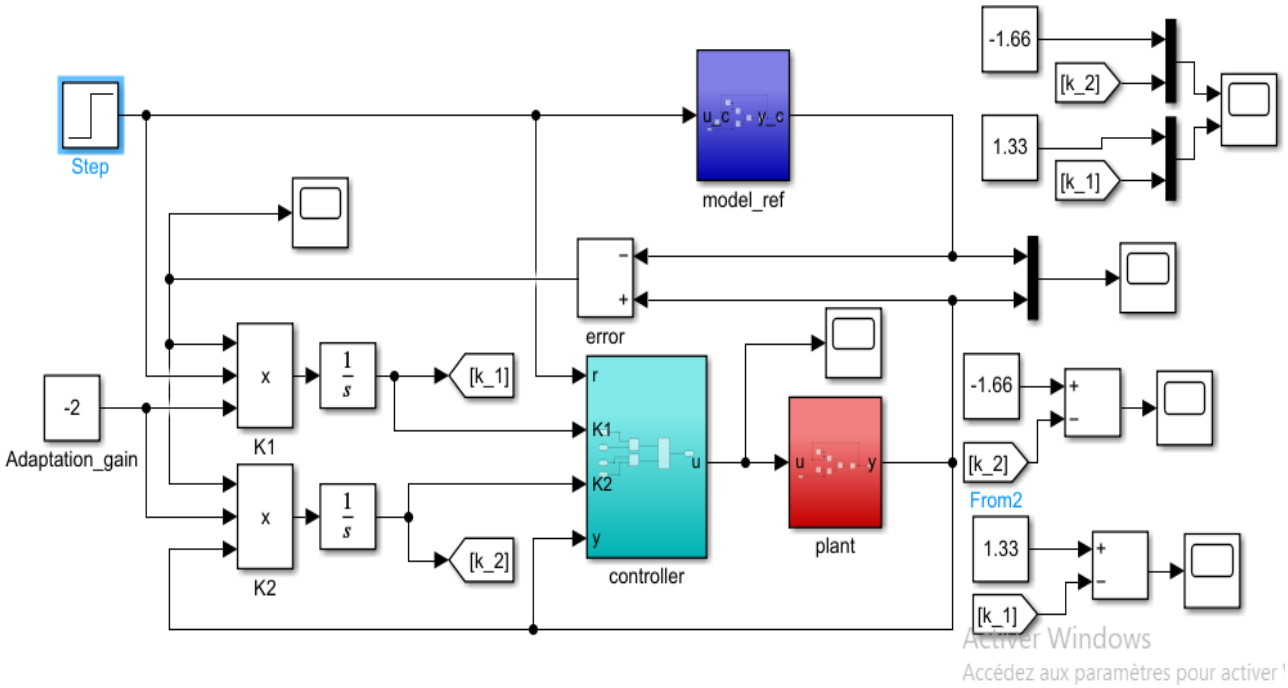


Figure 1.13: First ordre system, r = 4

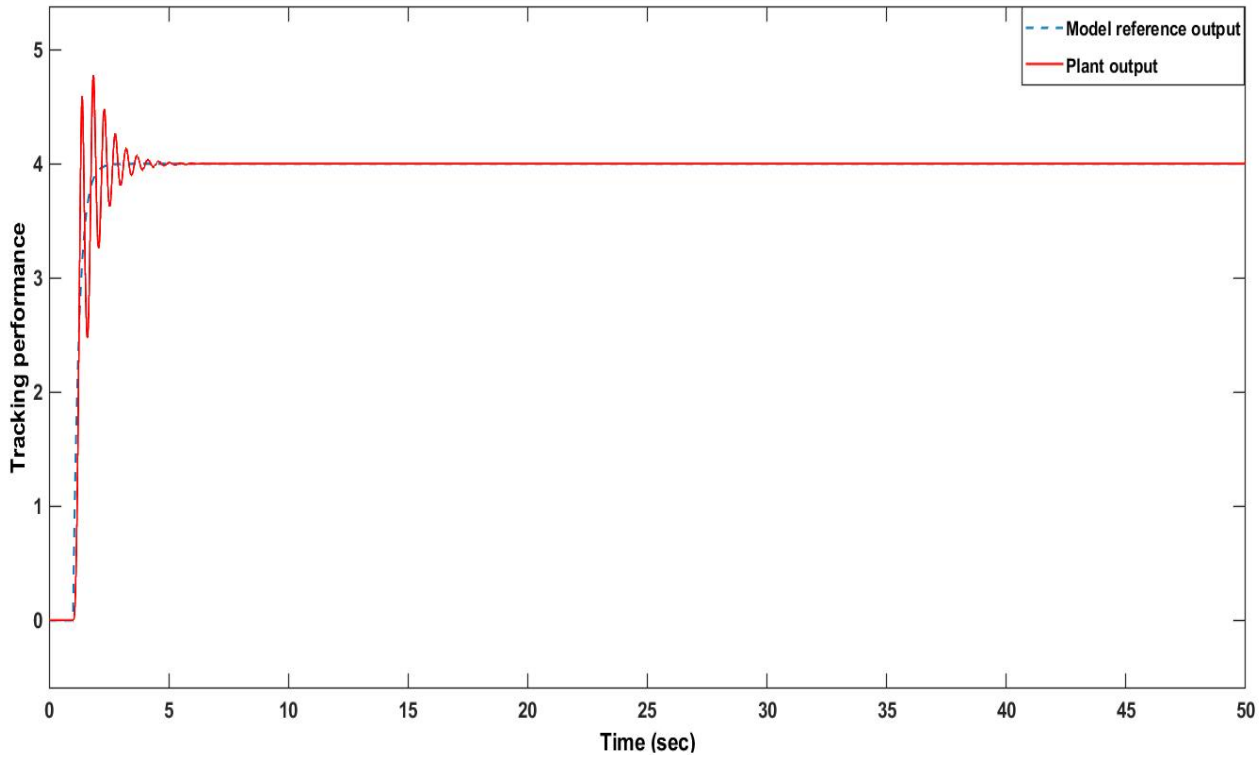


Figure 1.14: Tracking performance, r = 4

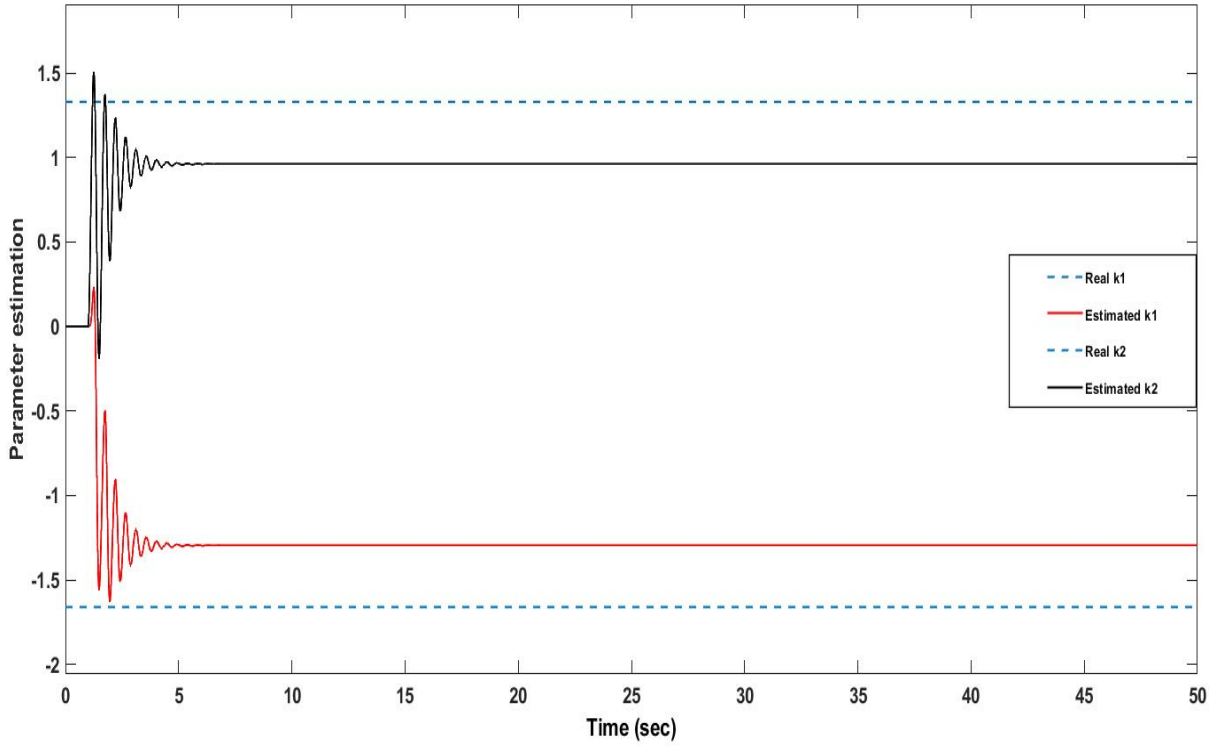


Figure 1.15: Parameter estimation, r = 4

- Now $r(t) = 4\sin(3t)$. It is seen from Figures 1.17 and 1.18 that both the tracking error and parameter error converge to zero.

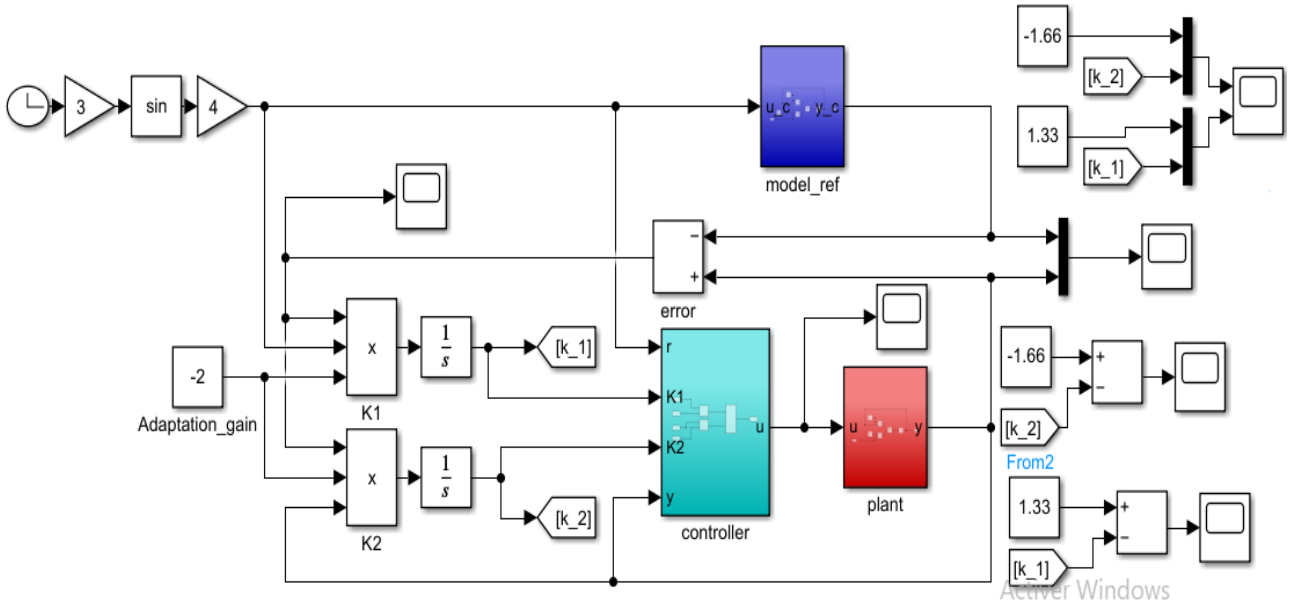


Figure 1.16: Tracking performance, r = 4 sin(3t)

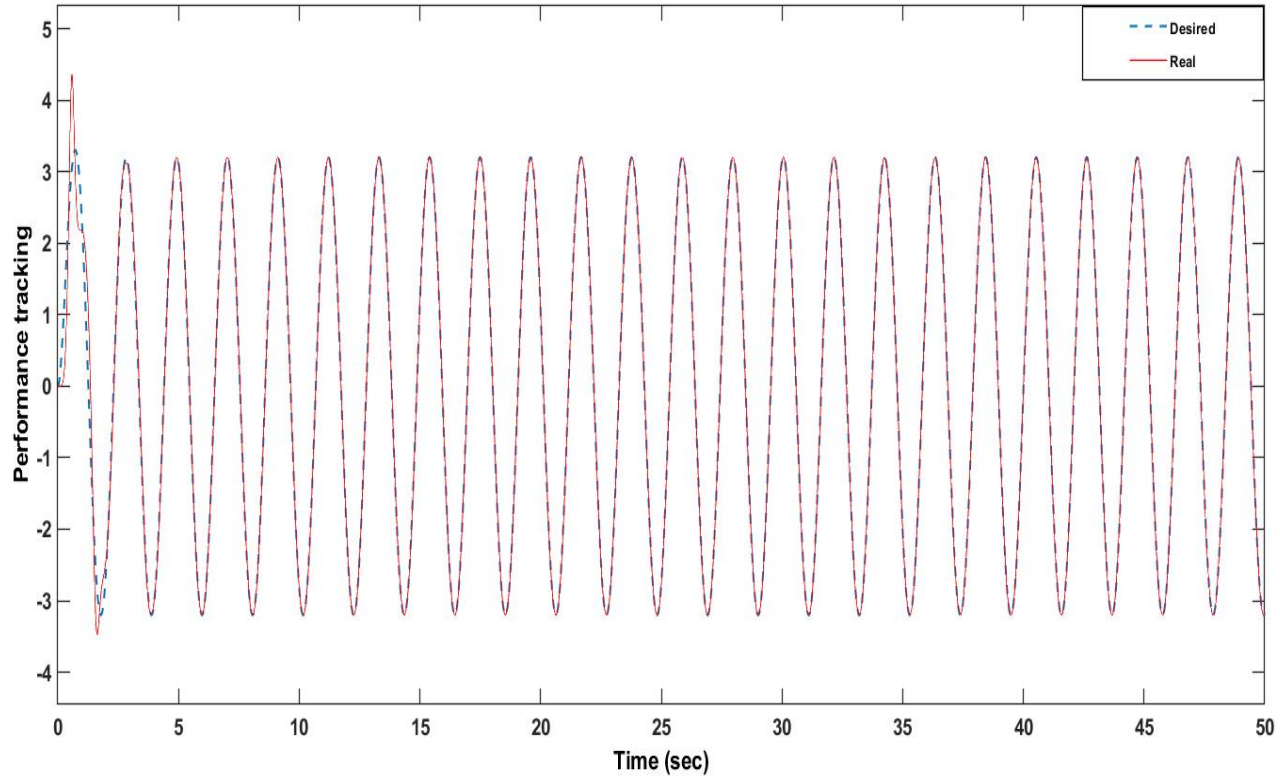


Figure 1.17: Tracking performance, r = 4 sin(3t)

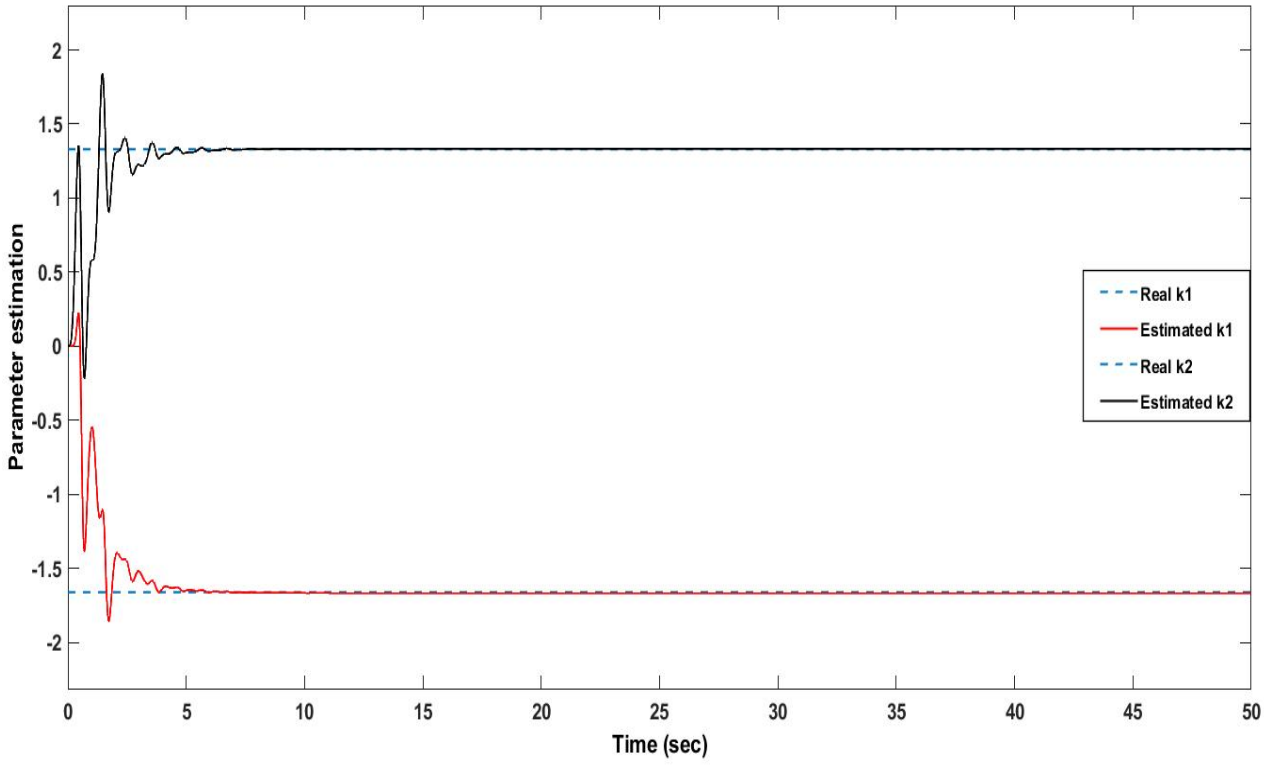


Figure 1.18: Parameter estimation, r = 4sin(3t)

Where the control law is :

$$u(t) = \hat{k}_1 x(t) + \hat{k}_2 r(t) \tag{1.4}$$

And

$$\dot{\hat{k}}_1 = -\gamma r e \tag{1.5a}$$

$$\dot{\hat{k}}_2 = -\gamma y e \tag{1.5b}$$

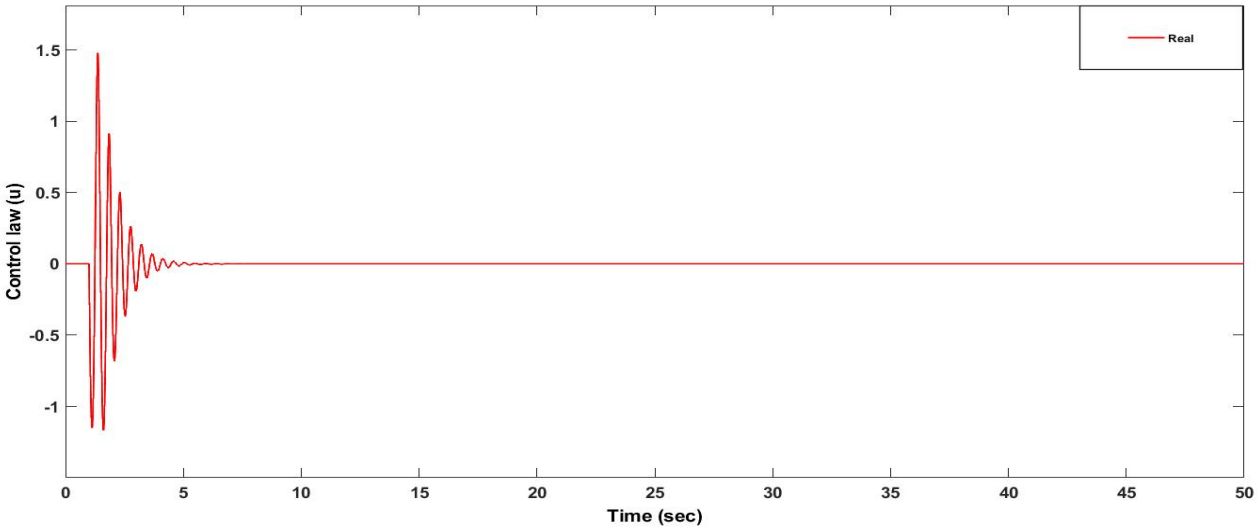


Figure 1.19: The control law, r = 4

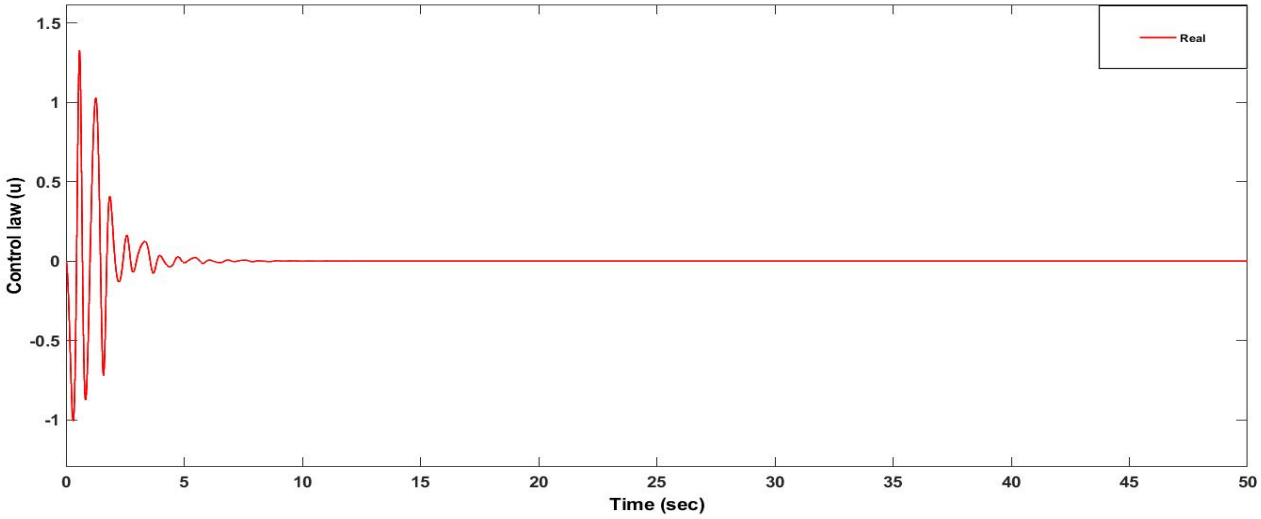


Figure 1.20: The control law, r = 4sin(3t)

Note that, in the above adaptive control design, although the stability and convergence of the adaptive controller are guaranteed for any positive γ, a_m, and b_m, the performance of the adaptive controller will depend critically on γ. If a small gain is chosen, the adaptation will be slow and the transient tracking error will be large. Conversely, the magnitude of the gain and, accordingly, the performance of the adaptive control system, are limited by the excitation of unmodeled dynamics, because too large an adaptation gain will lead to very oscillatory parameters.

Implementing adaptive flight control in a quadrotor can provide several benefits

- **Robustness to uncertainties:** Quadrotors operate in dynamic and uncertain environments where variations in wind, payload, or sensor measurements can affect their performance. Adaptive flight control can help the quadrotor handle these uncertainties by continuously estimating and updating the system parameters, allowing it to adapt and maintain stability and performance.

- **Improved maneuverability:** Quadrotors are highly maneuverable aerial vehicles, and adaptive flight control can enhance their maneuvering capabilities. By continuously adjusting the control inputs based on real-time parameter estimates, adaptive flight control enables the quadrotor to respond quickly and accurately to changing flight conditions, enabling precise trajectory tracking and agile maneuvers.

- **Fault tolerance:** In the event of component failures or degradation, adaptive flight control can help compensate for these faults. By continuously adapting the control laws based on the estimated parameters, the quadrotor can adjust its behavior and maintain stability, even in the presence of faults or failures in its subsystems.

- **Performance optimization:** Adaptive flight control allows the quadrotor to optimize its performance by adjusting the control parameters based on the real-time estimation of the system's characteristics. This optimization can lead to improved efficiency, reduced energy consumption, and enhanced flight characteristics.

- **Adaptability to varying flight regimes:** Quadrotors may operate in different flight regimes, such as hovering, vertical ascent/descent, or aggressive maneuvers. Adaptive flight control can dynamically adjust the control laws and parameters to suit the specific flight regime, ensuring optimal performance and stability across various operating conditions.

- **Enhanced safety:** By continuously monitoring and adapting to changes in the environment and system behavior, adaptive flight control can improve the safety of quadrotor operations. It enables the quadrotor to handle unexpected disturbances or variations, reducing the risk of accidents or instability.

Overall, implementing adaptive flight control in a quadrotor can enhance its robustness, maneuverability, fault tolerance, performance, adaptability, and safety, enabling it to operate effectively in challenging and uncertain environments.

1.7 Conclusion

In this chapter, we give an overview of the different kinds of drones, their different types, their advantages and disadvantages and the fields of their use. Afterward, we presented the state of the art of the different commands used for the stabilization and tracking of the trajectory of the quadrotor. Also, we have talked about adaptive flight control and its benefits on the quadrotor. Moreover, we explained the behavior of adaptive control in an example of the first-order system and we implemented the control law obtained by MRAC's approach to the system using Matlab Simulink to express the advantage of adaptive control and simulate the dynamics of the system. In the next chapter, we'll model the quadrotor.

Chapter 2

Quadrotor Dynamics Modeling

2.1 Introduction

In this chapter, we are interested in modeling the dynamics of a quadrotor with configuration 'X' having two plants of symmetry. The Euler-Newton approach is used, taking into account the physical effects that affect its dynamics, namely aerodynamic effects, gravity, gyroscopic effects, friction and the moment of inertia. The quadrotor has four propellers in a cross configuration where two pairs of propellers turn contrarily. Altering the speeds of the second and fourth propellers produces a bank rotation and lateral displacement. The first and third propeller velocities are reversed, which causes pitch rotation and the associated longitudinal displacement. The motion of the heading is more challenging since it depends on the counter-torque produced by each set of propellers. To limit the complexity of the dynamics modeling, the following assumptions are considered:

- The quadrotor structure is rigid and symmetrical.
- The propellers are rigid.
- The center of the mass and the origin of the body reference frame coincides.
- Thrust and forces are proportional to the square propellers velocity rotation.

Using these assumptions, it is feasible to describe the flight dynamics as that of a rigid body under aerodynamic forces caused by the rotation of the propellers.

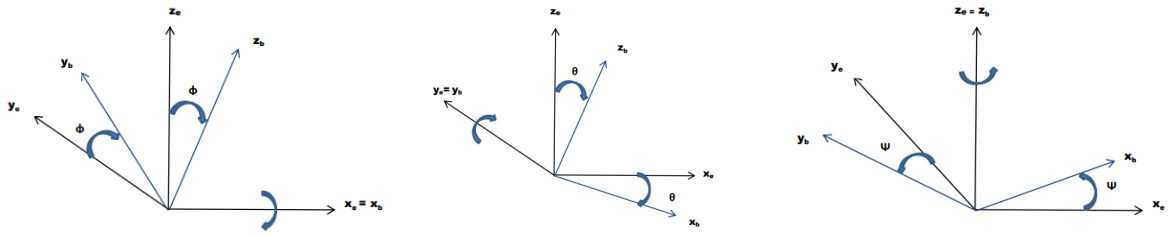
2.2 Reference frames and passage matrices

For the dynamic modeling of the drone, two reference frames are used, one fixed and the other mobile.

Let $E = \{O, x_e, y_e, z_e\}$ be the fixed inertial reference frame linked to the earth where O is the origin of the reference frame, x is oriented towards true north, y towards the east and z downwards.

Let $B = \{O', x_e, y_e, z_e\}$ be the mobile frame associated with the quadrotor whose origin O' is located at the center of gravity of the drone, x_b represents the longitudinal axis of the drone, y_b the lateral axis and z_b perpendicular to the plane (x_b, O, y_b) and facing down.

As the quadcopter is a system with six degrees of freedom, we must define three position variables and three attitude variables. The position of frame B associated with the quadrotor with respect to the inertial frame E expressed in E is denoted $\xi = (x, y, z)^T$ where x, y and z are respectively the longitudinal, lateral and vertical positions. The attitude of frame B associated with the quadrotor with respect to the inertial frame E expressed in E is given by the EULER angles and denoted $\eta = (\phi, \theta, \psi)^T$.For simplicity, the notations sin(.), tan(.) and cos(.) are abbreviated respectively by *"s","t"* and *"c"*.



(a) Roll (b) Pitch (c) Yaw

Figure 2.1: EULER Angles

- **Roll motion:** rotation around the x_b axis by the matrix R_ϕ.

$$\begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & s\phi \\ 0 & -s\phi & c\phi \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} \tag{2.1}$$

- **Pitch motion:** rotation around the y_b axis by the matrix R_θ.

$$\begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ \text{-}s\phi & 0 & c\theta \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} \tag{2.2}$$

- **Yaw motion:** rotation around the z_b axis by the matrix R_ψ.

$$\begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = \begin{bmatrix} c\psi & s\psi & 0 \\ \text{-}s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} \tag{2.3}$$

By the multiplication of the three previous matrices, we obtain the passage matrix which makes it possible to pass from the fixed inertial frame of reference E to the mobile frame of reference B where R is the homogenous matrix transformation.

$$\begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = \begin{bmatrix} c\theta c\psi & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ c\theta c\psi & c\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} \tag{2.4}$$

$\mathbf{R} \in \mathbb{R}^{3\times3}$ is a Direction Cosine Matrix and it satisfies the following properties:

$$R \cdot R^T = R^T \cdot R = I_3 \tag{2.5a}$$

$$\det(R) = 1 \tag{2.5b}$$

$$R^{-1} = R^T \tag{2.5c}$$

Where R^T is the transpose matrix of R, R^{-1} is the inverse matrix of R and I is the identity matrix.

2.3 Kinematic modeling

Kinematic modeling is used to establish the relationship between the translational and rotational speeds of the drone expressed in the fixed inertial frame of reference E as a function of the speeds expressed in the mobile frame of reference B.

Let:

- $\dot{\xi} = (\dot{x}, \dot{y}, \dot{z})^T$: the linear speed of the quadrotor expressed in the frame of reference E.
- $V = (u, v, w)^T$: the linear speed of the quadrotor expressed in the frame of reference B.
- $\dot{\eta} = (\dot{\phi}, \dot{\theta}, \dot{\psi})^T$: the angular speed of the quadrotor expressed in the reference frame E.
- $\Omega = (p, q, r)^T$:the angular speed of the quadrotor expressed in the reference frame B.

The relationship that connects the two linear speeds of the quadcopter is as follows:

$$\dot{\xi} = R^T V \tag{2.6}$$

To find the angular velocity state of the drone, we must take into account the different rotations:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = R_\phi R_\theta R_\psi \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + R_\phi R_\theta \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_\phi \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} \tag{2.7}$$

Where:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & s\phi c\theta \\ 0 & -s\phi & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{2.8}$$

This passing matrix, which we refer to as R_A, enables us to go from the $\dot{\eta}$ speed to the Ω speed.

Where:

$$R_A^{-1} = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{bmatrix} \tag{2.9}$$

Where:

$$\dot{\eta} = R_A^{-1}\omega \tag{2.10}$$

So:

$$\dot{\phi} = p + s\phi t\theta q + c\phi t\theta r \tag{2.11a}$$

$$\dot{\theta} = c\phi q - s\phi r \tag{2.11b}$$

$$\dot{\psi} = \frac{s\phi}{c\theta}q + \frac{c\phi}{c\theta}r \tag{2.11c}$$

2.4 Dynamic modeling

Equations relating to forces, moments, and quadrotor accelerations may be established by dynamic modeling.

The following notation is used to distinguish between physical values represented in different reference frames :

— X^E: for the graders expressed in the fixed inertial frame of reference.

— X^B: for the graders expressed in the mobile inertial frame of reference.

2.4.1 Translation dynamics

Applying the fundamental principle of dynamics (Newton's second law):

$$F^E = m\ddot{\xi} = ma^E \tag{2.12}$$

With:

— F^E: is the sum of the external forces acting on the quadrotor and expressed in the fixed inertial reference frame E.

— m: is the total mass of the quadrotor.

— a^E: is the quadrotor acceleration expressed in the fixed inertial reference frame.

To express this equation in the mobile reference frame B, we use the transposed passage matrix R^T:

$$R^T F^B = mR^T a^B = mR^T \left(\frac{dV}{dt}\right)_E \tag{2.13}$$

so:

$$F^B = m\left(\left(\frac{dV}{dt}\right)_B + \Omega \times V\right) \tag{2.14}$$

The equation of the translational dynamics expressed in the moving frame B is given by:

$$F^B = m\dot{V} + \Omega \times V \tag{2.15}$$

2.4.2 Rotational dynamics

Applying Newton's second law:

$$M^E = J\dot{\Omega} \tag{2.16}$$

With:

— M^E: is the sum of the external torques applied to the quadrotor.

— J: is the inertia matrix of the quadrotor expressed in the moving frame B and defined as follows:

$$J = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \tag{2.17}$$

To express this equation in the moving frame B, we use the inverse passage matrix R_A^{-1}:

$$R_A^{-1} M_B = R_A^{-1} \left(\frac{dJ\Omega}{dt} \right)_E \tag{2.18}$$

From where:

$$M^B = \left(\frac{dJ\Omega}{dt} \right)_E + \Omega \times J\Omega \tag{2.19}$$

The equation of the rotational dynamics expressed in the mobile frame B is:

$$M^B = J\dot{\Omega} + \Omega \times J\Omega \tag{2.20}$$

2.5 Forces and moments acting on the quadrotor

After the kinematic and dynamic modeling, the forces and moments applied on the quadrotor are defined. the forces acting on the drone are gravity, drag, air friction, thrust, lift, the torques produced by the propellers and the gyroscopic effect caused by the rotation of the propellers.

2.5.1 The forces

- **The gravity**

The force of gravity (the weight) is applied to the quadrotor's center of gravity, and deflected towards the center of the earth along the z_e axis.

$$F_g = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \tag{2.21}$$

Where g is gravity accelerator.

- **Lift force**

Each rotor's thrust force is represented by F_i, where i = 1,.., 4. It is pointed in the opposite direction of the axis z_e and perpendicular to the plane of rotation of the propellers. The sum of the thrust forces produced by the four rotors results in the overall lift force:

$$F_L = F_1 + F_2 + F_3 + F_4 \tag{2.22}$$

The expression of the thrust force F_i is given by:

$$F_i = K_L \omega_i^2 \tag{2.23a}$$

$$K_L = C_T \rho r_p^4 \pi \tag{2.23b}$$

With:

— K_L: is is the lift coefficient.

— C_T: is the thrust coefficient of the propeller.

— ρ: is the air density.

— r_p: is the radius of the propeller.

— ω_i: is the angular velocity of rotor i.

F_f is the resultant of the forces generated by the four rotors:

$$F_f = \begin{pmatrix} c\phi c\psi s\theta + s\phi s\psi \\ c\phi s\theta s\psi - s\phi c\psi \\ c\phi c\theta \end{pmatrix} \sum_{\mathrm{i}=1}^{4} F_i \tag{2.24}$$

- **Drag force**

The drag force produced by the resistance of the air on the propeller is parallel to the plane of rotation of the propeller and oriented in the opposite direction of rotation. The drag moment τ_i corresponding to the drag force:

$$\tau_i = d\omega_i^2 \tag{2.25a}$$

$$d = C_D \rho r_p^5 r_p^4 \pi \tag{2.25b}$$

Where:

— d: is the coefficient of proportionality between the square of the rotational speed of the rotor and the drag moment.

— C_D: is the drag coefficient of the propeller.

- **the frictional force of the air**

The quadrotor moves in the air, the latter causes friction that is modeled by a simple and effective model as follows:

$$F_d = \begin{pmatrix} -K_{fdx} & 0 & 0 \\ 0 & K_{fdy} & 0 \\ 0 & 0 & -K_{fdz} \end{pmatrix} \dot{\xi} \tag{2.26}$$

F_d denotes the resultant of the drag forces along (X, Y, Z) axis.

2.5.2 The torques

The rotors are placed at a distance d from the center of gravity of the quadrotor where $2d$ is the wingspan of the drone. This causes roll, pitch and yaw torques that are defined as follows:

— **Roll torque:**

The roll torque noted M_P allows the quadrotor to rotate around the x_b axis, its expression is given by:

$$M_P = d\,(F_3 - F_1) \tag{2.27}$$

— **Pitch torque**

The roll torque noted M_Q allows the quadrotor to rotate around the y_b axis, its expression is given by:

$$M_Q = d\,(F_4 - F_2) \tag{2.28}$$

— **Yaw torque:**

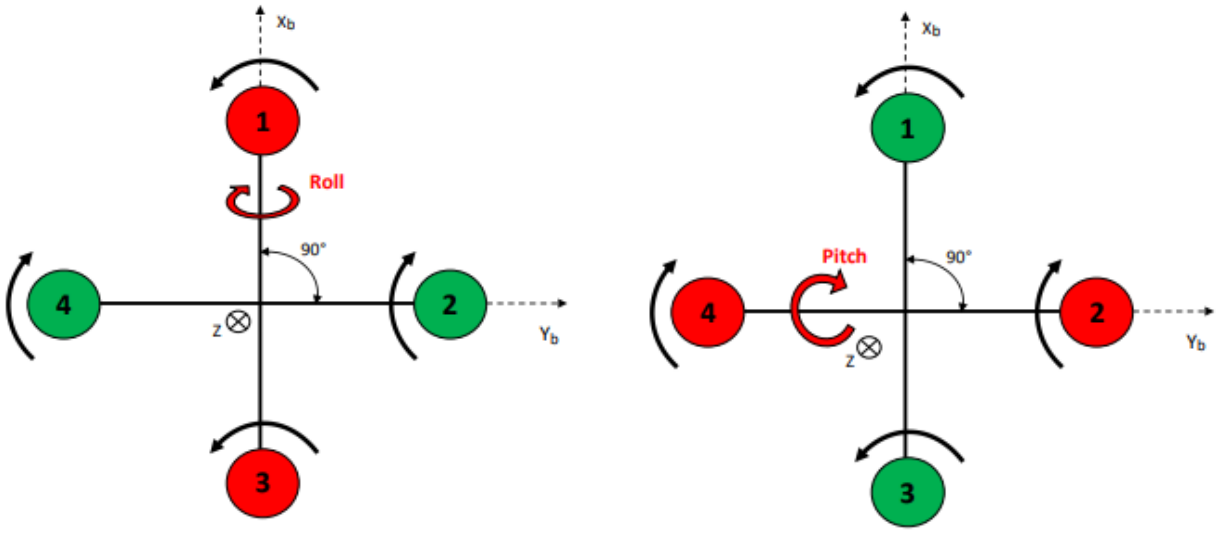
The yaw movement is produced by causing a difference in the rotational speeds of the four rotors.

The yaw torque noted M_R allows the quadrotor to rotate around the z_b axis, its expression is given by:

$$M_R = C_D \left(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2\right) \tag{2.29}$$

Γ_f is the moment developed by the quadrotor according to the body fixed frame such as:

$$\Gamma_f = \begin{pmatrix} d\left(F_3 - F_1\right) \\ d\left(F_4 - F_2\right) \\ C_D \left(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2\right) \end{pmatrix} \tag{2.30}$$



(a) Roll torque (b) Pitch torque

Figure 2.2: Roll and Pitch torques

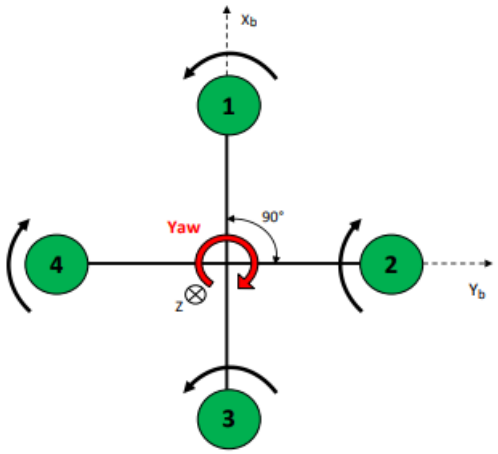


Figure 2.3: Yaw torque

— **Aerodynamic frictions:**

$$\Gamma_a = -A_r \|\Omega\|^2 \tag{2.31}$$

$$\Gamma_a = \begin{pmatrix} -K_{fax} & 0 & 0 \\ 0 & -K_{fay} & 0 \\ 0 & 0 & -K_{faz} \end{pmatrix} \|\Omega\|^2 \tag{2.32}$$

– Γ_a: is the Aerodynamic frictions.

– A_r: is the friction coefficient matrix.

2.5.3 Gyroscopic effect

Each rotor exerts a gyroscopic torque on the quadrotor. We denote it by Γ_g with $i = 1, 2, 3, 4$, it is defined as follows:

$$\Gamma_g = \sum_{i=1}^{4} \Omega \times J_r \begin{pmatrix} 0 \\ 0 \\ (-1)^{i+1}\omega_i \end{pmatrix} \tag{2.33}$$

Where J_r represents the rotor inertia.

$$\Gamma_g = \begin{pmatrix} J_r\dot{\theta}\bar{\Omega} \\ -J_r\dot{\phi}\bar{\Omega} \\ 0 \end{pmatrix} \tag{2.34}$$

$$\bar{\Omega} = \omega_1 - \omega_2 + \omega_3 - \omega_4 \tag{2.35}$$

2.6 Rotor dynamics

The rotor is a group of D.C. motors that a reducer uses to turn a propeller. The following equations provide information on how the D.C. motor works:

$$V = RI + L\frac{dI}{dt} + K_e\omega \tag{2.36a}$$

$$K_m I = J_\Gamma \frac{d\omega}{dt} + K_\Gamma \omega^2 + C_s \tag{2.36b}$$

Where:

— V: is the motor input tension.

— ω: is the rotational speed of the rotor.

— K_e and K_m: represent the electrical and mechanical torque constant, respectively.

— K_Γ: is the load constant torque.

— R and J_Γ: are the motor internal resistance and the rotor inertia, respectively.

— C_s: denotes the solid friction.

From equation 2.36, and as the inductance of the motors is small, it can therefore be neglected. So, we can write the model of the rotor in the following form:

$$\dot{\omega}_i = bV + \beta_0 + \beta_1 \omega_i + \beta_2 \omega_i^2 \tag{2.37}$$

Where: $b = \frac{K_m}{RJ_\Gamma}$, $\beta_0 = \frac{-C_s}{J_\Gamma}$, $\beta_1 = \frac{-K_e K_m}{RJ_\Gamma}$, $\beta_2 = \frac{-K_\Gamma}{J_\Gamma}$

2.7 Quadrotor model

2.7.1 Modeling with the Newton-Euler formalism

Using the formalism of the Newton-Euler, the dynamics equations are written in the following way:

$$m\ddot{\xi} = F_f + F_d + F_g \tag{2.38a}$$

$$J\dot{\Omega} = -\Omega \wedge J\Omega + \Gamma_f - \Gamma_a - \Gamma_g \tag{2.38b}$$

Where ξ is the position of the quadrotor center of mass concerning the inertial frame E,m is the mass of the structure and $J \in \mathbb{R}^{3\times3}$ is a symmetric positive definite constant inertia matrix of the quadrotor concerning the body fixed frame.

Consequently, the complete dynamics are as follows:

- **Translation motion :**

$$\ddot{x} = \frac{1}{m}\left[-K_{fdx}\dot{x} + (c\phi s\theta c\psi + s\phi s\psi)U_1\right] \tag{2.39a}$$

$$\ddot{y} = \frac{1}{m}\left[-K_{fdy}\dot{y} + (c\phi s\theta s\psi - s\phi c\psi)U_1\right] \tag{2.39b}$$

$$\ddot{z} = \frac{1}{m}\left[-K_{fdz}\dot{z} - mg + c\phi c\theta U_1\right] \tag{2.39c}$$

- **Rotational motion:**

$$\dot{p} = \frac{1}{I_x}\left[(I_y - I_z)\, rq - K_{fax}p^2 - J_r\bar{\Omega}q + U_2\right] \tag{2.40a}$$

$$\dot{q} = \frac{1}{I_y}\left[(I_z - I_x)\, rp - K_{fay}q^2 + J_r\bar{\Omega}p + U_3\right] \tag{2.40b}$$

$$\dot{r} = \frac{1}{I_z}\left[(I_x - I_y)\, pq - K_{faz}r^2 + U_4\right] \tag{2.40c}$$

Rotational kinematics is concerned with the quadrotor's attitude movements. The variables $(\dot{\phi}, \dot{\theta}, \dot{\psi})$ describe the rotational velocity in the Earth-fixed reference frame. The rotational velocity is, however, represented by the rates of roll, pitch, and yaw (p, q, r) respectively, in the body reference frame. Equation (2.11) expresses the relationship between these two sets of variables.

where U_1, U_2, U_3 and U_4 denote the system control inputs that are written according to the angular velocities of the four rotors:

$$\begin{pmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{pmatrix} = \begin{pmatrix} K_p & K_p & K_p & K_p \\ -K_p & 0 & K_p & 0 \\ 0 & -K_p & 0 & K_p \\ C_D & -C_D & C_D & -C_D \end{pmatrix} \begin{pmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{pmatrix} \tag{2.41}$$

The dynamic modeling developed above (2.39) and (2.40) is completed by the following control inputs constraints:

$$0 \leq U_1 \leq 4K_p\omega_{\max}^2 \tag{2.42a}$$

$$-K_p\omega_{\max}^2 \leq U_2 \leq K_p\omega_{\max}^2 \tag{2.42b}$$

$$-K_p\omega_{\max}^2 \leq U_3 \leq K_p\omega_{\max}^2 \tag{2.42c}$$

$$-2C_D\omega_{\max}^2 \leq U_4 \leq 2C_D\omega_{\max}^2 \tag{2.42d}$$

2.7.2 Modeling with the Euler-Lagrange formalism

We present in this section the Euler-Lagrange approach to model the quadrotor. In this work, the aim is to find Lagrangian (L). The kinetic energy (T) is composed of two terms, The first term is the kinetic energy of translation and the second term is the kinetic energy of rotation and the potential energy (U).

The generalized coordinates of the quadrotor are described by:

$$q = \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^T \tag{2.43}$$

where $\xi = [x\ y\ z]^T$ represents the position of the center of mass of the quadrotor concerning a fixed frame.

$\eta = [\phi\ \theta\ \psi]^T$ are the three Euler angles (yaw, pitch, roll) and represent the orientation of the quadrotor. The Lagrangian is:

$$L(q, \dot{q}) = T_{trans} + T_{rot} - U \tag{2.44}$$

The kinetic energy of translation is given by:

$$T_{trans} = \frac{1}{2} m \dot{\xi}^T \dot{\xi} \tag{2.45}$$

The kinetic energy of rotation is given by:

$$T_{rot} = \frac{1}{2} J \dot{\eta}^T \dot{\eta} - \Omega \wedge J\Omega \tag{2.46}$$

Where J represents the inertia matrix. The potential energy is given by:

$$U = mgz \tag{2.47}$$

Where z is the vertical position, m represents the mass of the quadrotor.

The Lagrangian L is:

$$L(q, \dot{q}) = \frac{1}{2} m \dot{\xi}^T \dot{\xi} + \frac{1}{2} J \dot{\eta}^T \dot{\eta} - \Omega \wedge J\Omega - mgz \tag{2.48}$$

The dynamic model of the quadrotor is obtained from the Euler-Lagrange equations with the generalized external force.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} + \frac{\partial D}{\partial \dot{q}} = \begin{bmatrix} F_\xi \\ M_\eta \end{bmatrix} = F_{ext} \tag{2.49}$$

Where $F_\xi \in \mathbb{R}^3$ is the translational force applied to the quadrotor due to control inputs $F_\xi = [F_x \ F_y \ F_z]^T$.

$$F_\xi = \begin{pmatrix} c\phi c\psi s\theta + s\phi s\psi \\ c\phi s\theta s\psi - s\phi c\psi \\ c\phi c\theta \end{pmatrix} \sum_{\mathrm{i}=1}^{4} F_i \tag{2.50a}$$

$$F_i = K_L \omega_i^2 \tag{2.50b}$$

$$F_L = \sum_{\mathrm{i}=1}^{4} F_i \tag{2.50c}$$

And $M_\eta \in \mathbb{R}^3$ is the generalized moments $M_\eta = [M_\phi \ M_\theta \ M_\psi]^T$ and R is the transformation matrix representing the orientation of the quadrotor.

$$M_\eta = \begin{pmatrix} dK_L\left(\omega_3 - \omega_1\right) \\ dK_L\left(\omega_4 - \omega_2\right) \\ C_D\left(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2\right) \end{pmatrix} \tag{2.51}$$

And $D \in \mathbb{R}^2$ is the dissipation energy $D = [D_\xi \ D_\eta]^T$

$$D_\xi = \begin{pmatrix} -\frac{1}{2}K_{fdx} & 0 & 0 \\ 0 & -\frac{1}{2}K_{fdy} & 0 \\ 0 & 0 & -\frac{1}{2}K_{fdz} \end{pmatrix} \dot{\xi}^2 \tag{2.52a}$$

$$D_\eta = \begin{pmatrix} \frac{1}{2}J_r\bar{\omega} & 0 & 0 \\ 0 & -\frac{1}{2}J_r\bar{\omega} & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{\theta}^2 \\ \dot{\phi}^2 \\ \dot{\psi}^2 \end{pmatrix} \tag{2.52b}$$

$$\frac{\partial L}{\partial \dot{q}}|_\xi = \begin{bmatrix} 0 & 0 & -mg \end{bmatrix}^T \tag{2.53a}$$

$$\frac{\partial L}{\partial \dot{q}}|_\eta = \begin{bmatrix} -\dot{\psi}\dot{\theta}(I_{yy} - I_{zz}) + K_{fax}\dot{\phi}^2 & -\dot{\psi}\dot{\phi}(I_{zz} - I_{xx}) + K_{fay}\dot{\theta}^2 & -\dot{\theta}\dot{\psi}(I_{xx} - I_{yy}) + K_{faz}\dot{\psi}^2 \end{bmatrix}^T \tag{2.53b}$$

$$\frac{\partial D}{\partial \dot{q}} = \begin{bmatrix} K_{fdx}\dot{x} & K_{fdy}\dot{y} & K_{fdz}\dot{z} & -J_r\bar{\omega}\dot{\theta} & J_r\bar{\omega}\dot{\phi} & 0 \end{bmatrix}^T \tag{2.53c}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) = \begin{bmatrix} m\ddot{x} & m\ddot{y} & m\ddot{z} & I_{xx}\ddot{\phi} & I_{yy}\ddot{\theta} & I_{zz}\ddot{\psi} \end{bmatrix}^T \tag{2.53d}$$

the external force F_{ext} is:

$$F_{ext} = \begin{bmatrix} F_L\left(c\phi c\psi s\theta + s\phi s\psi\right) \\ F_L\left(c\phi s\theta s\psi - s\phi c\psi\right) \\ F_L\left(c\phi c\theta\right) \\ M_\phi \\ M_\theta \\ M_\psi \end{bmatrix} \tag{2.54}$$

Then, the model of the quadrotor with the Euler-Lagrange formalism is:

$$\begin{bmatrix} m\ddot{x} \\ m\ddot{y} \\ m\ddot{z} \\ I_{xx}\ddot{\phi} \\ I_{yy}\ddot{\theta} \\ I_{zz}\ddot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ mg \\ -\dot{\psi}\dot{\theta}(I_{yy} - I_{zz}) + K_{fax}\dot{\phi}^2 \\ -\dot{\psi}\dot{\phi}(I_{zz} - I_{xx}) + K_{fay}\dot{\theta}^2 \\ -\dot{\theta}\dot{\psi}(I_{xx} - I_{yy}) + K_{faz}\dot{\psi}^2 \end{bmatrix} + \begin{bmatrix} K_{fdx}\dot{x} \\ K_{fdy}\dot{y} \\ K_{fdz}\dot{z} \\ -J_r\bar{\omega}\dot{\theta} \\ J_r\bar{\omega}\dot{\phi} \\ 0 \end{bmatrix} = \begin{bmatrix} F_L\left(c\phi c\psi s\theta + s\phi s\psi\right) \\ F_L\left(c\phi s\theta s\psi - s\phi c\psi\right) \\ F_L\left(c\phi c\theta\right) \\ M_\phi \\ M_\theta \\ M_\psi \end{bmatrix} \tag{2.55}$$

2.8 Conclusion

This chapter presented a mathematical modeling approach for the quadrotor UAV. The system was modeled by subsystems (motion kinematics, motion dynamics, control efficiency and

motor dynamics). We have thus obtained the relationship between the quadrotor's movements (position and orientation) and the voltages supplied to the four motors. We have used two different types of formalism, Euler-Lagrange and Euler-Newton, to establish the mathematical model of the UAV as a rigid body with two orthogonal planes of symmetry. We took into account the force of friction with the air and the gyroscopic effect of the rotors on the quadcopter to make the model as close as possible to reality. The development of the control law is presented in the next chapter, based on this model. And we will validate the synthesized control laws in a Simulink MATLAB R2021A simulation environment.

Chapter 3

Control design techniques

3.1 Introduction

In this chapter, we present the control part of the quadrotor. Before moving on to the control of the real system, we need to run simulations to validate the mathematical model and identify the controller parameters. It would be difficult to control a quadrotor UAV without implementing a control law. This must enable the calculation of the speed setpoint for each of the four motors, to ensure drone stability while following a position and/or attitude (orientation) reference.

The objective of the chapter is to ensure that the quadrotor achieves a predefined trajectory $x_d(t)$, $y_d(t)$, $z_d(t)$, $\psi_d(t)$ while remaining stable along its mission. To achieve this, three controls will be studied and synthesized, namely: a PID as a classical technique, a nonlinear control by Backstepping, and finally an Adaptive control.

3.2 General command structure

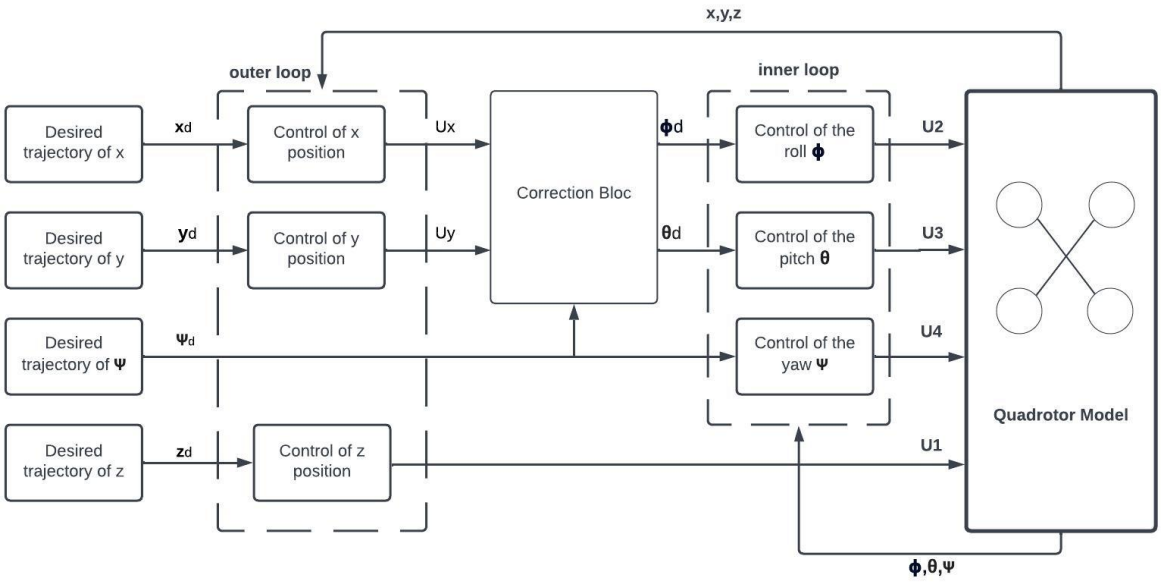


Figure 3.1: Quadrotor control structure flowchart

Quadrotor control is achieved by a cascade structure consisting of two loops. The controller of the internal loop related to attitude dynamics is responsible for tracking the drone's orientation reference, i.e. to follow angles ϕ_d, θ_d and ψ_d, and the external loop linked to po-

sition dynamics, which position dynamics, which generates the reference roll and pitch angles and the total lift force required to position the drone at a given altitude. The derivatives of the quadrotor's position and orientation form its complete dynamics. The overall system can be subdivided into two subsystems, describing the dynamics of rotation and translational dynamics.

In Figure 3.1, we present a simplified synoptic diagram explaining the controller structure of a quadrotor UAV. The translational motion controller (x, y) outputs the desired orientation (ϕ_d, θ_d) of the quadrotor due to the coupling between these variables. The synthesis of a second controller then stabilizes the attitude on the desired heading. For example, to hover, the roll and pitch angles must be maintained at zero. Tilting the machine causes it to move in the (x, y) plane. Hence the importance of the attitude controller.

In the first case, the control of orientation and altitude is the simplest to realize, since it is completely independent of the control of other degrees of freedom. As it is completely independent of the control of other degrees of freedom.

In the second case, all three position coordinates plus yaw are controlled. However, this latter control mode uses both the roll and pitch orientation controllers. In short, the control signals from the three position controllers define a lift force vector in the inertial reference frame. The orientation of this force defines the setpoint sent to the roll and pitch controllers.

3.3 Command synthesis model

Quadrotor control can be achieved by various approaches, including model-based approaches. Model-based control approaches exploit, as their name suggests, the model for control law synthesis. However, using the complete model to satisfy the control objective may be unnecessary, or even costly in terms of computing capacity (computing time and memory space). For this reason, we use simplified models to develop control laws, known as "synthesis models". Simplification is based on assumptions that are more or less respected during system evolution.

In the case of the quadrotor, on the assumption that it performs angular movements of low

amplitude, the angular velocity vector Ω expressed in the body reference frame can be reduced to :

$$\Omega = \begin{pmatrix} p \\ q \\ r \end{pmatrix} \simeq \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \tag{3.1}$$

Let's take as attitude control inputs: U_2, U_3 and U_4 the roll, pitch and yaw torques respectively. The synthesis model for attitude control is given by :

$$\ddot{\phi} = \frac{1}{I_x}\left[\dot{\theta}\dot{\psi}\left(I_y - I_z\right) - K_{fax}\dot{\phi}^2 - J_r\varpi\dot{\theta} + U_2\right] \tag{3.2a}$$

$$\ddot{\theta} = \frac{1}{I_y}\left[\dot{\phi}\dot{\psi}\left(I_z - I_x\right) - K_{fay}\dot{\theta}^2 + J_r\varpi\dot{\phi} + U_3\right] \tag{3.2b}$$

$$\ddot{\psi} = \frac{1}{I_z}\left[\dot{\phi}\dot{\theta}\left(I_x - I_y\right) - K_{faz}\dot{\psi}^2 + U_4\right] \tag{3.2c}$$

As for position control, the new model uses as control inputs U_2, U_3 and U_4 for position control in x, y and z.

The synthesis model for position control is given by:

$$\ddot{x} = \frac{1}{m}\left[-K_{fdx}\dot{x} + u_x U_1\right] \tag{3.3a}$$

$$\ddot{y} = \frac{1}{m}\left[-K_{fdy}\dot{y} + u_y U_1\right] \tag{3.3b}$$

$$\ddot{z} = \frac{1}{m}\left[-K_{fdz}\dot{z} - mg + c\phi c\theta U_1\right] \tag{3.3c}$$

Where:

$$u_x = \left(c\phi s\theta c\psi + s\phi s\psi\right) \tag{3.4a}$$

$$u_y = (c\phi s\theta s\psi - s\phi c\psi) \tag{3.4b}$$

Using the expressions for u_x and u_y given in (3.4), we obtain the desired angles ϕ_d and θ_d as follows:

$$\phi_d = \arcsin\left(u_x s\psi_d - u_y c\psi_d\right) \tag{3.5a}$$

$$\theta_d = \arcsin\left(\frac{u_x c\psi_d + u_y s\psi_d}{c\phi_d}\right) \tag{3.5b}$$

3.4 PID controller

PID control is one of the most basic and frequently implemented types of control in the industry. The advantages of the PID controller are its simple structure and low requirements on the system model (error-based control). The PID controller aims to minimize the current error e(t) = r(t) - y (t), where r(t) is a desired value, called reference or setpoint, and y(t) is a measured value of the process output (figure 3.2). Since the process output is used by the controller to calculate a control signal which is fed back into the process, a closed loop is formed.

The controller output is made up of three terms that give it its name:

3.4.1 Proportionnel

The term P (Proportional) is an amplification of the error where:

$$P = K_p e(t) \tag{3.6}$$

The proportional gain K_p, is a constant. If K_p is high, the control signal will be large and the error will be large, making the controller sensitive and resulting in reduced stability and large overshoot. Since the proportional term is an amplification of the error, it will get smaller

and smaller as the error approaches zero. The steady-state error depends on the size of K_p; a high value of K_p will give a low steady-state error. In theory, an infinite gain would leave a zero steady-state error. In many implementations, the proportional part has the greatest influence on the control signal.

3.4.2 Integrator

The term I (Integrator) depends on the accumulated differences of old errors where:

$$I = K_I \int_0^t e(\tau) d\tau \tag{3.7}$$

The integral gain K_i, is a constant. By integrating the error signal, the output will reach the reference faster, and the steady-state error mentioned above is eliminated. The disadvantage is that the integral part can increase too quickly, causing the control signal to become larger than necessary to keep the error at zero, and since the **I** term can only become smaller if the error is negative, this will lead to an overshoot.

3.4.3 Derivator

The D term (Derivator) is proportional to the derivative of the error where:

$$D = K_D \frac{d}{dt} e(t) \tag{3.8}$$

The gain of the derivative K_D is a constant. By looking at the derivative, it is possible to predict the future to correct for rapid changes and thus prevent overshoot and introduce stability into the closed-loop system. Since the derivative of the error is used, the D term is likely to amplify noise and therefore can be dangerous to use without proper filtering.

The total output of the PID controller is:

$$u(t) = K_p e(t) + K_I \int_0^t e(\tau)d\tau + K_D \frac{d}{dt} e(t) \tag{3.9}$$

It is not necessary to use the three terms of the PID regulator, any combination of the terms P, I and D is possible.

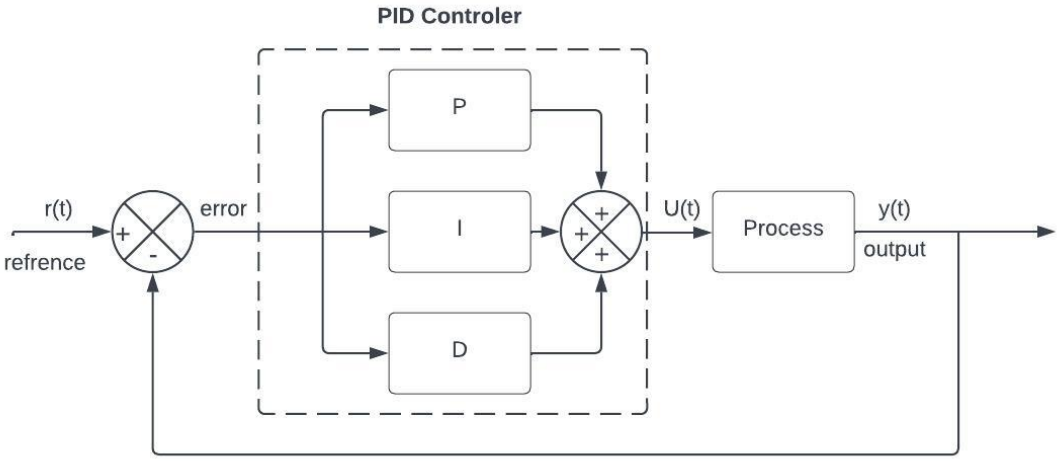


Figure 3.2: Closed loop PID control block diagram

3.4.4 Objective of the command

In our case, the objective is to design a classical (PID) controller for trajectory tracking. The controller parameters will be set empirically. However, the quadrotor is a six-degree-of-freedom robot, but the PID structure is mono-variable. Thus, we develop here a set of PIDs for all quadrotor measured variables.

Figure 3.3 shows the control structure when a (x, y, z, ψ) trajectory is planned. In general, stabilization of a quadrotor is achievable by a PD controller for each degree of freedom. However, a residual error is not canceled out at altitude z when the quadrotor is stationary, due to the effect of its weight. For this reason, a PD controller is designed for every degree of freedom except the altitude for which a PID controller has been chosen.

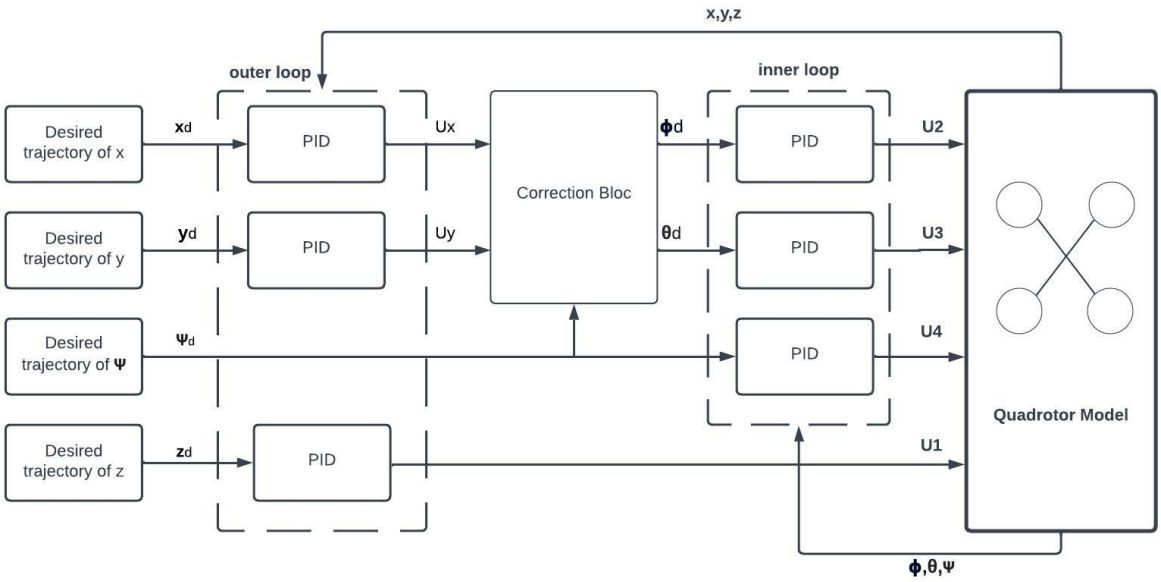


Figure 3.3: Block diagram of the PID command applied to the quadrotor

3.4.5 Control laws

In the quadrotor, we have to control six variables (6 DOF). The PID control laws for these variables are given by :

- **Pilotage loop**

$$U_2(t) = K_{P\phi}e_\phi(t) + K_{I\phi}\int_0^t e_\phi(\tau)d\tau + K_{D\phi}\dot{e}_\phi(t); e_\phi(t) = \phi_d(t) - \phi(t) \tag{3.10a}$$

$$U_3(t) = K_{P\theta}e_\theta(t) + K_{I\theta}\int_0^t e_\theta(\tau)d\tau + K_{D\theta}\dot{e}_\theta(t); e_\theta(t) = \theta_d(t) - \theta(t) \tag{3.10b}$$

$$U_4(t) = K_{P\psi}e_\psi(t) + K_{I\psi}\int_0^t e_\psi(\tau)d\tau + K_{D\psi}\dot{e}_\psi(t); e_\psi(t) = \psi_d(t) - \psi(t) \tag{3.10c}$$

- **Guidance loop**

$$U_x(t) = K_{Px}e_x(t) + K_{Ix}\int_0^t e_x(\tau)d\tau + K_{Dx}\dot{e}_x(t); e_x(t) = x_d(t) - x(t) \tag{3.11a}$$

$$U_y(t) = K_{Py}e_y(t) + K_{Iy}\int_0^t e_y(\tau)d\tau + K_{Dy}\dot{e}_y(t); e_y(t) = y_d(t) - y(t) \quad (3.11b)$$

$$U_1(t) = K_{Pz}e_z(t) + K_{Iz}\int_0^t e_z(\tau)d\tau + K_{Dz}\dot{e}_z(t); e_z(t) = z_d(t) - z(t) \quad (3.11c)$$

3.5 PID control implementation

We will present the simulation results from the application of the PID control technique seen previously on the quadrotor.

The simulation platform used is Simulink MATLAB R2021A. The simulation model of the quadrotor implemented in Simulink is complete without any simplification. The physical parameters of the quadrotor UAV are listed in the following table [17]:

Table 3.1: The physical parameters of the quadrotor

Parameter	**Value**
Mass (Kg)	0.486
Distance d(m)	0.25
I_x $\left(N.\text{m/rad/s}^2\right)$	3.8278×10^{-3}
I_y $\left(N.\text{m/rad/s}^2\right)$	3.8278×10^{-3}
I_z $\left(N.\text{m/rad/s}^2\right)$	7.6566×10^{-3}
$K_{fax}(N/\text{rad/s})$	5.5670×10^{-4}
$K_{fay}(N/\text{rad/s})$	5.5670×10^{-4}
$K_{faz}(N/\text{rad/s})$	6.3540×10^{-4}
$K_{fdx}(N/\text{m/s})$	5.5670×10^{-4}
$K_{fdy}(N/\text{m/s})$	5.5670×10^{-4}
$K_{fdz}(N/\text{m/s})$	6.3540×10^{-4}
Lift force coefficient K_p(N.m/rad/s)	2.9842×10^{-5}
Drag force coefficient C_D(N.m/rad/s)	3.2320×10^{-7}
Rotor inertia J_r $\left(N.\text{m/rad/s}^2\right)$	2.8385×10^{-5}

3.5.1 Implementation

The block diagram of the quadrotor simulation model in Matlab Simulink is shown in Figure 3.4.

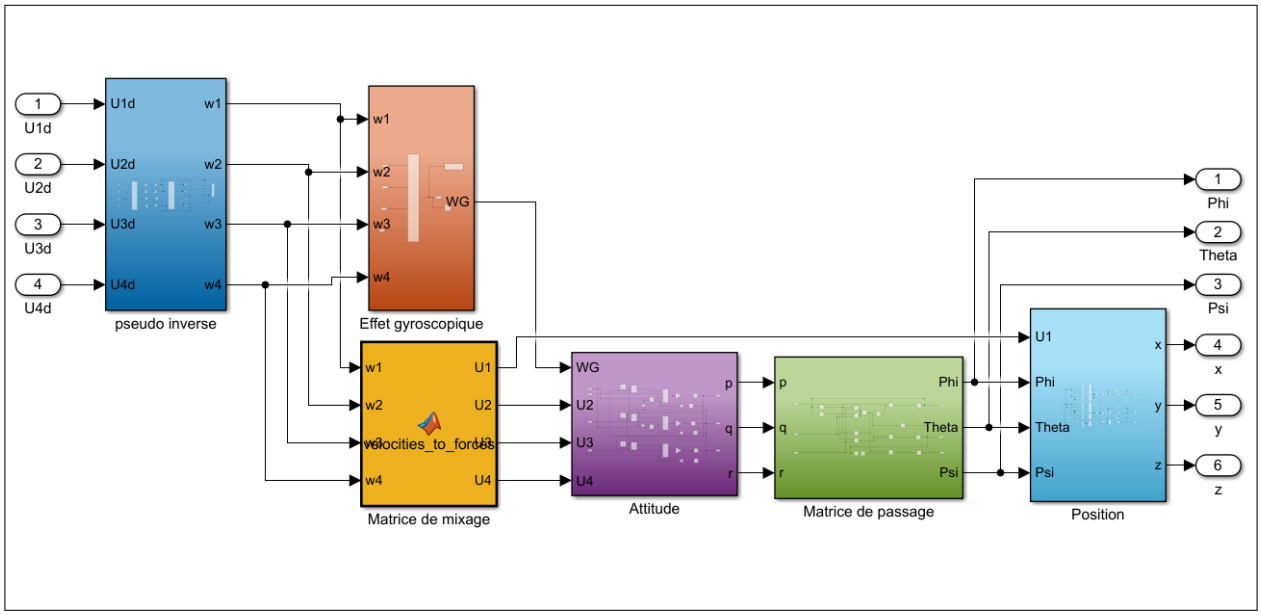


Figure 3.4: The block diagram of the quadrotor simulation model

The block diagram of the PID control simulation model in Matlab Simulink applied to the quadrotor is shown in Figure 3.5.

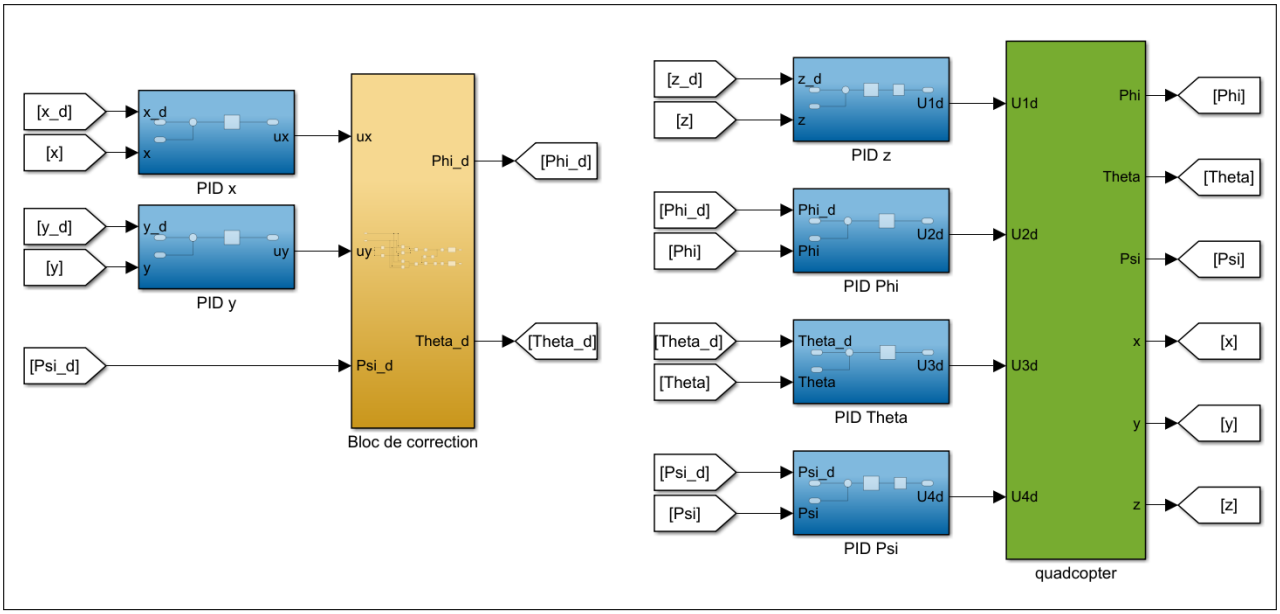


Figure 3.5: The block diagram of the PID control

The block diagram of dynamic inversion control is shown in Figure 3.6.

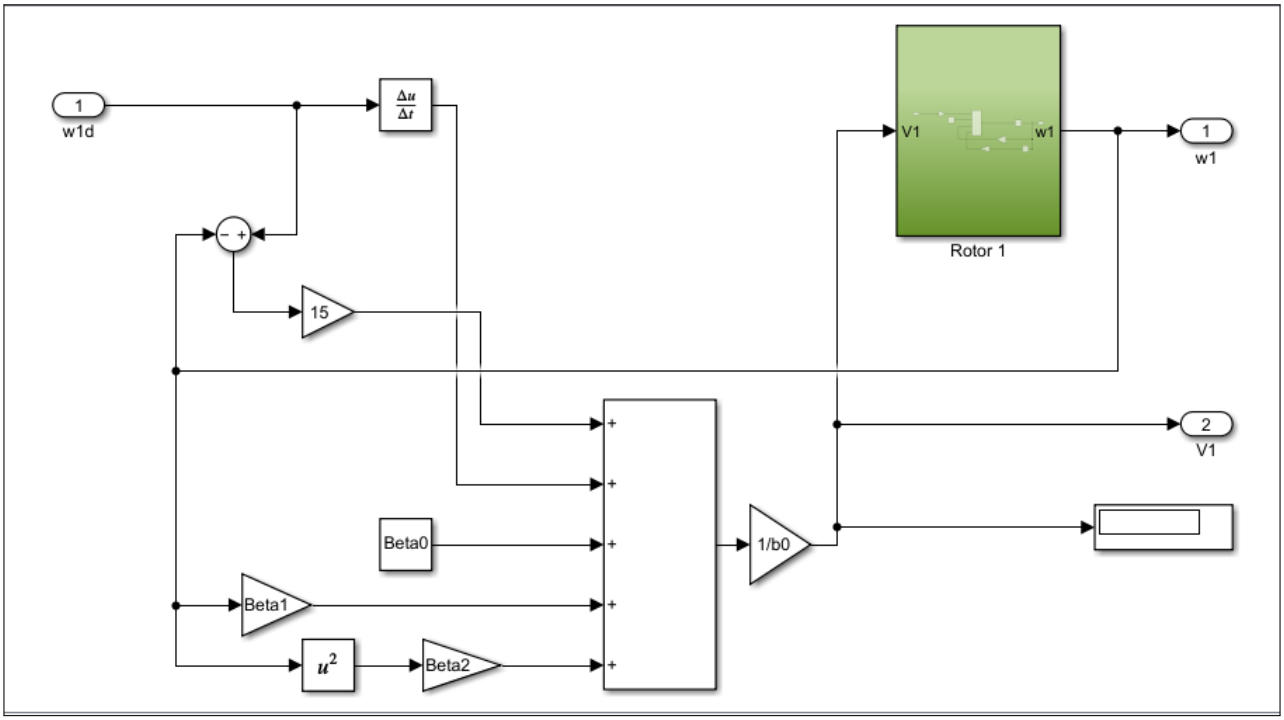


Figure 3.6: Block diagram of dynamic inversion control

The rotor parameters are shown in the following table [17]:

Table 3.2: The rotor parameters

β_0	189.63
β_1	6.0612
β_2	0.0122
b	280.19

3.5.2 Trajectory tracking

The Parameters of the PID controller are shown in the following table:

Table 3.3: The Parameters of the PID controller

	Kp	**Ki**	**Kd**
U_x	0.07	0	0.5
U_y	0.1	0	0.7
U_1	18	8	12
U_2	1	0	0.8
U_3	0.5	0	4
U_4	1.2	0	1.2

Example of the command U_x of the PID controller in Matlab Simulink:

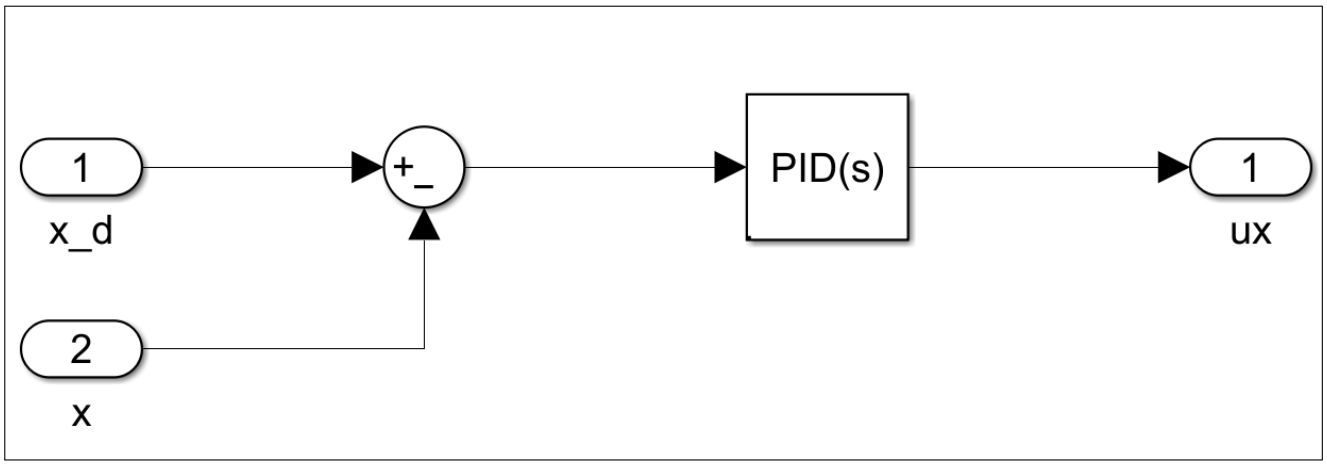


Figure 3.7: Longitudinal motion (x) PID controller

The first scenario

The quadrotor will fly at an altitude of 25m. We have set the duration of the mission to 50 seconds. The trajectory is given by the equations:

$$x_d = sin(0.25t + 30) \text{ m} \tag{3.12a}$$

$$y_d = cos(0.25t + 30) \text{ m} \tag{3.12b}$$

$$z_d = 25 \text{ m} \tag{3.12c}$$

And the ψ_d is a Pulse input with amplitude $\frac{\pi}{8}$, and period 25 seconds and pulse width 50% of the period.

Figure 3.8 shows the evolution of the quadrotor position along the three axes x, y and z by the PID controller.

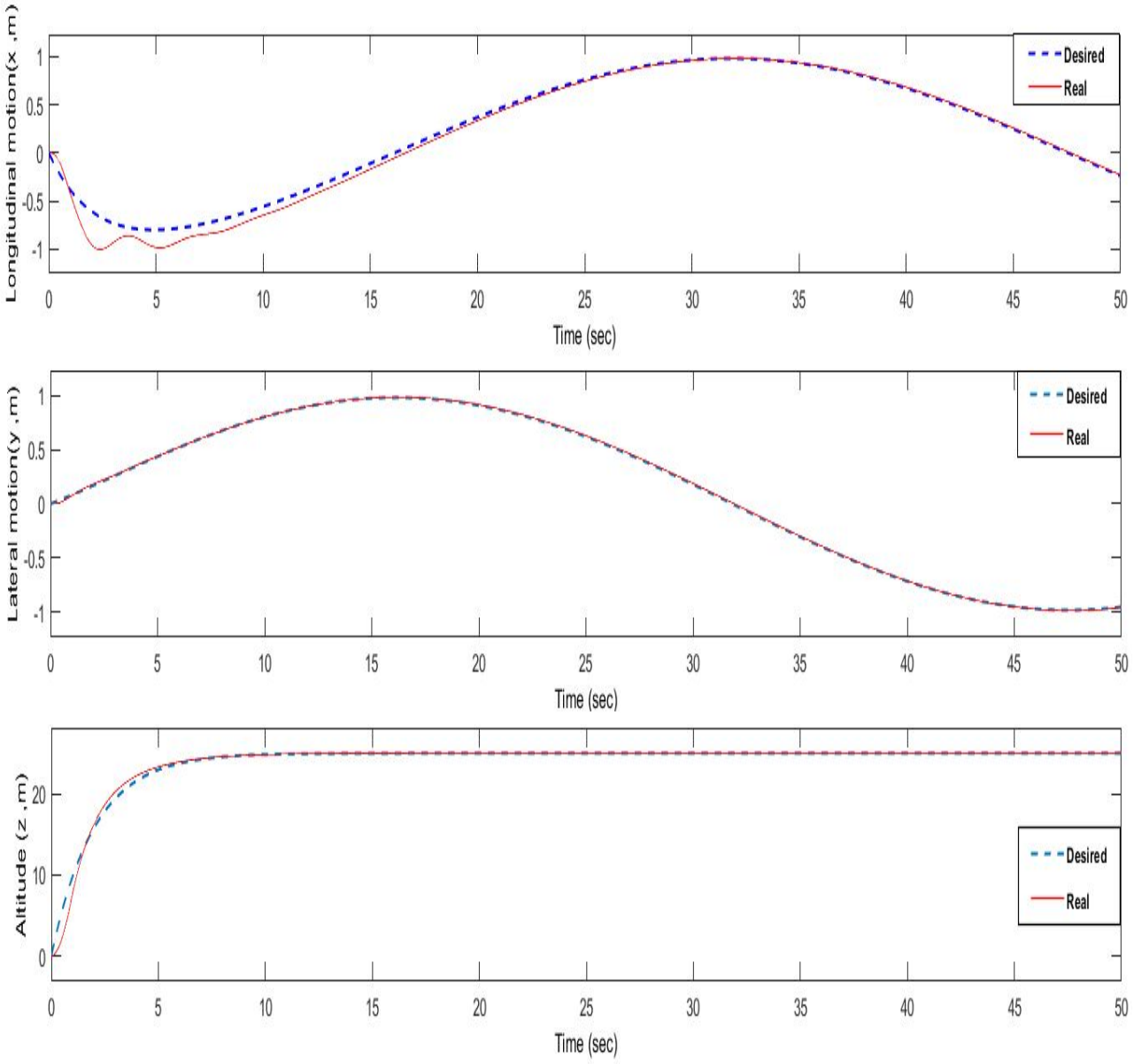


Figure 3.8: Generation of the desired trajectory by PID control

Figure 3.9 illustrates the evolution of the attitude (roll, pitch and yaw angles) of the quadrotor during the realization of the desired trajectory.

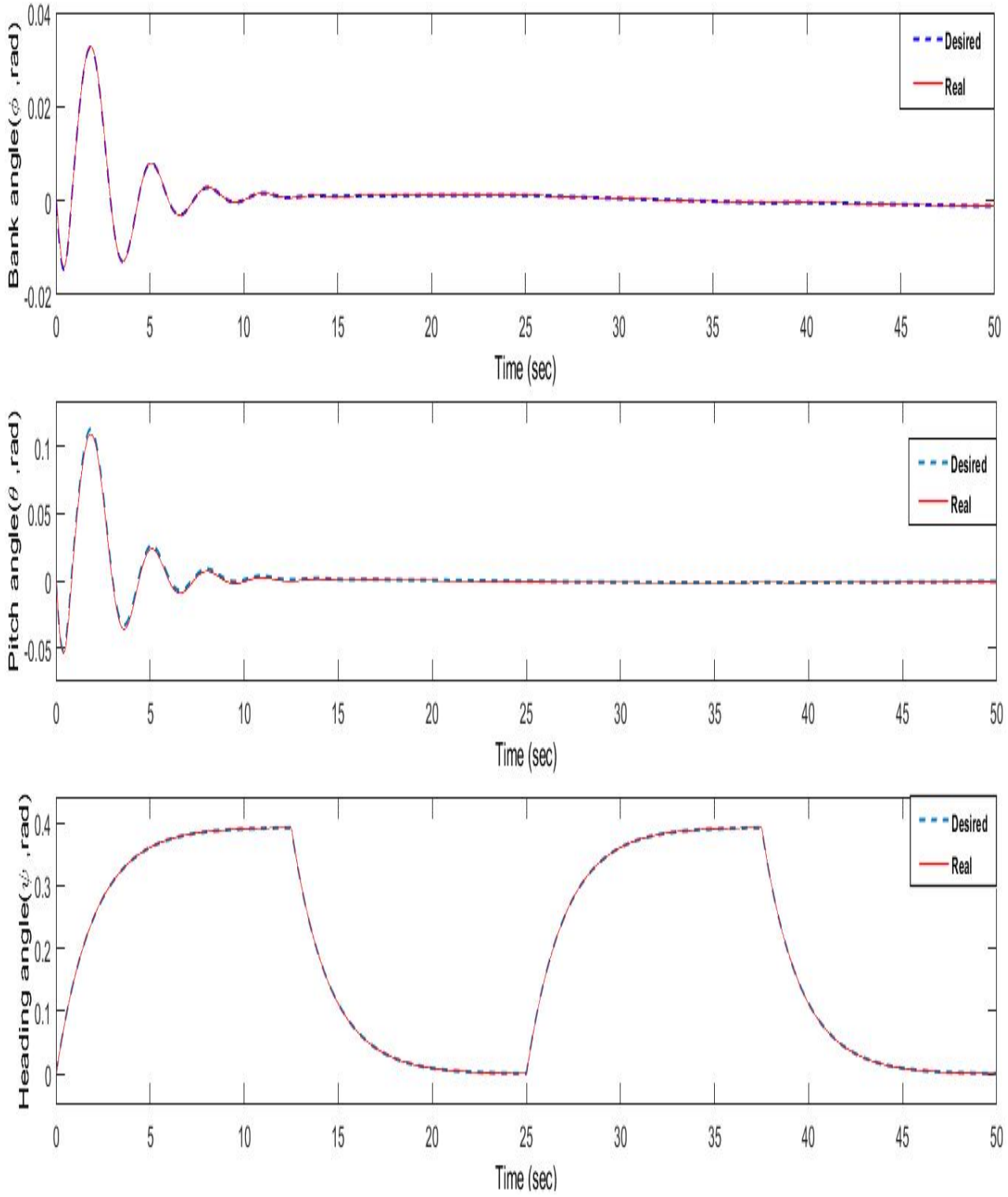


Figure 3.9: The attitude of the quadrotor achieving the desired trajectory

Figure 3.10 shows the result of the 3D helical trajectory by the PID controller.

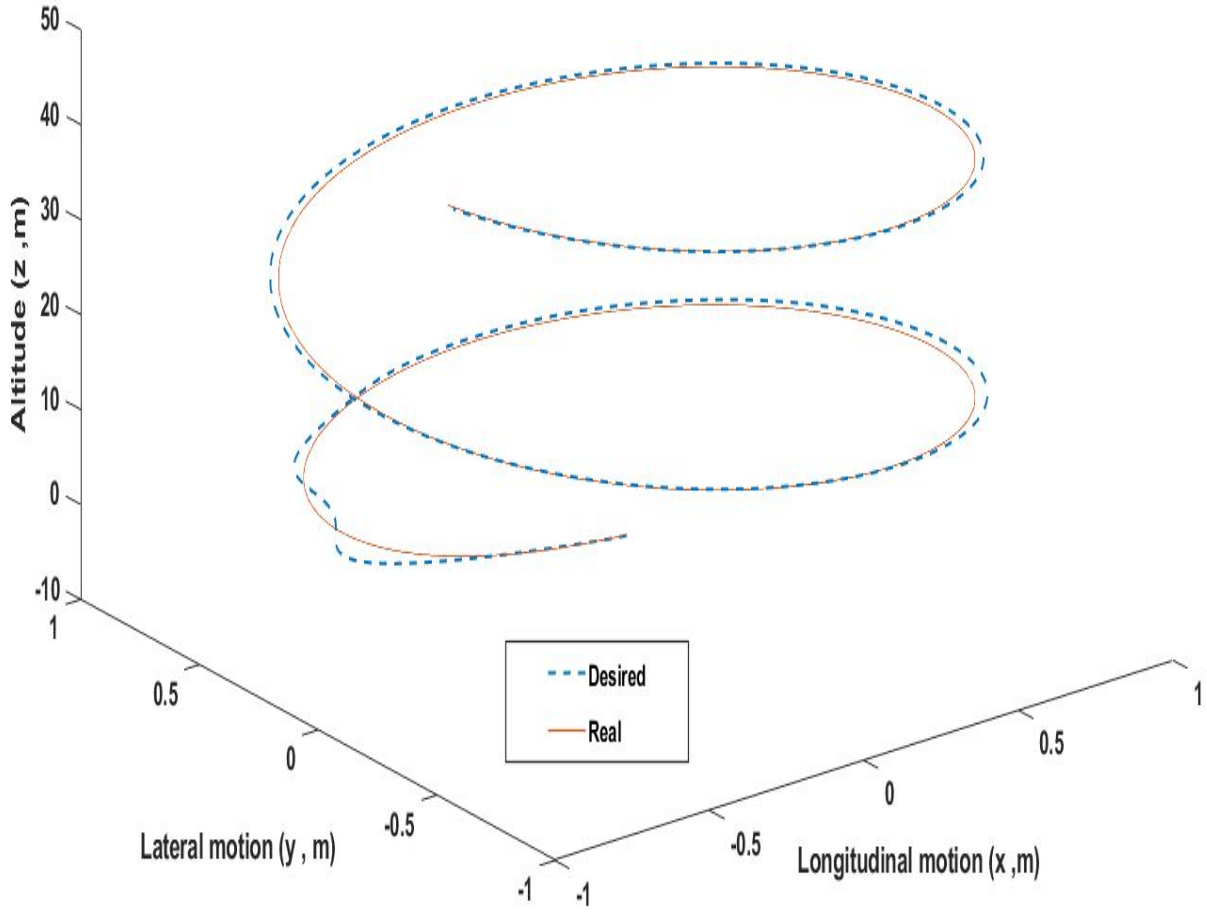


Figure 3.10: 3D helical trajectory by the PID controller (where z = t)

Now we present the commands of PID control, as well as the angular speeds of the four rotors.

Figure 3.11 present the commands U_1, U_2, U_3 and U_4 of PID control and Figure 3.12 present the rotational speeds of the four rotors during the execution of the desired trajectory.

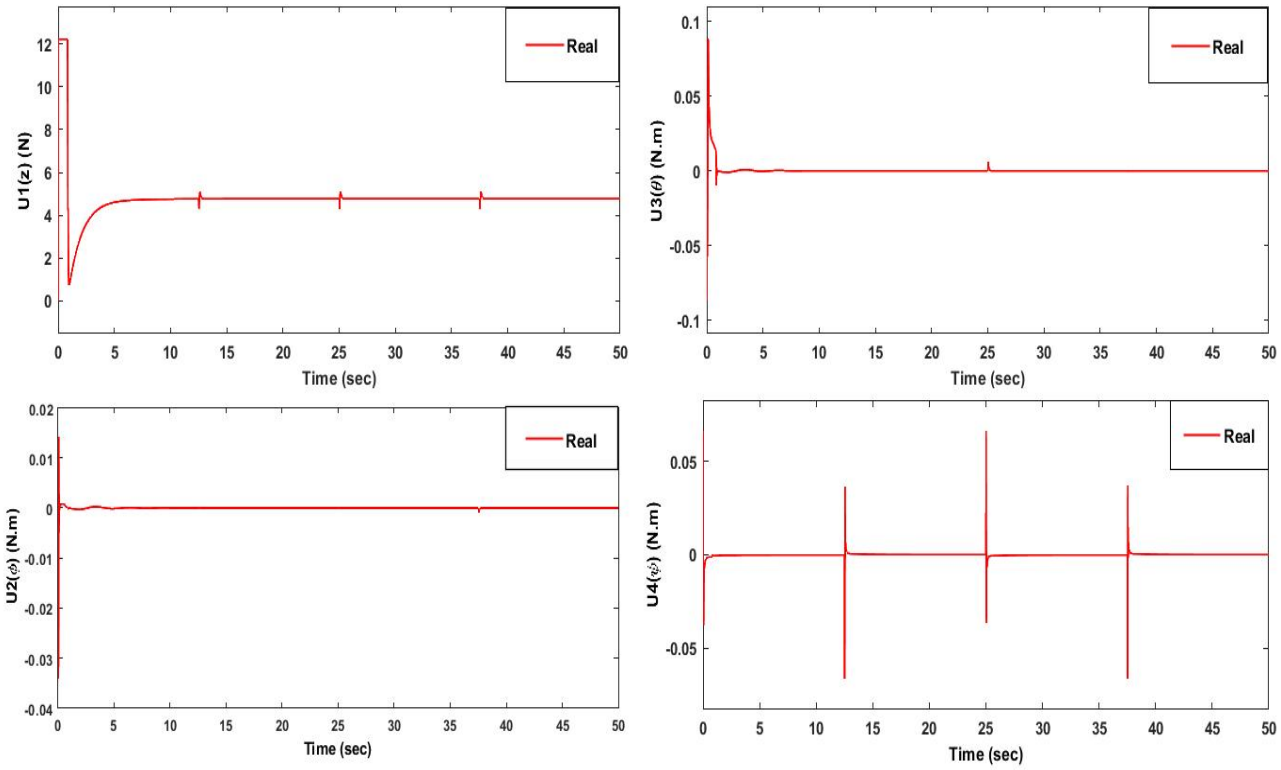


Figure 3.11: The commands of the PID control technique

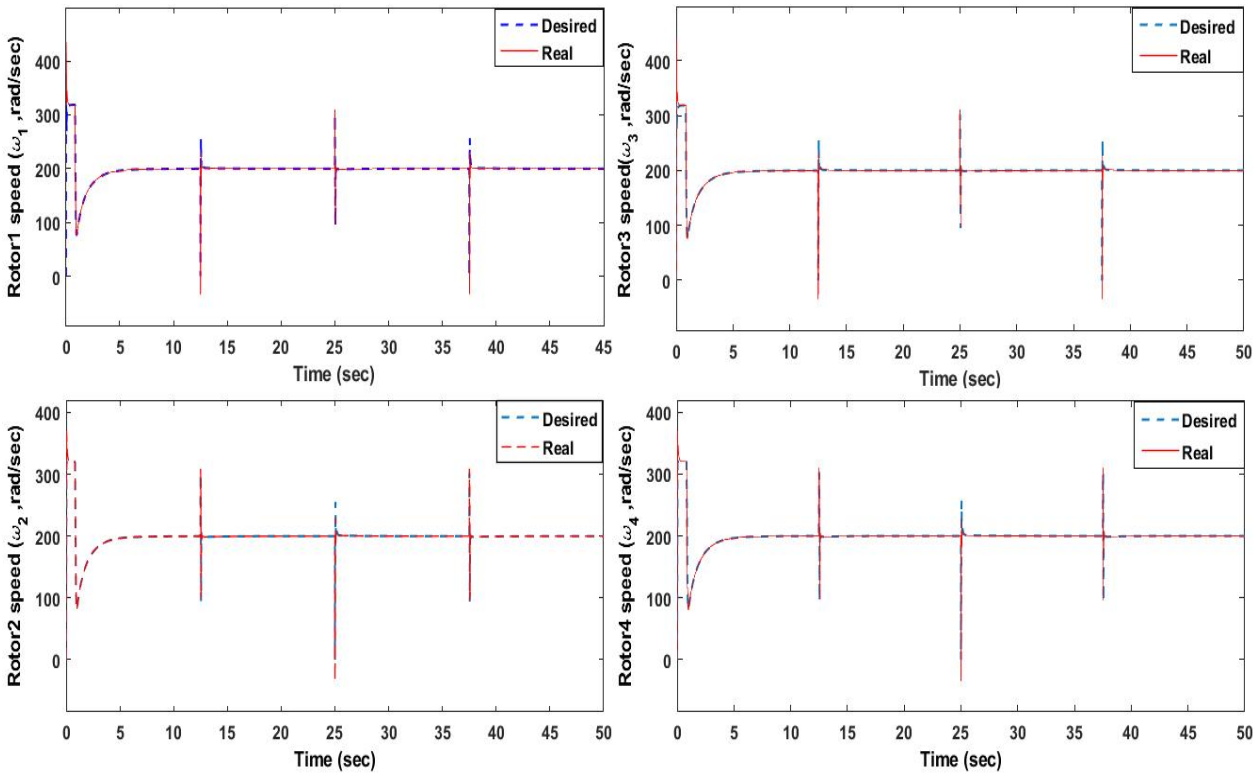


Figure 3.12: The rotational speeds of the four rotors

The second scenario

The "circle of runway" navigation scenario or also called aerodrome circuit is a maneuver that aircraft make around the runway in a rectangular shape, with turns to the left, and at a desired height above the aerodrome. In the case of these simulations, we set the duration of the mission to 50 seconds. The reference trajectory is given by the equations:

$$x_d(t) = \begin{cases} 0 \text{ m} & \text{for } t \in [0;5[\\ 5 \text{ m} & \text{for } t \in [5;20[\\ -7 \text{ m} & \text{for } t \in [20;50] \end{cases} \tag{3.13a}$$

$$y_d(t) = \begin{cases} 0 \text{ m} & \text{for } t \in [0;15[\\ 5 \text{ m} & \text{for } t \in [15;35[\\ -7 \text{ m} & \text{for } t \in [35;50] \end{cases} \tag{3.13b}$$

$$z_d(t) = \begin{cases} 5 \text{ m} & \text{for } t \in [0;40[\\ 0 \text{ m} & \text{for } t \in [40;50[\end{cases} \tag{3.13c}$$

And the ψ_d is a Pulse input with amplitude $\frac{\pi}{8}$, and period 25 seconds and pulse width 50% of the period. To generate the planned trajectory, we used the block *Signal builder*.

Figure 3.13 shows the evolution of the quadrotor position along the three axes x, y and z by the PID controller.

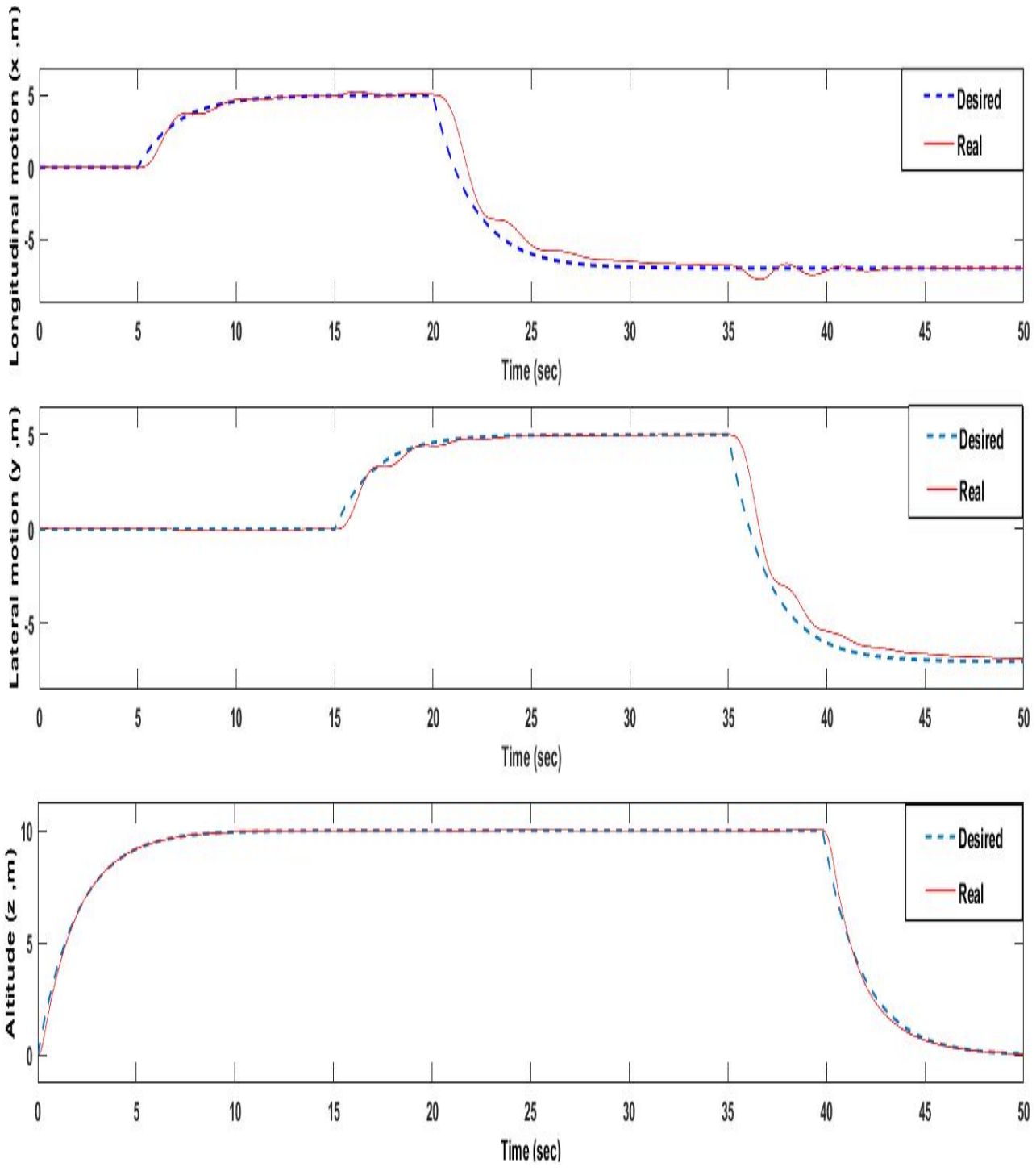


Figure 3.13: Generation of the desired trajectory by the PID controller

Figure 3.14 illustrates the evolution of the attitude (roll, pitch and yaw angles) of the quadrotor during the realization of the desired trajectory.

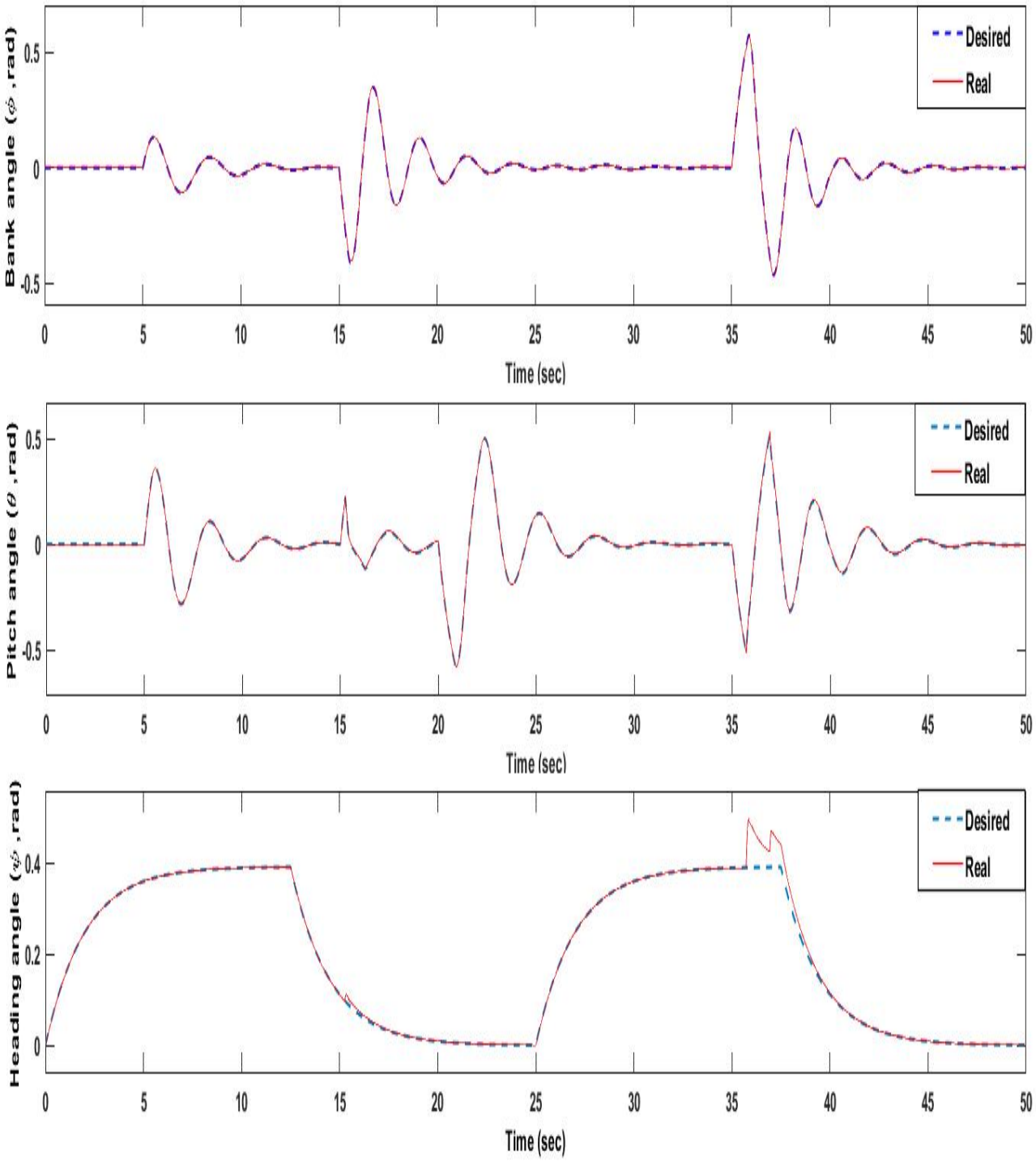


Figure 3.14: The attitude of the quadrotor achieving the desired trajectory

Figure 3.15 shows the result of the 3D visualization of the trajectory by the PID controller.

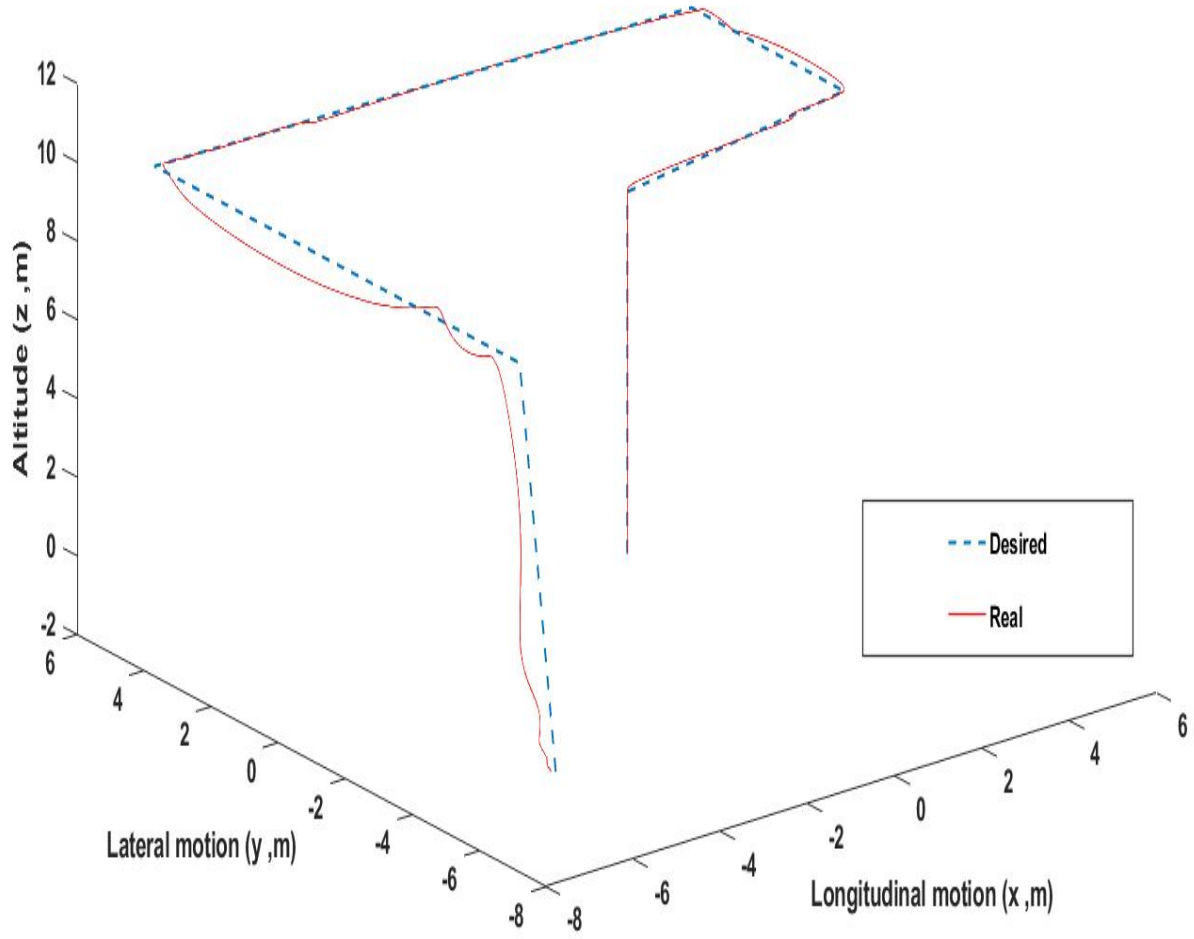


Figure 3.15: 3D visualization of the trajectory by the PID controller

Now we present the commands of the PID controller, as well as the angular speeds of the four rotors.

Figure 3.16 present the commands U1, U2, U3 and U4 of the PID controller and Figure 3.17 present the rotational speeds of the four rotors during the execution of the desired trajectory.

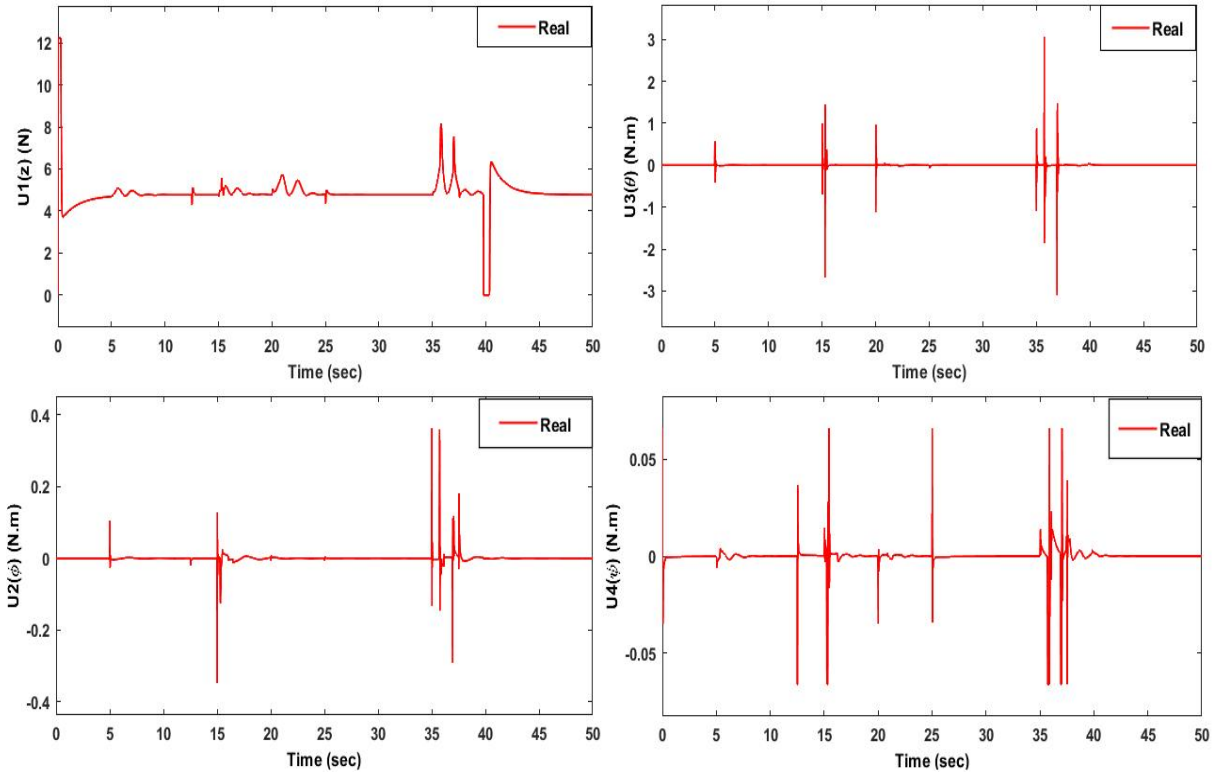


Figure 3.16: The commands of the PID controller technique

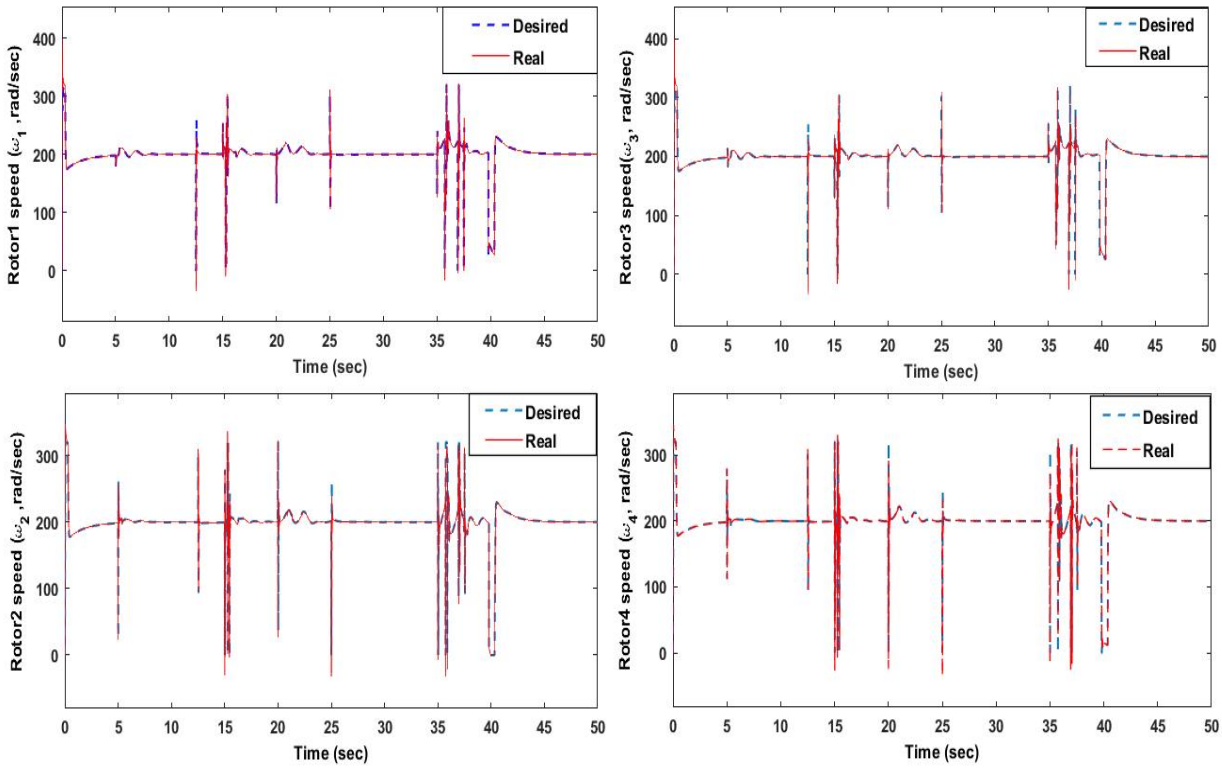


Figure 3.17: The rotational speeds of the four rotors

3.6 Backsepping controller

The backstepping technique was developed in the early 90s. The arrival of backstepping control gave a new lease of life to the control of nonlinear systems, which despite the great progress made, lacked general approaches. Backstepping is a systematic, recursive method for synthesizing nonlinear control laws based on Lyapunov's stability principle, which can be applied to a wide range of nonlinear systems.

The basic idea behind Backstepping control is to make looped systems equivalent to Lyapunov-stable cascaded first-order subsystems, leading to asymptotic global stability. In other words, it's a multi-stage method. At each stage of the process, a virtual control is generated to ensure convergence of the system towards its equilibrium state. This can be achieved using Lyapunov functions, which ensure step-by-step stabilization of each synthesis step. It should be remembered that Backstepping is characterized by:

- It applies to looped systems with strict feedback, i.e. the derivative of each component of the state vector must be a function of the preceding components and depend additively on the following component.
- We start with the first differential equation of the $\dot{x}_1$ system, which is further away from the control input U, and complete the control law only in the last step.
- In our case, we can synthesize the control laws for the control loop (U_2, U_3, U_4) and the guidance loop (U_x, U_y, U_1), forcing the system to follow the desired trajectory.

Purpose of the Control

In the context of trajectory tracking the control objective will be a pursuit problem, we want the state $x(t)$ to follow a time-varying setpoint $x_d(t)$ and that when t tends to t_f.

Description of design steps

The Backstepping controller design method is mainly based on Lyapunov's stability theory of dynamical systems. The essence of the theory is presented in (Khalil, 1992). In our work,

we have to consider the quadrotor roll synthesis nonlinear state model to illustrate the design method:

$$\ddot{\phi} = \frac{1}{I_x}[(I_y - I_z)\dot{\theta}\dot{\psi} - K_{fax}\dot{\phi} - J_r\bar{\omega}\dot{\theta} + U_2] \tag{3.14}$$

Consider the state vector:

$$X = \begin{bmatrix} \phi & \dot{\phi} & \theta & \dot{\theta} & \psi & \dot{\psi} & x & \dot{x} & y & \dot{y} & z & \dot{z} \end{bmatrix}^T \tag{3.15}$$

From (3.2) and (3.3) and (3.13), we obtain the following state representation:

$$\dot{x}_1 = x_2 \tag{3.16a}$$

$$\dot{x}_2 = a_1 x_4 x_6 + a_2 x_2^2 + b_1 U_\phi \tag{3.16b}$$

$$\dot{x}_3 = x_4 \tag{3.16c}$$

$$\dot{x}_4 = a_4 x_2 x_6 + a_5 x_4^2 + b_2 U_\theta \tag{3.16d}$$

$$\dot{x}_5 = x_6 \tag{3.16e}$$

$$\dot{x}_6 = a_7 x_4 x_2 + a_8 x_6^2 + b_3 U_\psi \tag{3.16f}$$

$$\dot{x}_7 = x_8 \tag{3.16g}$$

$$\dot{x}_8 = a_9 x_8 + U_x \frac{U_z}{m} \tag{3.16h}$$

$$\dot{x}_9 = x_1 0 \tag{3.16i}$$

$$\dot{x}_{10} = a_{10} x_{10} + U_y \frac{U_z}{m} \tag{3.16j}$$

$$\dot{x}_{11} = x_1 2 \tag{3.16k}$$

$$\dot{x}_{12} = a_{11}x_{12} + \frac{Cx_1Cx_3}{m}U_z - g \tag{3.16l}$$

$$a_1 = \left(\frac{I_y - I_z}{I_x}\right), a_2 = \frac{-K_{fax}}{I_x}, a_3 = \frac{-J_r}{I_x} \tag{3.17a}$$

$$a_4 = \left(\frac{I_z - I_x}{I_y}\right), a_5 = \frac{-K_{fay}}{I_y}, a_6 = \frac{J_r}{I_y} \tag{3.17b}$$

$$a_7 = \left(\frac{I_x - I_y}{I_z}\right), a_8 = \frac{-K_{faz}}{I_z}, a_9 = \frac{-K_{ftx}}{m} \tag{3.17c}$$

$$a_{10} = \frac{-K_{fty}}{m}, a_{11} = \frac{-K_{fzz}}{m} \tag{3.17d}$$

$$b_1 = \frac{d}{I_x}, b_2 = \frac{d}{I_y}, b_3 = \frac{1}{I_z} \tag{3.17e}$$

Hence the equation (3.12) becomes in the following state form:

$$\dot{x}_1 = x_2 \tag{3.18a}$$

$$\dot{x}_2 = a_1 x_4 x_6 + a_2 x_2^2 + a_3 x_4 \bar{\omega} + b_1 U_2 \tag{3.18b}$$

- **First step**

Let's define the first variable in the procedure e_1 to be the error between the state and the desired state such that $e_1 = x_1 - x_{1d}$. The derivative concerning time is:

$$\dot{e}_1 = x_2 - \dot{x}_{1d} = x_2 - x_{2d} \tag{3.19}$$

And a second Backstepping variable denoted $z_1 = x_2 - \bar{x_2}$ with $\bar{x_2}$ a virtual control law to be determined later. To find this control law we construct a partial Lyapunov function of quadratic

type :

$$V_1(e) = \frac{1}{2}e_1^2 \tag{3.20}$$

Its derivative concerning time is:

$$\dot{V}_1(e_1) = e_1.\dot{e_1} = e_1(x_2 - \dot{x_{1d}}) \tag{3.21}$$

$\bar{x_2}$ is chosen such that $\dot{V}_1(e_1)$ is negative definite:

$$\bar{x_2} = \dot{x_{1d}} - k_1 e_1 \tag{3.22}$$

Where $k_1 > 0$ is a positive regulation constant. Noting that $\bar{x_2}$ was chosen so as to have $\dot{V}_1(e_1) < 0$. Substituting $\bar{x_2}$ into $\dot{V}_1(e_1)$ we find:

$$\dot{V}_1(e_1) = e_1.(-k_1 e_1 + z_1) = -k_1 e_1^2 + e_1.z_1 \tag{3.23a}$$

$$z_1 = x_2 - \dot{x}_{1d} + k_1 e_1 \tag{3.23b}$$

For global stability, the last term $e_1.z_1$ will be eliminated in the next step.

- **Second step**

We now need to define a new system from this new state. It is typically referred to as the "augmented system". We also note that in the second design step, the $x_1$state will no longer appear. This is implicitly taken into account via the error state. The augmented system can be written as follows:

$$\dot{e}_1 = -k_1 e_1 + z_1 \tag{3.24a}$$

$$\dot{z}_1 = a_1 x_4 x_6 + a_2 x_2^2 + b_1 U_2 - \ddot{x}_{1d} + k_1 \dot{e}_1 \tag{3.24b}$$

Consider the candidate Lyapunov function $V_2(e_1, z_1)$ of the augmented system:

$$V_2(e_1, z_1) = V_1(e_1) + \frac{1}{2}z_1^2 \tag{3.25}$$

Its derivative concerning time is:

$$\dot{V}_2(e_1, z_1) = e_1\dot{e}_1 + z_1\dot{z}_1 = e_1(-k_1e_1 + z_1) + z_1(a_1x_4x_6 + a_2x_2^2 + b_1U_2 - \ddot{x}_{1d} + k_1\dot{e}_1) \tag{3.26}$$

By choosing the following controller for roll:

$$U_2 = \frac{1}{b_1}(-a_1x_4x_6 - a_2x_2^2 - e_1 + \ddot{x}_{1d} - k_1\dot{e}_1 - k_2z_1) \tag{3.27}$$

We obtain :

$$\dot{V}_2(e, z_1) = -k_1e^2 - k_2z_1^2 \tag{3.28}$$

With : k_1, k_2 are positive tuning gains.

This ensures that the system is Globally Asymptotically Stable (GAS). Our control objective has therefore been achieved.

Following the same steps for the roll controller, the control input U_3 responsible for generating the pitch rotation and U_4 responsible for generating the yaw rotation are given by:

$$U_2 = \frac{1}{b_1}[-a_1x_4x_6 - a_2x_2^2 - e_1 + \ddot{x}_{1d} - k_1(-k_1e_1 + z_1) - k_2z_1] \tag{3.29a}$$

$$U_3 = \frac{1}{b_2}[-a_4x_2x_6 - a_5x_4^2 - e_2 + \ddot{x}_{4d} - k_3(-k_3e_2 + z_2) - k_4z_2] \tag{3.29b}$$

$$U_4 = \frac{1}{b_3}[-a_7x_2x_4 - a_8x_6^2 - e_3 + \ddot{x}_{6d} - k_5(-k_5e_3 + z_3) - k_6z_3] \tag{3.29c}$$

The altitude command U_1, the longitudinal and lateral commands (U_x, U_y) are obtained using

the same approach described previously, we find :

$$U_x = \frac{m}{U_z}[-a_9 x_8 + \ddot{x}_d + k_7(-k_7 e_7 + e_8) + k_8 e_8 + e_7] \tag{3.30a}$$

$$U_y = \frac{m}{U_z}[-a_{10} x_{10} + \ddot{y}_d + k_9(-k_9 e_9 + e_{10}) + k_{10} e_{10} + e_9] \tag{3.30b}$$

$$U_1 = \frac{1}{\cos x_1 . \cos x_3}[-a_{11} x_{12} + g + \ddot{z}_d + k_{11}(-k_{11} e_{11} + e_{12}) + k_{12} e_{12} + e_{11}] \tag{3.30c}$$

The z_i represent the tracking errors for the state variables and the different k_i are adjustment gains for each degree of freedom.

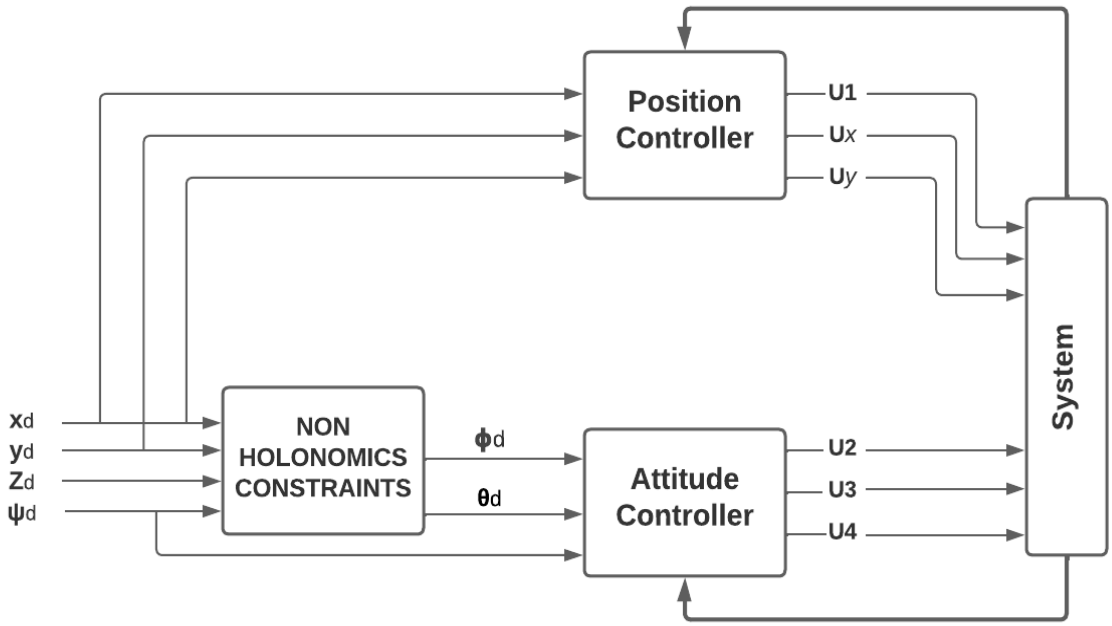


Figure 3.18: Block diagram of the backstepping controller applied to the quadrotor

3.7 Backstepping control implementation

We will present the simulation results from the application of the Backstepping control technique seen previously on the quadrotor.

The simulation platform used is Simulink MATLAB R2021A. The simulation model of the quadrotor implemented in Simulink is complete without any simplification.

3.7.1 Implementation

The block diagram of the Backstepping controller simulation model in Matlab Simulink applied to the quadrotor is shown in Figure 3.19.

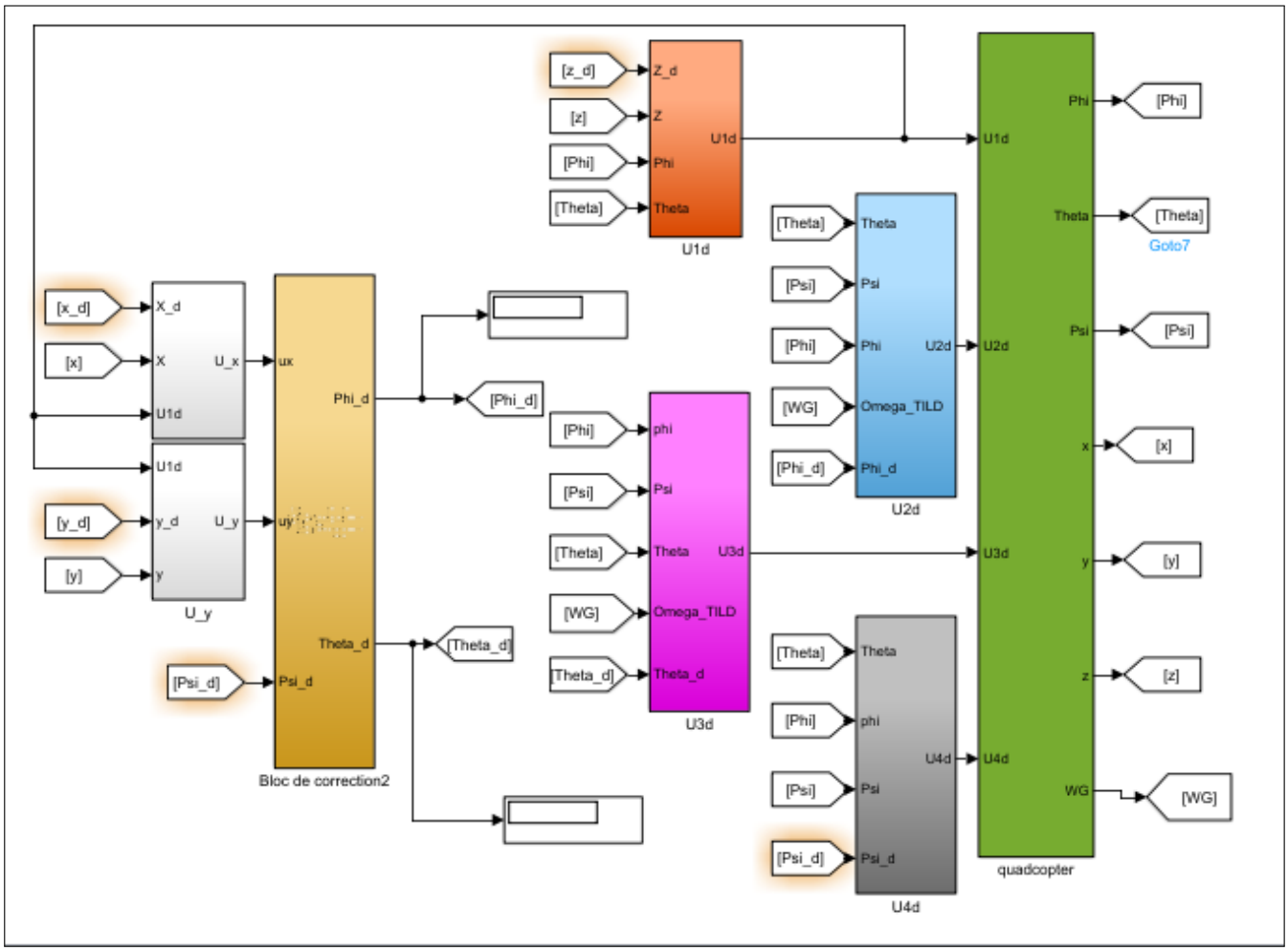


Figure 3.19: The block diagram of the Backstepping controlller

3.7.2 Trajectory tracking

Example of the command U_2 of the Backstepping controller in Matlab Simulink:

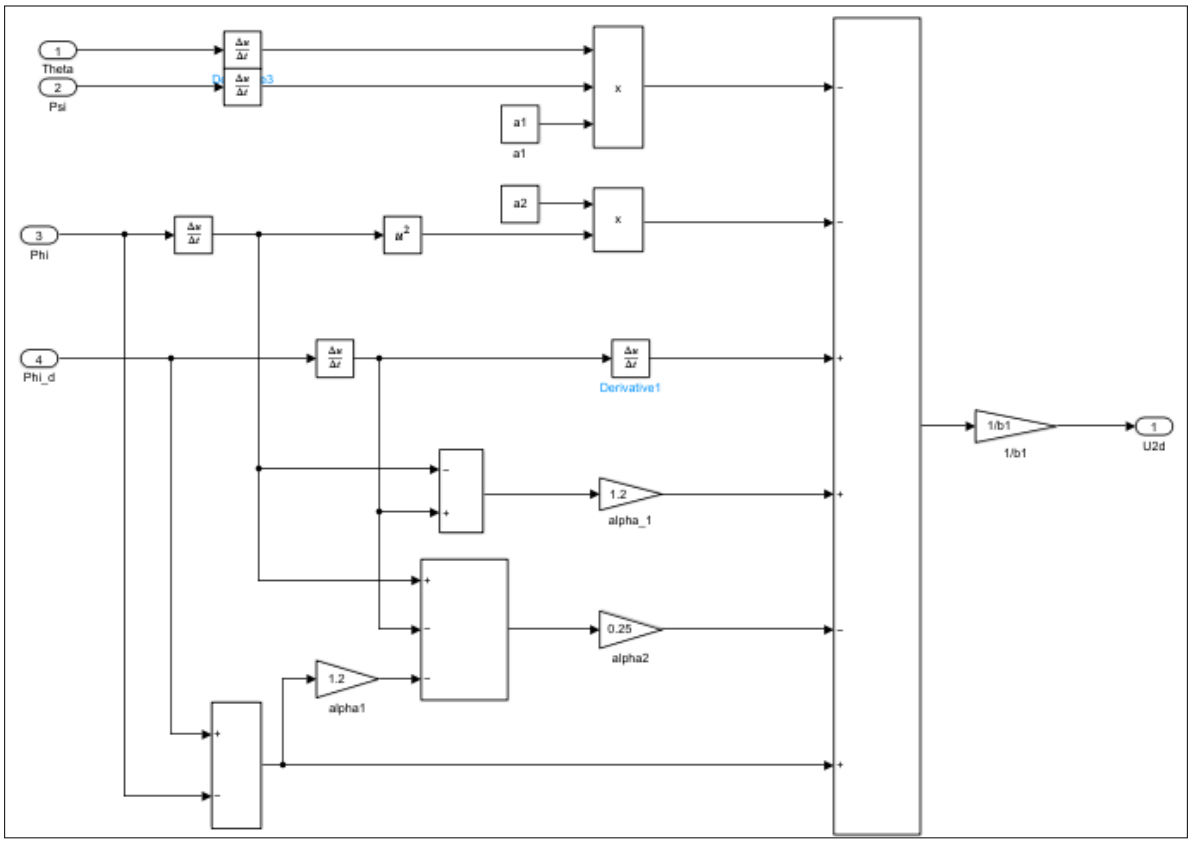


Figure 3.20: The control input U2 of the Backstepping controller

The first scenario

The quadrotor will fly. We have set the duration of the mission to 50 seconds. The trajectory is given by the equations:

$$x_d = sin(0.25t + 30) \text{ m} \tag{3.31a}$$

$$y_d = cos(0.25t + 30) \text{ m} \tag{3.31b}$$

$$z_d = t \tag{3.31c}$$

And the ψ_d is a Pulse input with amplitude 1, and period 20 seconds and a pulse width 50% of the period.

Figure 3.21 shows the evolution of the quadrotor position along the three axes x, y and z by the Backstepping controller.

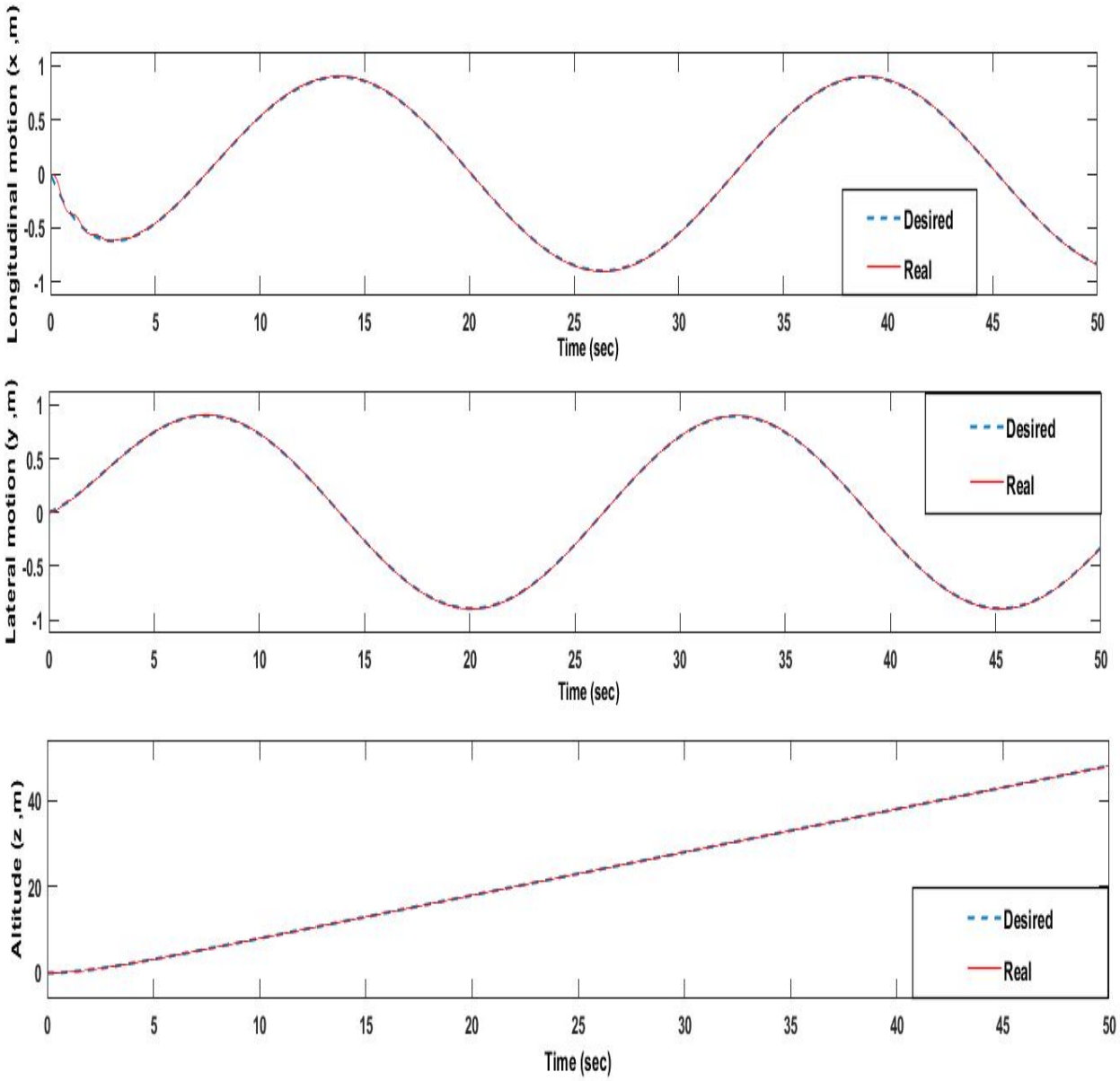


Figure 3.21: Generation of the desired trajectory by the Backstepping controller

Figure 3.22 illustrates the evolution of the attitude (roll, pitch and yaw angles) of the quadrotor during the realization of the desired trajectory.

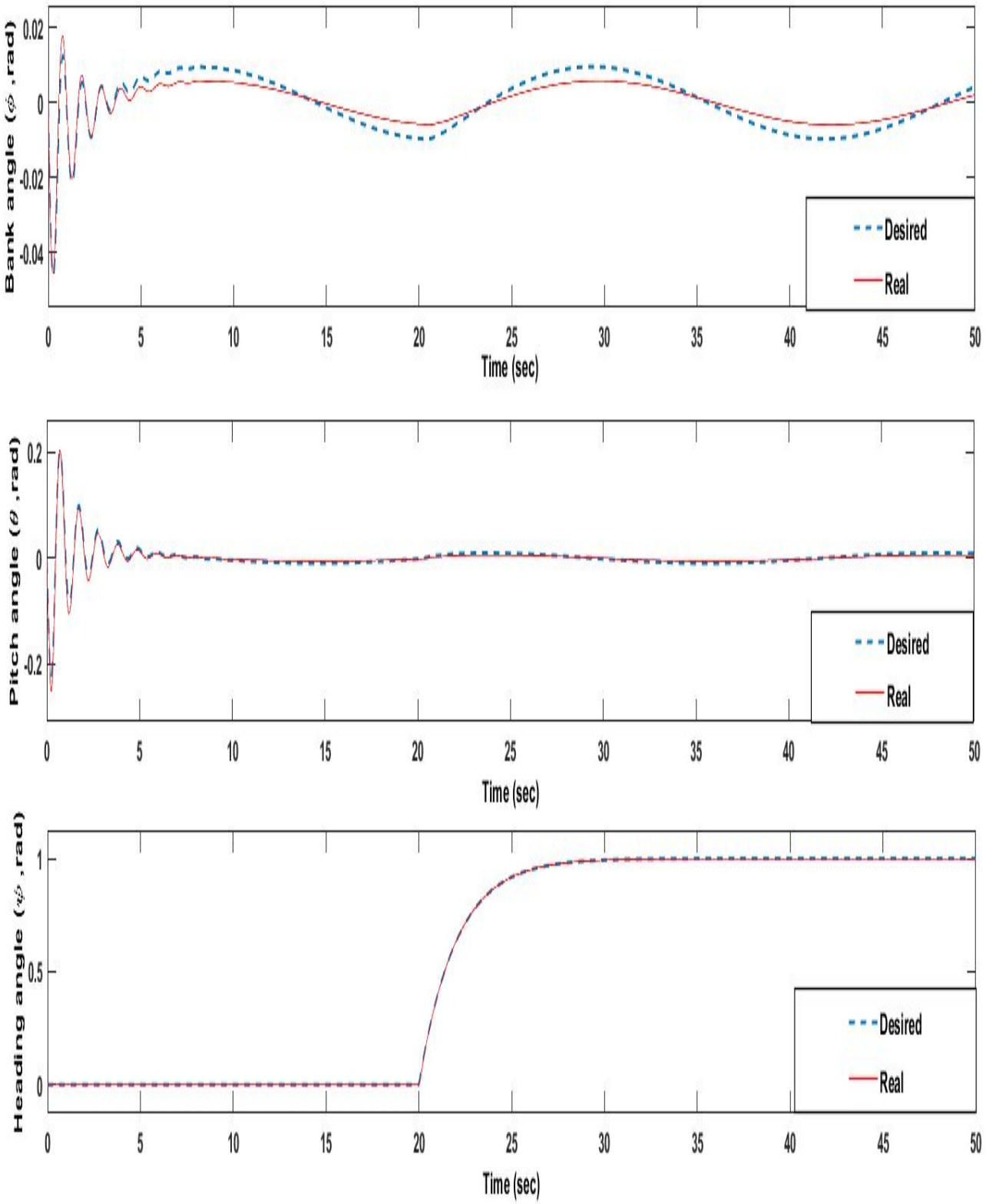


Figure 3.22: The attitude of the quadrotor achieving the desired trajectory

Figure 3.23 shows the result of the 3D helical trajectory by the Backstepping controller.

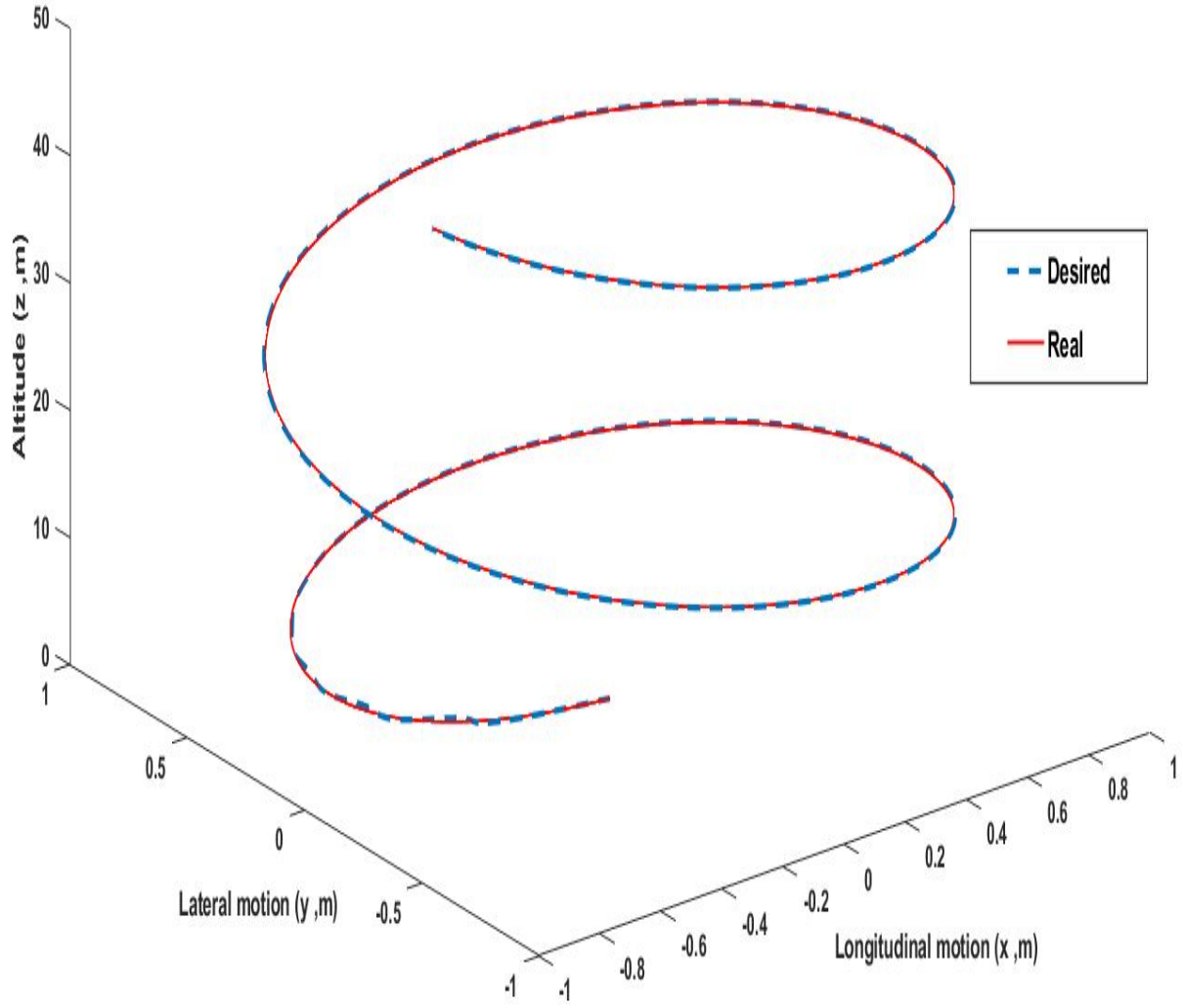


Figure 3.23: 3D helical trajectory by the Backstepping controller (where z = t)

Now we present the control inputs of Backstepping control, as well as the angular speeds of the four rotors.

Figure 3.24 present the control inputs U_1, U_2, U_3 and U_4 of PID control and Figure 3.25 present the rotational speeds of the four rotors during the execution of the desired trajectory.

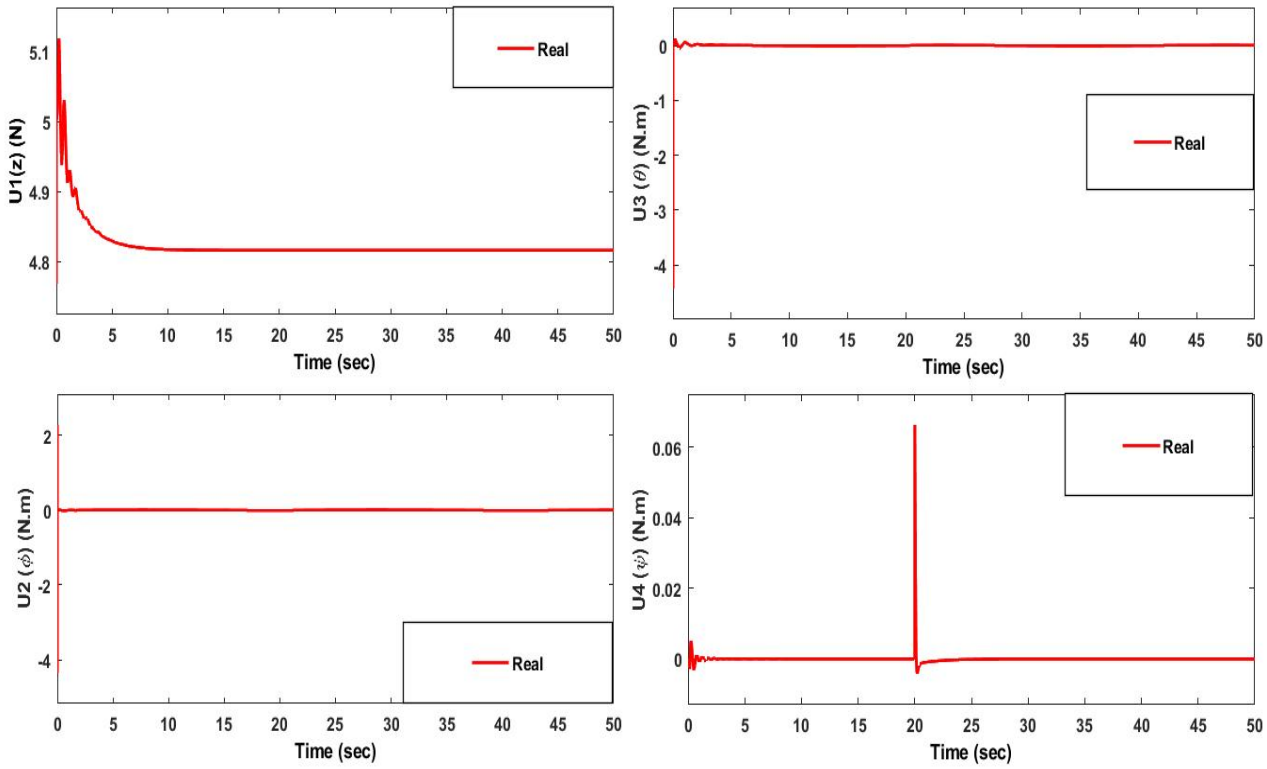


Figure 3.24: The control inputs of the Backstepping controller

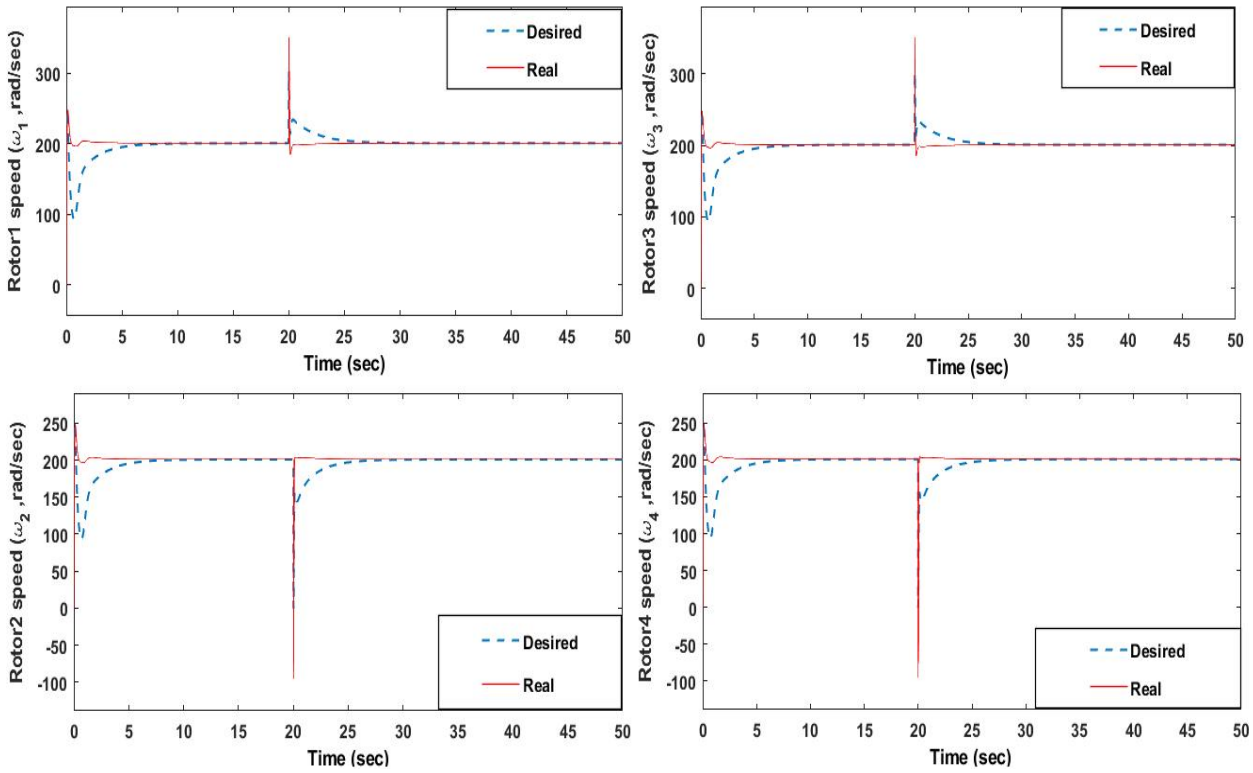


Figure 3.25: The rotational speeds of the four rotors for $z = t$, $\psi = 1$

The second scenario

The "circle of runway" navigation scenario or also called aerodrome circuit is a maneuver that aircraft make around the runway in a rectangular shape, with turns to the left, and at a desired height above the aerodrome. In the case of these simulations, we set the duration of the mission to 50 seconds. The reference trajectory is given by the equations:

$$x_d(t) = \begin{cases} 0 \text{ m} & \text{for } t \in [0;5[\\ 5 \text{ m} & \text{for } t \in [5;20[\\ -7 \text{ m} & \text{for } t \in [20;50] \end{cases} \tag{3.32a}$$

$$y_d(t) = \begin{cases} 0 \text{ m} & \text{for } t \in [0;15[\\ 5 \text{ m} & \text{for } t \in [15;35[\\ -7 \text{ m} & \text{for } t \in [35;50] \end{cases} \tag{3.32b}$$

$$z_d(t) = \begin{cases} 5 \text{ m} & \text{for } t \in [0;40[\\ 0 \text{ m} & \text{for } t \in [40;50[\end{cases} \tag{3.32c}$$

And the ψ_d is a Pulse input with amplitude $\frac{\pi}{8}$, and period 25 seconds and pulse width 50% of the period. To generate the planned trajectory, we used the block *Signal builder*.

Figure 3.26 shows the evolution of the quadrotor position along the three axes x, y and z by the Backstepping controller.

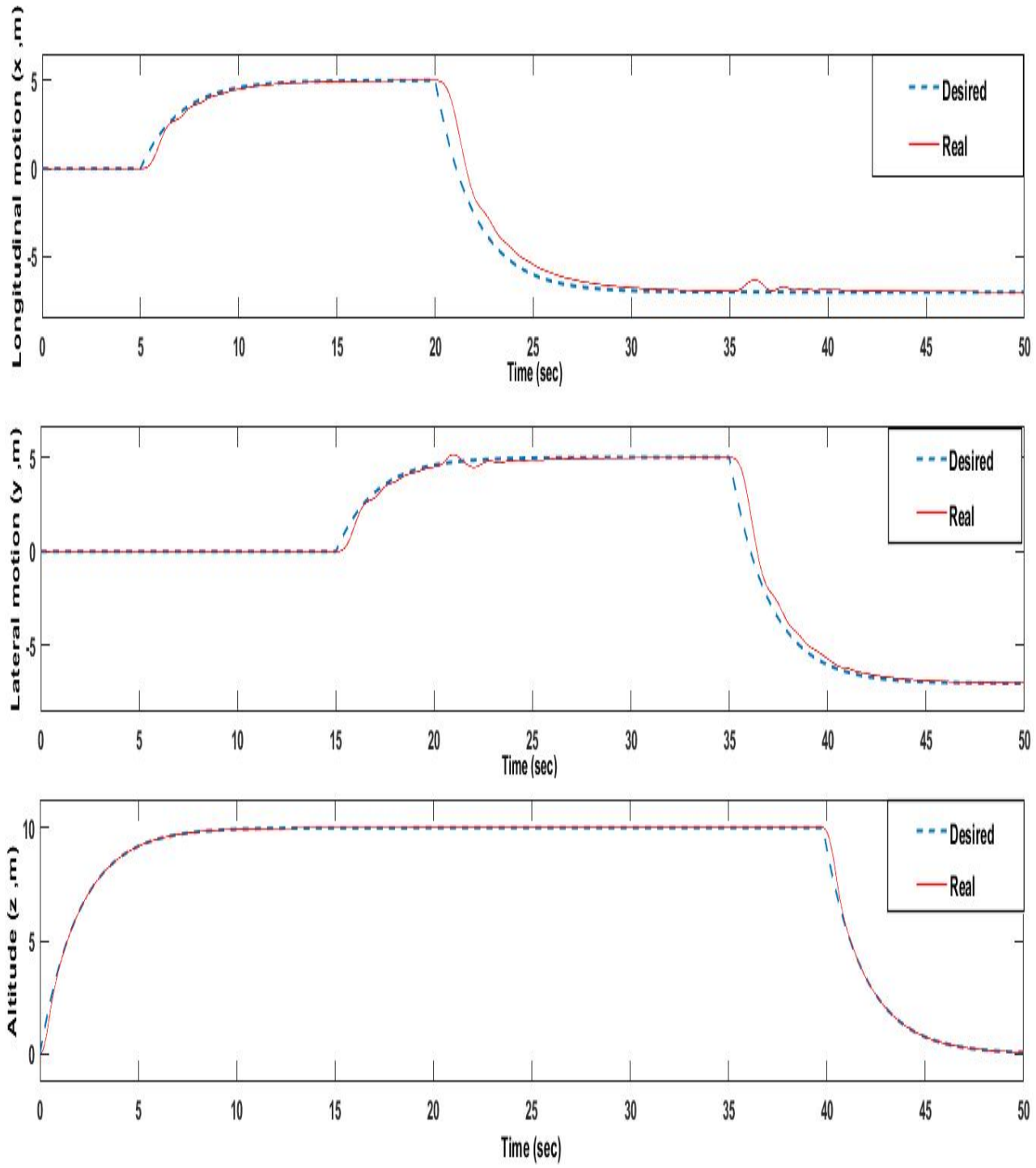


Figure 3.26: Generation of the desired trajectory by the Backstepping controller

Figure 3.27 illustrates the evolution of the attitude (roll, pitch and yaw angles) of the quadrotor during the realization of the desired trajectory.

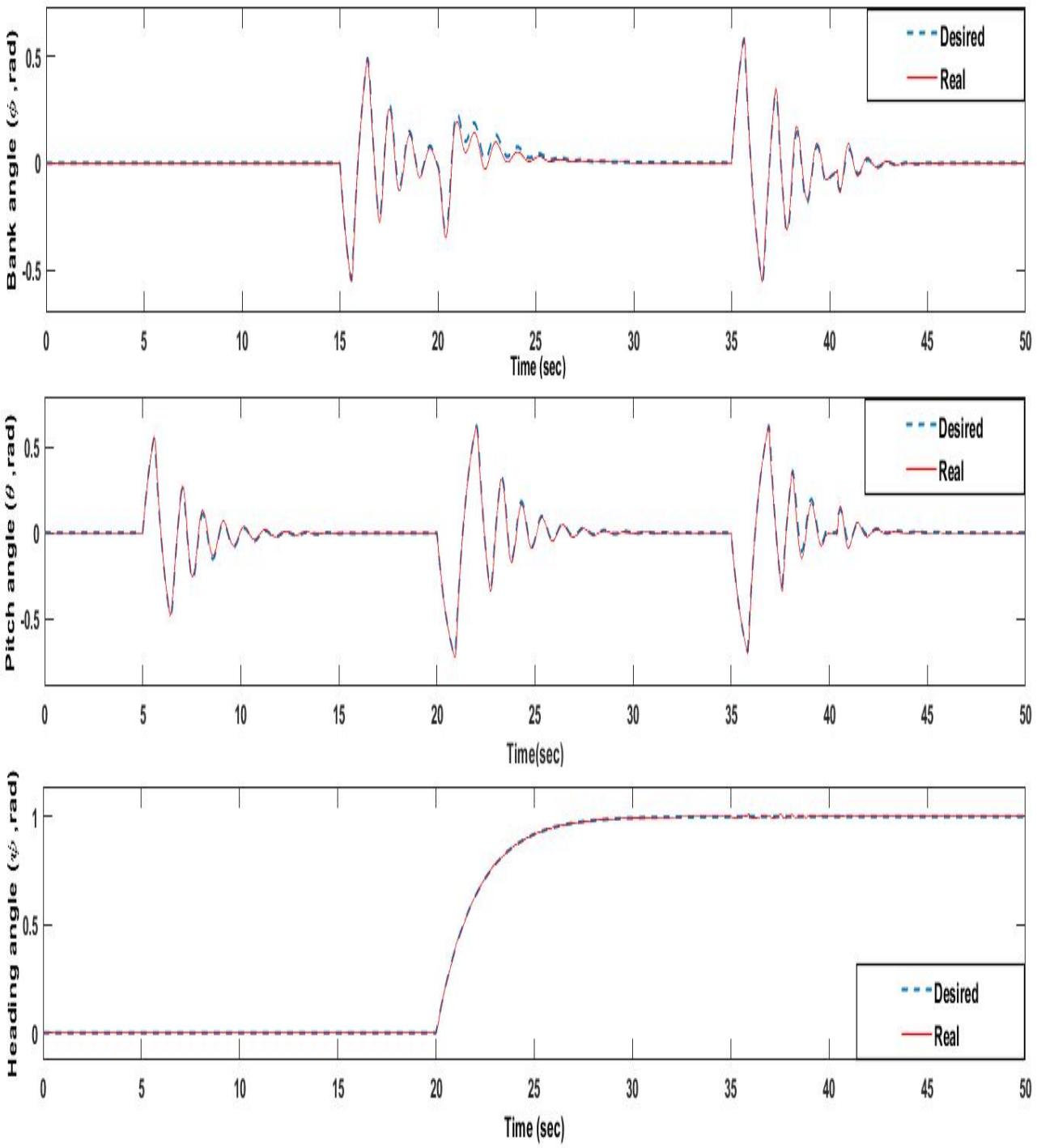


Figure 3.27: The attitude of the quadrotor achieving the desired trajectory

Figure 3.28 shows the result of the 3D visualization of the trajectory by the Backstepping controller.

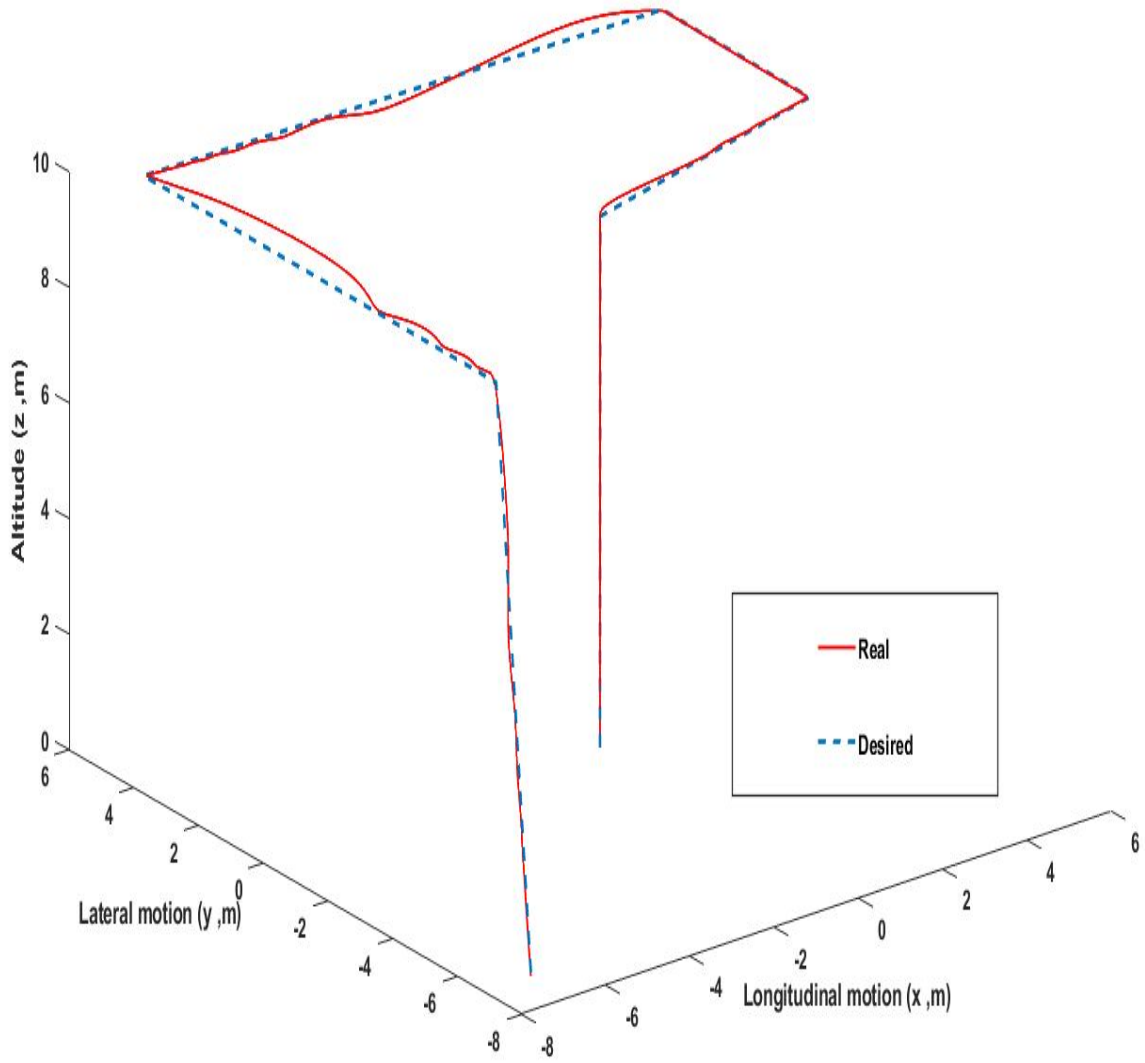


Figure 3.28: 3D visualization of the trajectory by the Backstepping controller

Now we present the control inputs of the Backstepping controller, as well as the angular speeds of the four rotors.

Figure 3.29 present the control inputs U1, U2, U3 and U4 of the Backstepping controller and Figure 3.30 present the rotational speeds of the four rotors during the execution of the desired trajectory.

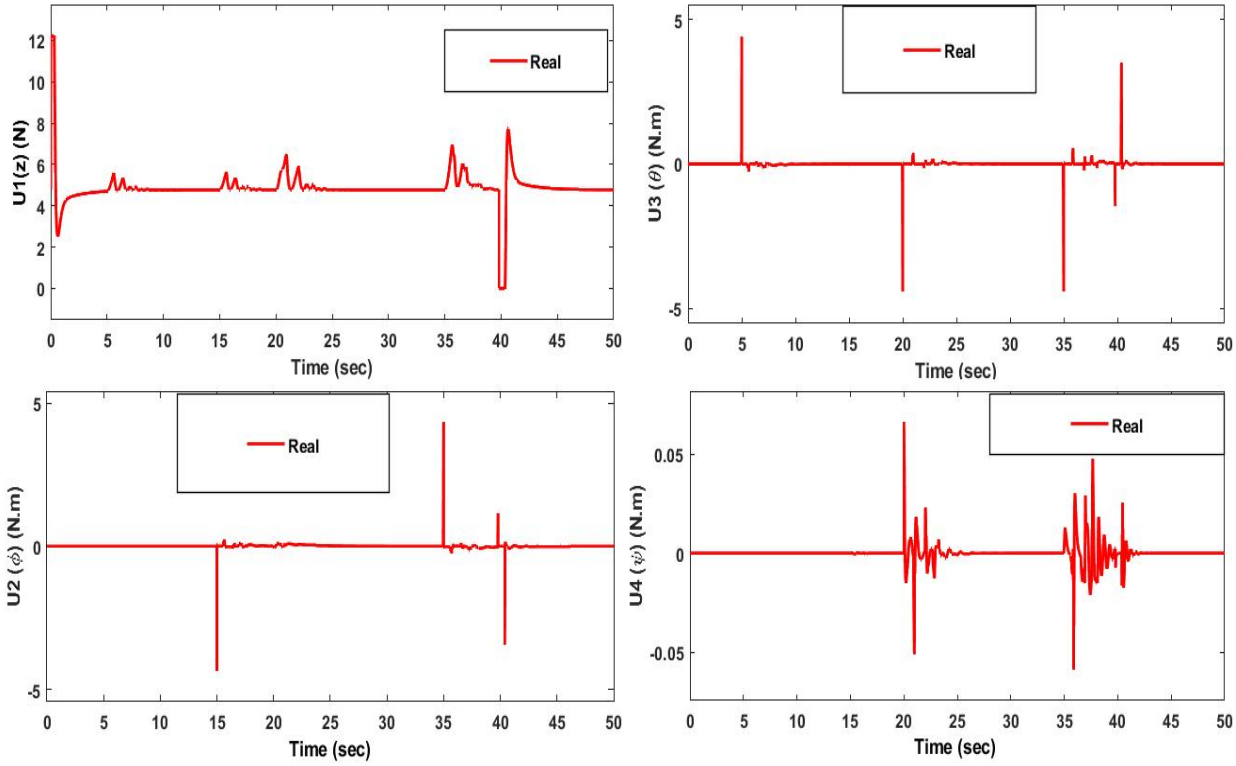


Figure 3.29: The control inputs of the Backstepping controller technique

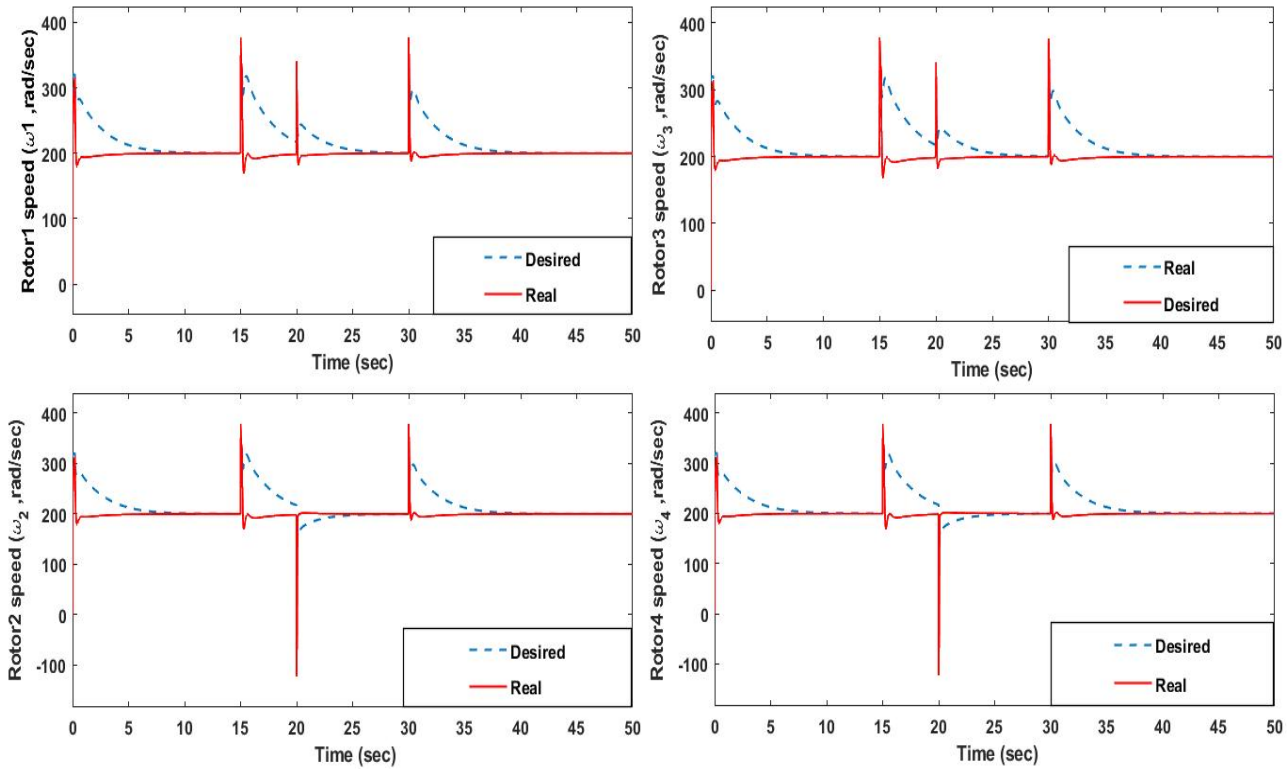


Figure 3.30: The rotational speeds of the four rotors

3.7.3 Simulation results

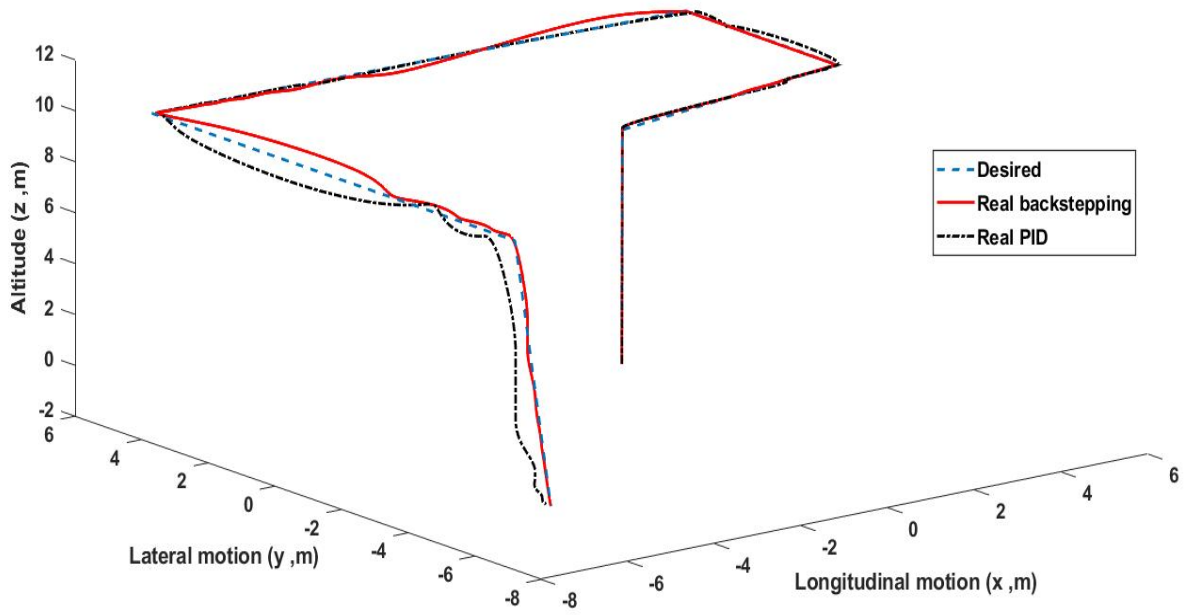


Figure 3.31: 3D visualization of the trajectory by the Backstepping and the PID controllers

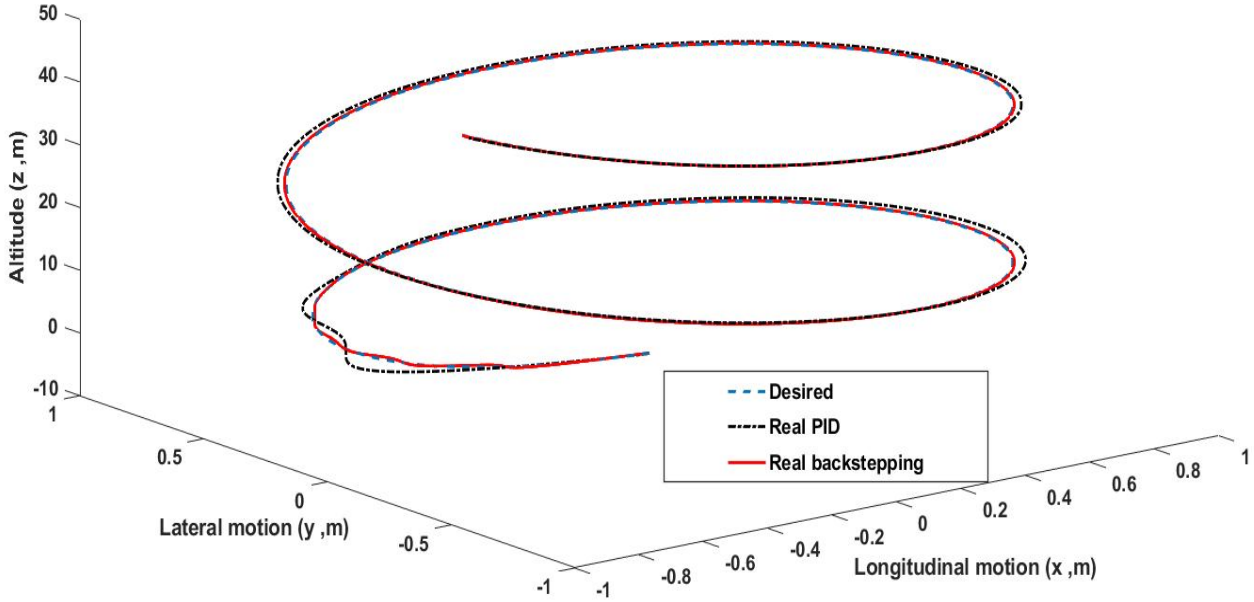


Figure 3.32: 3D helical trajectory by the Backstepping and PID controllers

For the bouth simulations of PID and Backstepping controllers and with a different scenarios, the Figures (3.8), (3.13), (3.21) and (3.26) depict the evolution of the quadrotor's position along the three axes (x, y, and z) using the Backstepping and PID controllers. According to these figures, it is evident that the quadrotor's position converges to the reference position with nearly zero steady-state errors, indicating excellent precision.

The Backstepping controller exhibits the best response times for position convergence along all three axes in this simulation.

Figures (3.9), (3.14), (3.22) and (3.27) illustrate the attitude evolution (roll, pitch, and yaw angles) of the quadrotor during the execution of the desired trajectory. From these figures, it is noticeable that the roll and pitch angles achieved by the drone are not the same, which can be attributed to the commands U_ϕ and U_θ.

In this case, there are no reference angles to evaluate the performance of the obtained angles. Convergence and return to zero for the roll and pitch angle curves are faster with PID control, but notable self-oscillations are observed, which explains the superior response time for this control.

Figures (3.12), (3.17), (3.25) and (3.30) present the rotational speeds of the four rotors during the execution of the desired trajectory. The angular speeds of the four rotors are equal during the hovering phases, at a speed of 200 rad/s.

Motors 2 and 4 contribute to a pitching moment and the first displacement along the x-axis. The same occurs at t = 12 sec.

During lateral movement (displacement along the y-axis at t = 7 sec and t = 17 sec), all four motors contribute to the rolling moment. The same occurs at t = 35 sec when the drone executes a yaw rotation.

The obtained results do not exactly correspond to expectations, which are influenced by the control allocation matrix, not being precisely calculated (pseudo-inverse). This explains the occurrence of motor speeds at inappropriate instants related to the movements associated with those motors.

The magnitude of the speed reflects the energy consumed by the corresponding control.

The faster the motors rotate, the higher the consumption.

Furthermore, the Euler angles, including roll, pitch, and yaw, approached zero, indicating the stabilization of the quadcopter's orientation. The nullification of Euler angles is a crucial requirement for maintaining Stable flight. The control law effectively tackled the complex dynamics and uncertainties associated with the quadcopter system, resulting in minimized oscillations and rapid convergence of the Euler angles to zero.

The achieved stability in both altitude and Euler angles indicates the successful design and implementation of the backstepping and the PID controllers. The simulation results demonstrate theThe simulation results demonstrate the robustness and efficiency of the control laws in ensuring a Stable flight, thereby enhancing the quadcopter's overall performance and maneuverability.

3.8 Conclusion

In this chapter, we designed the control laws using two methods: PID and Backstepping to ensure the stability and trajectory of the four motors. Also, we applied these control laws to all-wheel drives. In the simulation, both PID and Backstepping had to follow the desired paths. The results confirm the effectiveness of Backstepping and PID control laws in reference tracking. Moreover, PID is a linear command which means there are non-linear elements in the command. This leads to a less stable flight, which is why we added a backstepping command to eliminate the nonlinear elements for a more stable flight. Second, we fit the law of background control in addressing the complex dynamics and uncertainties inherent in quadcopter systems for stable flight. In the next chapter, we will see the contribution of an adaptive flight control strategy on modeling and attitude stabilization.

Chapter 4

Adaptive Flight Control

4.1 Adaptive Flight Control

Adaptive backstepping is a control design methodology that is used to design controllers for nonlinear systems. It is a recursive design procedure that uses Lyapunov functions to ensure the stability of the closed-loop system. The backstepping approach provides a recursive method for stabilizing the origin of a system in strict-feedback form. The adaptive backstepping approach has been an effective tool in the control of uncertain linear systems. However, its traditional recursive non-linear design is complicated, especially for a plant whose relative degree is high. The main idea behind backstepping control is to transform the system into a canonical form that facilitates the design process of the feedback controller. The main advantage of adaptive control over backstepping control is its adaptability to changes in the system's dynamics, making it suitable for applications where the system's dynamics vary over time.

Adaptive sliding mode for rotor

The model of the rotor in the following form [17]:

$$\dot{\omega} = bV + \beta_0 + \beta_1\omega + \beta_2\omega^2 \tag{4.1}$$

The control law:

$$V = \frac{1}{b}(\dot{\omega} - \beta_0 - \beta_1\omega - \beta_2\omega^2) \tag{4.2}$$

The new control law is:

$$V = h\dot{\omega} + \lambda_1 - \lambda_2\omega + \lambda_3\omega^2 \tag{4.3}$$

where: $h = \frac{1}{b}$, $\lambda_1 = -\frac{\beta_0}{b}$, $\lambda_2 = -\frac{\beta_1}{b}$, $\lambda_3 = -\frac{\beta_2}{b}$

To synthesize the adaptive control law V, it is assumed that the full state vector components are available through the measure and that the sign of h is known. We assume also that the parameters h and λ_i are unknown constants.

Now, let us choose a sliding surface σ_1 such as:

$$\sigma_1 = w - v_1 \tag{4.4a}$$

$$v_1 = w_d - ke \tag{4.4b}$$

$$e = w_d - w \tag{4.4c}$$

where e represent the error between the desired angular velocity and the real angular velocity, w_d is the desired angular velocity and k denotes a real positive parameter. Now a candidate Lyapunov positive definite function $V_1(\sigma)$ is defined:

$$V_1 = \frac{1}{2}\sigma_1^2 \tag{4.5a}$$

$$\dot{V_1} = \sigma_1 \left[\frac{1}{h}(V - \lambda_1 - \lambda_2\omega - \lambda_3\omega^2) - v_1\right] \tag{4.5b}$$

The non-linear dynamic of the rotor is written in the following way [7]:

$$hy^{n)} + \sum_{i=1}^{n} a_i f_i(x,t) = u \tag{4.6}$$

- **Sliding mode control**

Consider the control law :

$$u = hy_r^{(n)} - ks + \sum_{i=1}^{n} a_i f_i(x,t) \tag{4.7}$$

For our adaptive control, the control law (4.7) is replaced by :

$$u = \hat{h}y_r^{(n)} - ks + \sum_{i=1}^{n} \hat{a}_i f_i(x,t) \tag{4.8}$$

Slotine suggests choosing the following adaptation law:

$$\dot{\hat{h}} = -\gamma sign(h) s y_r^{(n)} \tag{4.9a}$$

$$\dot{\hat{a}}_i = -\gamma sign(h) s f_i \tag{4.9b}$$

By replacing in (4.5b):

$$u = \hat{h}\left[\ddot{\omega}_d - k\dot{e} - k\sigma\right] + \hat{\lambda}_1 + \hat{\lambda}_2\omega + \hat{\lambda}_3\omega^2 \tag{4.10}$$

The following figures are the simulation of the rotor with adaptive sliding mode control:

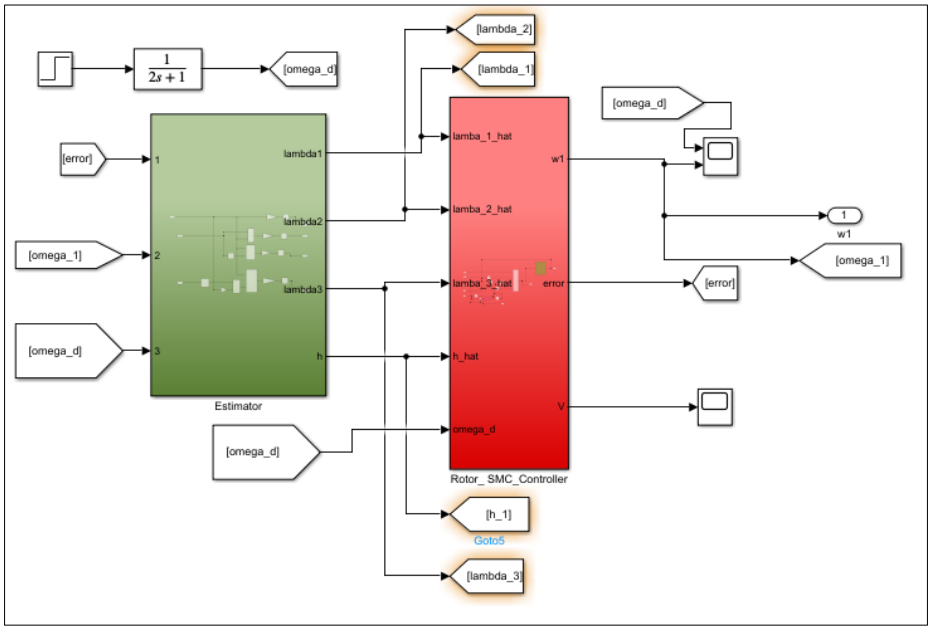


Figure 4.1: Rotor with adaptive control

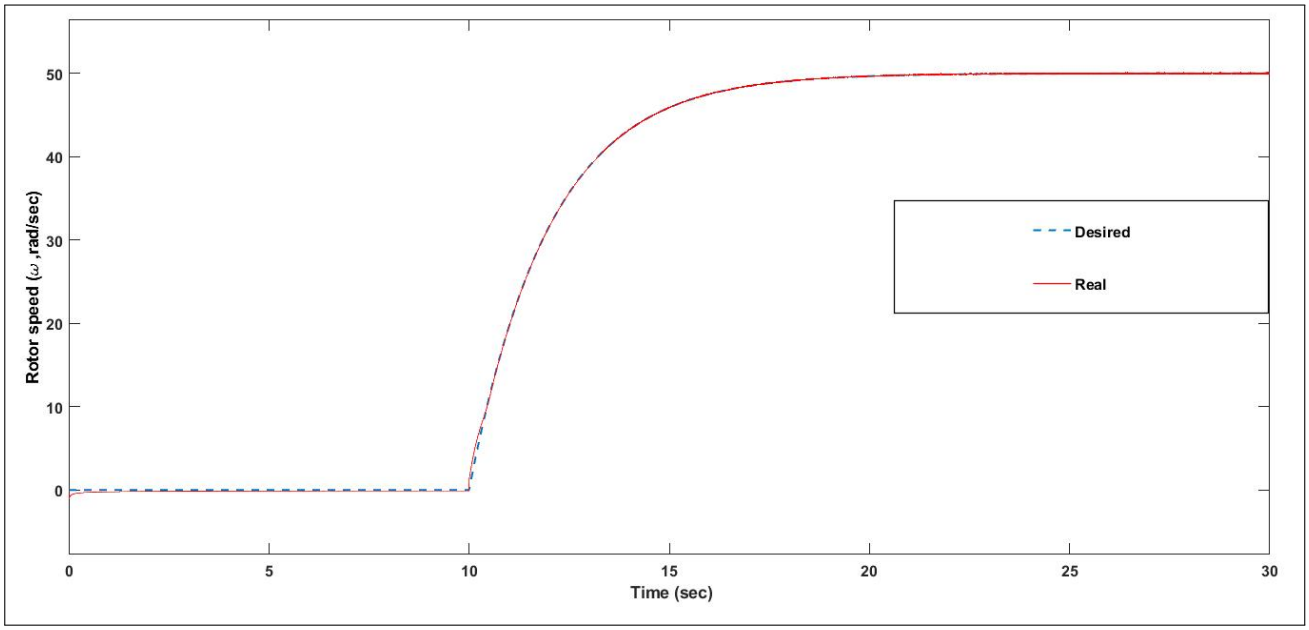


Figure 4.2: Rotor speed tracking

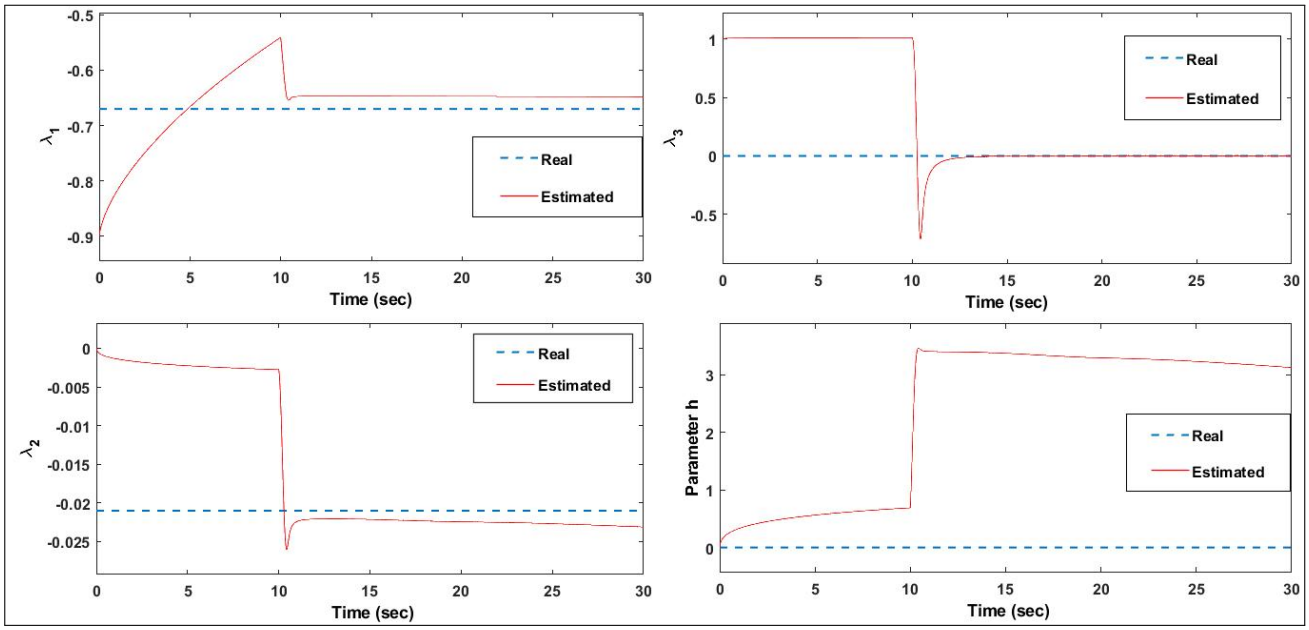


Figure 4.3: Parameters estimation

Analysis of Simulation Results:

Upon synthesizing adaptive sliding mode control based on Slotine's book and applying the control laws to the rotor system, we conducted simulations to evaluate the performance of the control strategy. The primary objective was to track the reference velocity while addressing the nonlinearities and uncertainties in the system.

The simulation results revealed satisfactory tracking of the reference velocity, indicating the effectiveness of the adaptive sliding mode control in achieving the desired tracking objective. However, an issue arose in the steady state, resulting in shattering behavior. This can be attributed to the discontinuity of the sign function employed in the control law. The shattering phenomenon can be problematic as it leads to oscillations and instability in the system.

Furthermore, we examined the estimation results of the system's parameters, specifically the convergence of the three parameters (λ_i) to their real values. The estimation algorithm demonstrated successful convergence. However, one parameter, denoted as 'h', did not converge to its real value. This discrepancy can be attributed to the adaptation gain used in the control strategy. Adjusting the adaptation gain using optimization techniques could potentially improve the convergence of 'h' to its true value.

4.1.1 Description of design steps for the quadrotor

To synthesize adaptive control laws, it is assumed that some uncertainty remains concerning the main coefficients w.r.t the aerodynamic moments, inertia matrix and mass of the system. It is also assumed that the full-state vector components are available through the measure and that the sign of b1 is known.

Now, let us consider roll dynamics expressed from (3.14a) and (3.14b) such as:

$$\dot{x}_1 = x_2 \tag{4.11a}$$

$$\dot{x}_2 = a_1 x_4 x_6 + a_2 x_2^2 + a_3 x_4 \bar{\omega} + b_1 U_2 \tag{4.11b}$$

Where a_1, a_2, a_3 and b_1 are considered unknown parameters.

Let z_1 be a new coordinate and ξ_ϕ the tracking error related to roll angle such as:

$$z_1 = \xi_\phi = \phi - \phi_d \tag{4.12}$$

- **First step**

Let $V_1(z_1)$ be a candidate positive definite Lyapunov function:

$$V_1(z_1) = \frac{1}{2} z_1^2 \tag{4.13}$$

Whose time-derivative $\dot{V}_1(z_1)$ is rendered negative definite:

$$\dot{V}_1(z_1) = -c_1 z_1^2 \qquad c_1 > 0 \tag{4.14}$$

By a stabilizing function x_{2_d} which is the desired value of x_2 defined as a virtual control as follows:

$$x_{2_d} = \dot{\phi}_d - c_1 z_1 \tag{4.15}$$

Since x_2 is just a state variable, let z_2 be the corresponding error variable:

$$z_2 = x_2 + c_1 z_1 - \dot{\phi}_d \tag{4.16}$$

And $\dot{z}_2$ its time derivative which can be expressed in the following form:

$$h_1 \dot{z}_2 = \lambda_1 x_4 x_6 + \lambda_2 x_2^2 + U_\phi - h_1 \left[\ddot{\phi}_d - c_1 \left(z_2 - c_1 z_1 \right) \right] \tag{4.17}$$

With :

$$h_1 = \frac{1}{b_1} \quad \lambda_1 = \frac{a_1}{b_1} \quad \lambda_2 = \frac{a_2}{b_1} \quad \lambda_3 = \frac{a_3}{b_1}$$

- **Second step**

To synthesize the control law U_ϕ, let us try to achieve this objective by augmenting the already existing Lyapunov function $V_1(z_1)$ with a quadratic term in the error variable z_2. Consequently, the augmented Lyapunov function $V_2(z_1, z_2)$ is introduced as follows:

$$V_2(z_1, z_2) = \frac{1}{2}z_1^2 + \frac{|h_1|}{2}z_2^2 \tag{4.18}$$

Its time derivative is rendered negative definite:

$$\dot{V}_2(z_1, z_2) = -c_1 z_1^2 - |c_2| z_2^2 \leq 0 \quad , h_1 c_2 > 0 \tag{4.19}$$

By the stabilizing control law U_ϕ chosen as follows:

$$U_\phi = -\operatorname{sgn}(h_1)\, z_1 - c_2 z_2 - \lambda_1 x_4 x_6 - \lambda_2 x_2^2 + h_1 \left[\ddot{\phi}_d - c_1 (z_2 - c_1 z_1)\right] \tag{4.20}$$

Of course, the control law (3.38) can not be implemented, since λ_i et h_1 are unknown parameters. Instead, one can employ the certainty equivalence principle in which unknown parameters are replaced by their estimates $\hat{\lambda}_i$ and $\hat{h}_1$. It results:

$$U_\phi = -\operatorname{sgn}(h_1)\, z_1 - c_2 z_2 - \hat{\lambda}_1 x_4 x_6 - \hat{\lambda}_2 x_2^2 + \hat{h}_1 \left[\ddot{\phi}_d - c_1 (z_2 - c_1 z_1)\right] \tag{4.21}$$

And the time derivative of $V_2(z_1, z_2)$ becomes:

$$\dot{V}_2 = -c_1 z_1^2 - |c_2|\, z_2^2 + z_2 \operatorname{sgn}(h_1) \left[\tilde{\lambda}_1 x_4 x_6 + \tilde{\lambda}_2 x_2^2 + \tilde{\lambda}_3 \bar{\Omega} x_4 - \tilde{h}_1 \left(\ddot{\phi}_d - c_1 (z_2 - c_1 z_1)\right)\right] \tag{4.22}$$

Where $\tilde{\lambda}_i$ and $\tilde{h}_1$ denote the estimation errors:

$$\tilde{\lambda}_i = \lambda_i - \hat{\lambda}_i \tag{4.23a}$$

$$\tilde{h}_1 = h_1 - \hat{h}_1 \tag{4.23b}$$

Replacing the control law (3.39) in (3.35), we get the new closed-loop dynamics expressed as

follows:

$$\dot{z}_2 = -\frac{1}{|h_1|}z_1 - \frac{c_2}{h_1}z_2 + \frac{1}{h_1}\left[\tilde{\lambda}_1 x_4 x_6 + \tilde{\lambda}_2 x_2^2 + \tilde{\lambda}_3 \bar{\Omega} x_4 - \tilde{h}_1\left(\ddot{\phi}_d - c_1\left(z_2 - c_1 z_1\right)\right)\right] \tag{4.24}$$

This yields:

$$\dot{z} = \mathbb{A}z + \frac{1}{h_1}\tilde{\lambda}_\phi^T W(\underline{x}, \bar{\Omega}) \tag{4.25}$$

With:

$$\tilde{\lambda}_\phi^T = \left[\tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\lambda}_3, \tilde{h}_1\right] \tag{4.26a}$$

$$W(\underline{x}, \bar{\Omega}) = \left[x_4 x_6, x_2^2, \bar{\Omega} x_4, \ddot{\phi}_d - c_1\left(z_2 - c_1 z_1\right)\right]^T \tag{4.26b}$$

$$\begin{pmatrix} \dot{z}_1 \\ \\ \dot{z}_2 \end{pmatrix} = \begin{pmatrix} -c_1 & 1 \\ \\ -\frac{1}{|h_1|} & -\frac{c_2}{h_1} \end{pmatrix} \begin{pmatrix} z_1 \\ \\ z_2 \end{pmatrix} = \mathbb{A}z \tag{4.26c}$$

- **Third step**

Since the third term of (3.40) is indefinite and contains the estimation errors $\tilde{\lambda}_i$ and $\tilde{h}_1$, we can not conclude anything about the stability of (3.36). Therefore, we augment $V_2(z_1, z_2)$ with a quadratic terms in the parameters errors $\tilde{\lambda}_i$ and $\tilde{h}_1$ to obtain the following Lyapunov function $V_3(z_1, z_2, \tilde{\lambda}_\phi)$:

$$V_3(z_1, z_2, \tilde{\lambda}_\phi) = \frac{1}{2}z_1^2 + \frac{|h_1|}{2}z_2^2 + \frac{1}{2\gamma_\phi}\tilde{\lambda}_\phi^T\tilde{\lambda}_\phi \tag{4.27}$$

Where $\gamma_\phi > 0$ is the adaptation gain according to the roll dynamics. The time derivative of $V_3(z_1, z_2, \tilde{\lambda}_\phi)$ is as follows:

$$\begin{aligned}\dot{V}_3 &= z_1\dot{z}_1 + |h_1|\, z_2\dot{z}_2 - \frac{1}{\gamma_\phi}\left[\tilde{h}_1\dot{\hat{h}}_1 + \sum_{i=1}^{3}\tilde{\lambda}_i\dot{\hat{\lambda}}_i\right] \\ &= -c_1 z_1^2 - |c_2|\, z_2^2 + \tilde{\lambda}_1\left[z_2\,\mathrm{sgn}\,(h_1)\, x_4 x_6 - \frac{\dot{\hat{\lambda}}_1}{\gamma_\phi}\right] \\ &\quad + \tilde{\lambda}_2\left[z_2\,\mathrm{sgn}\,(h_1)\, x_2^2 - \frac{\dot{\hat{\lambda}}_2}{\gamma_\phi}\right] \\ &\quad - \tilde{h}_1\left[z_2\,\mathrm{sgn}\,(h_1)\left(\ddot{\phi}_d - c_1\left(z_2 - c_1 z_1\right)\right) + \frac{\dot{\hat{h}}_1}{\gamma_\phi}\right]\end{aligned} \tag{4.28}$$

The terms of (3.46) containing $\tilde{\lambda}_\phi$ components are still indefinite. However, the situation is much better than in (3.40). In fact, the dynamics of estimation errors vector $\dot{\tilde{\lambda}}_\phi = -\dot{\hat{\lambda}}_\phi$ appears explicitly. Note that, with the appropriate choice of $\dot{\hat{\lambda}}_\phi$, the indefinite term can be canceled. Thus, we choose the adaptation laws as follows:

$$\dot{\hat{\lambda}}_\phi = \gamma_\phi\,\mathrm{sgn}\,(h_1)\begin{pmatrix} 0 & x_4 x_6 \\ 0 & x_2^2 \\ 0 & -\left[\ddot{\phi}_d - c_1\left(z_2 - c_1 z_1\right)\right] \end{pmatrix}\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \tag{4.29}$$

Which yields:

$$\dot{V}_3\left(z_1, z_2, \tilde{\lambda}_\phi\right) = -c_1 z_1^2 - |c_2|\, z_2^2 \leq 0 \tag{4.30}$$

This implies that $z = 0$, $\tilde{\lambda}_\phi = 0$ equilibrium point of the closed-loop adaptive system consisting of (3.43) and (3.46) is globally stable. Adaptive flight control and adaptation laws related to pitch, heading and altitude dynamics are obtained by following the same steps

presented above, we get:

$$U_\theta = -\operatorname{sgn}(h_2)\, z_3 - c_4 z_4 - \hat{\lambda}_4 x_2 x_6 - \hat{\lambda}_5 x_4^2 + \hat{h}_2 \left[\ddot{\theta}_d - c_3 (z_4 - c_3 z_3)\right]; c_3, h_2 c_4 > 0 \tag{4.31a}$$

$$U_\psi = -\operatorname{sgn}(h_3)\, z_5 - c_6 z_6 - \hat{\lambda}_7 x_2 x_4 - \hat{\lambda}_8 x_6^2 + \hat{h}_3 \left[\ddot{\psi}_d - c_5 (z_6 - c_5 z_5)\right]; \quad c_5, h_3 c_6 > 0 \tag{4.31b}$$

$$U_z = \frac{1}{\cos x_1 \cos x_3} \left[-\hat{\lambda}_{11} x_{12} - c_{12} z_{12} - z_{11} + \hat{m} (\ddot{z}_d + g - c_{11} (z_{12} - c_{11} z_{11}))\right]; \quad c_{11}, c_{12} > 0 \tag{4.31c}$$

With:

$$\dot{\hat{\lambda}}_\theta = \gamma_\theta \operatorname{sgn}(h_2) \begin{pmatrix} 0 & x_2 x_6 \\ 0 & x_4^2 \\ 0 & -\left[\ddot{\theta}_d - c_3 (z_4 - c_3 z_3)\right] \end{pmatrix} \begin{pmatrix} z_3 \\ z_4 \end{pmatrix} \tag{4.32a}$$

$$\dot{\hat{\lambda}}_\psi = \gamma_\psi \operatorname{sgn}(h_3) \begin{pmatrix} 0 & x_2 x_4 \\ 0 & x_6^2 \\ 0 & -\left[\ddot{\psi}_d - c_5 (z_6 - c_5 z_5)\right] \end{pmatrix} \begin{pmatrix} z_5 \\ z_6 \end{pmatrix} \tag{4.32b}$$

$$\begin{pmatrix} \dot{\hat{\lambda}}_{11} \\ \dot{\hat{m}} \end{pmatrix} = \gamma_z \begin{pmatrix} 0 & x_{12} \\ 0 & -[\ddot{z}_d + g - c_{11} (z_{12} - c_{11} z_{11})] \end{pmatrix} \begin{pmatrix} z_{11} \\ z_{12} \end{pmatrix} \tag{4.32c}$$

And

$$z_3 = \xi_\theta = \theta - \theta_d \quad , \quad z_4 = x_4 - \dot{\theta}_d + c_3 z_3 \tag{4.33a}$$

$$z_5 = \xi_\psi = \psi - \psi_d \quad , \quad z_6 = x_6 - \dot{\psi}_d + c_5 z_5 \tag{4.33b}$$

$$z_{11} = \xi_z = z - z_d \quad , \quad z_{12} = x_{12} - \dot{z}_d + c_{11} z_{11} \tag{4.33c}$$

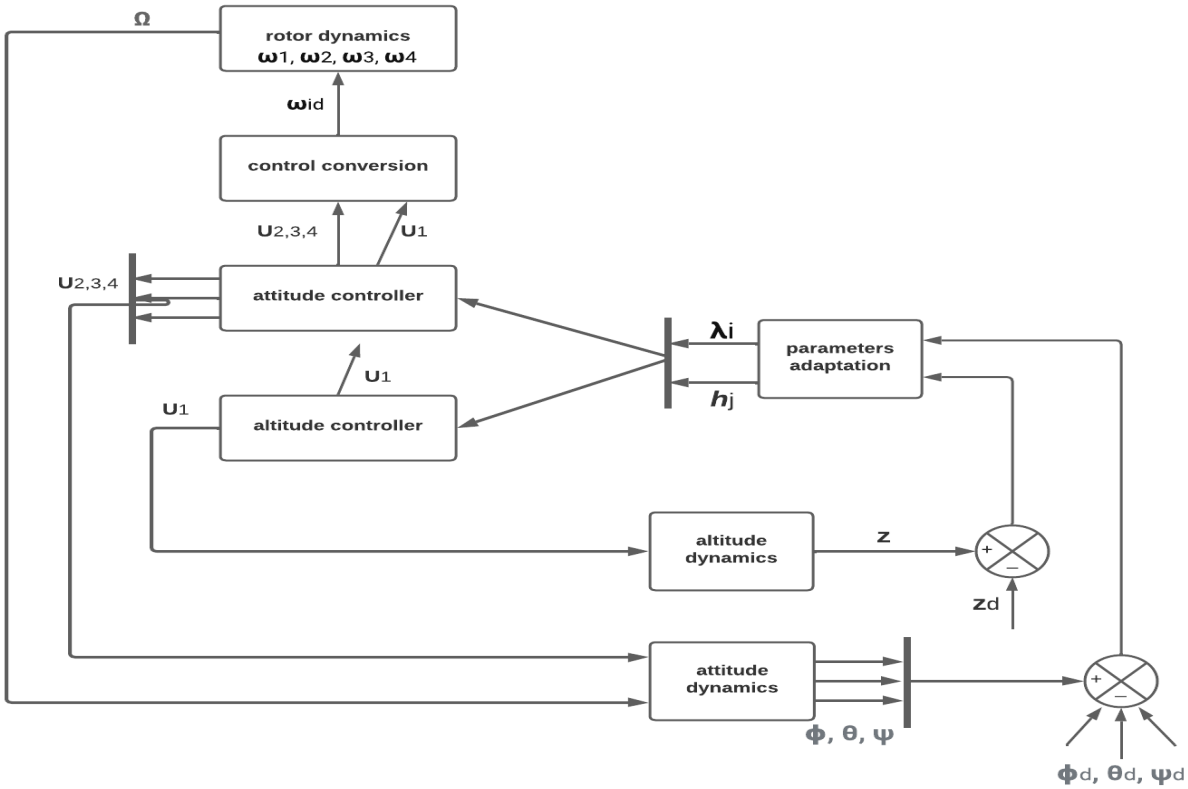


Figure 4.4: Block diagram of the Adaptive backstepping flight command applied to the quadrotor

4.2 Adaptive Backstepping Flight Control implementation

We will present the simulation results from the application of the Adaptive Backstepping Flight control technique seen previously on the quadrotor.

The simulation platform used is Simulink MATLAB R2021A. The simulation model of the quadrotor implemented in Simulink is complete without any simplification.

4.2.1 Implementation in the yaw angle

Suppose I_x is not equal to I_y, and the drag coefficient $Kfaz$ is unknown. In such a scenario, the previous control law is inadequate and fails to achieve the desired tracking angle yaw. To address this issue, an adaptive approach law needs to be implemented in the yaw control. The objective of this adaptive approach is to estimate the unknown parameters of the system and control law. By continuously updating and adapting the control parameters based on the estimated values, the adaptive approach law enables the system to better compensate for the unknowns, ultimately improving the accuracy and effectiveness of the yaw control.

The block diagram of the Adaptive Backstepping flight control simulation model in Matlab Simulink applied to the yaw control is shown in Figure 4.8. Adaptive flight control and adaptation law equations related to heading dynamic are given by:

$$U_\psi = -\operatorname{sgn}(h_3)\, z_5 - c_6 z_6 - \hat{\lambda}_7 x_2 x_4 - \hat{\lambda}_8 x_6^2 + \hat{h}_3 \left[\ddot{\psi}_d - c_5 (z_6 - c_5 z_5)\right]; \quad c_5, h_3 c_6 > 0 \tag{4.34a}$$

$$\dot{\hat{\lambda}}_\psi = \gamma_\psi \operatorname{sgn}(h_3) \begin{pmatrix} 0 & x_2 x_4 \\ 0 & x_6^2 \\ 0 & -\left[\ddot{\psi}_d - c_5 (z_6 - c_5 z_5)\right] \end{pmatrix} \begin{pmatrix} z_5 \\ z_6 \end{pmatrix} \tag{4.34b}$$

4.2.2 Trajectory tracking and parameters estimation

In the simulation of the Adaptive Backstepping flight control, we will simulate the altitude z and the angle ψ to make the heading of the quadrotor.

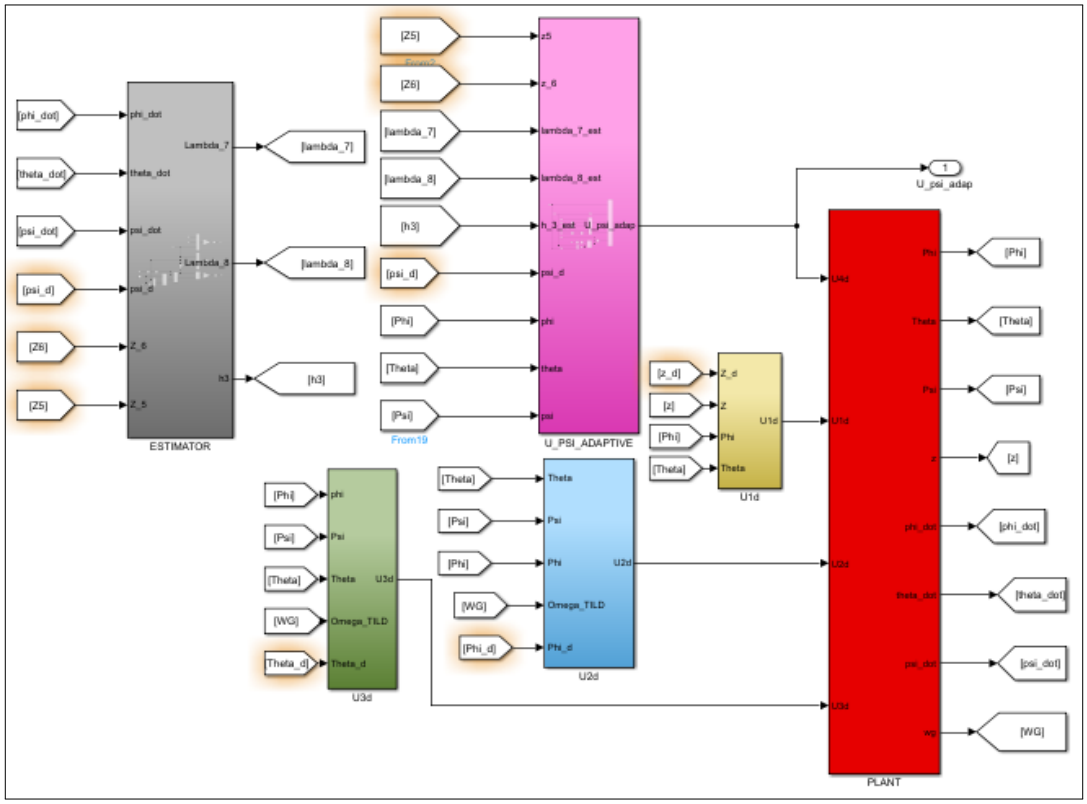


Figure 4.5: The block diagram of Adaptive Backstepping flight control applied to the yaw control

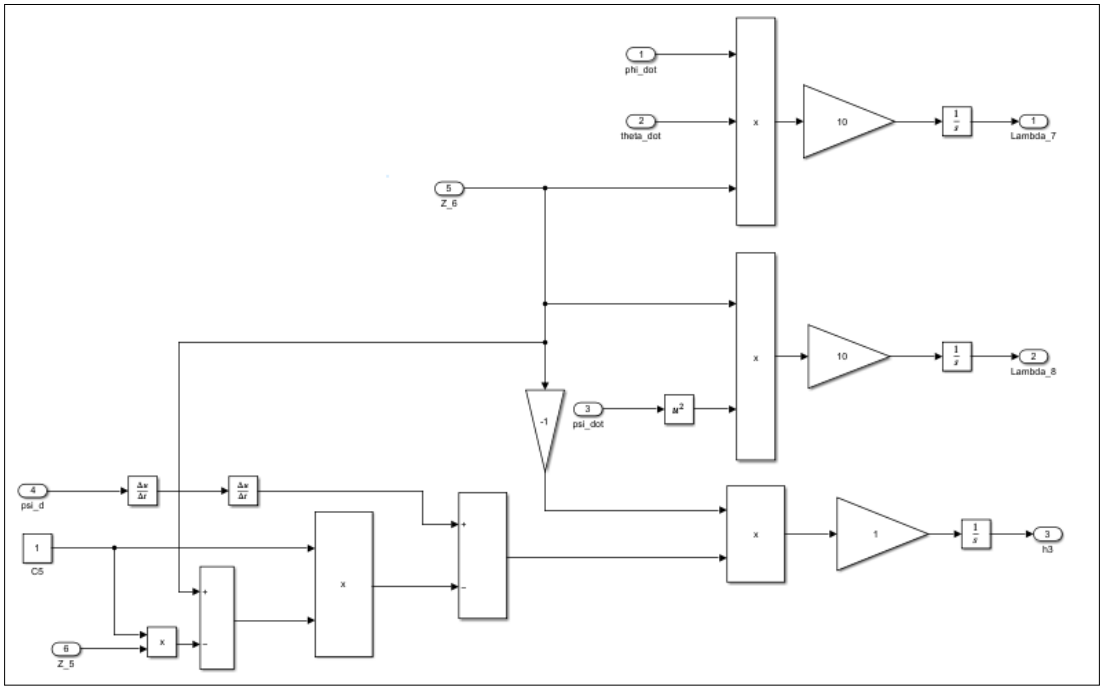


Figure 4.6: The estimator equation

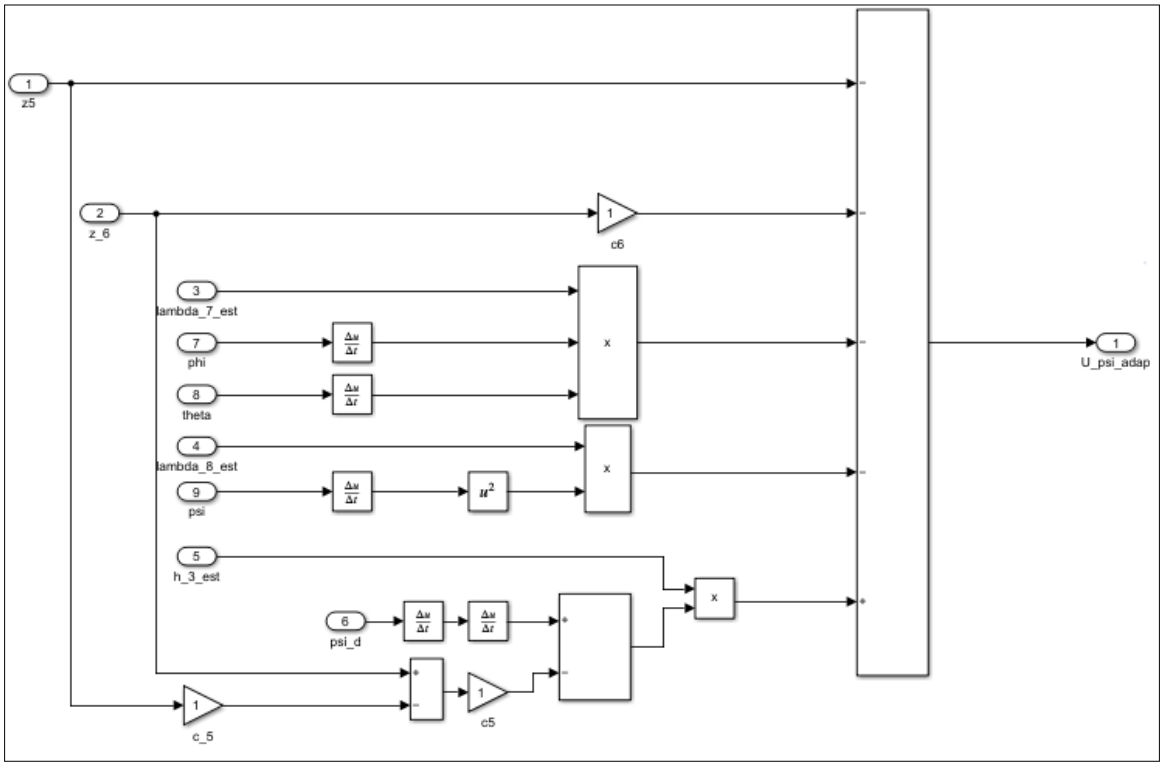


Figure 4.7: The U_4 command equation

We have set the duration of the mission to 50 seconds. The reference trajectory is given by the equations:

$$x_d(t) = \begin{cases} 0 \text{ m} & \text{for } t \in [0; 50[\end{cases} \tag{4.35a}$$

$$y_d(t) = \begin{cases} 0 \text{ m} & \text{for } t \in [0; 50[\end{cases} \tag{4.35b}$$

$$z_d(t) = \begin{cases} 0 \text{ m} & \text{for } t \in [0; 15[\\ 20 \text{ m} & \text{for } t \in [15; 35[\\ 5 \text{ m} & \text{for } t \in [35; 50[\end{cases} \tag{4.35c}$$

And the ψ_d is a sine wave input with amplitude $\frac{\pi}{4}$, and frequance 0.5 s^{-1}. To generate the planned trajectory, we used the block *Signal builder*.

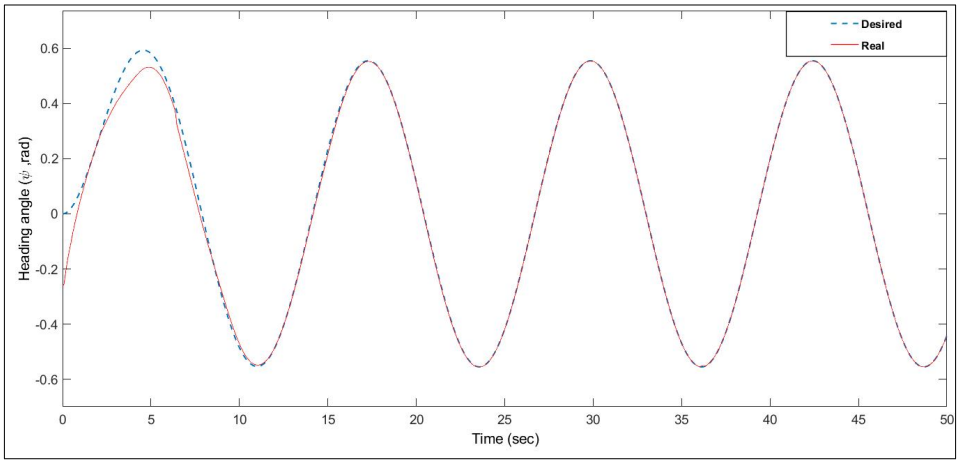


Figure 4.8: The heading angle ψ_d and ψ outputs

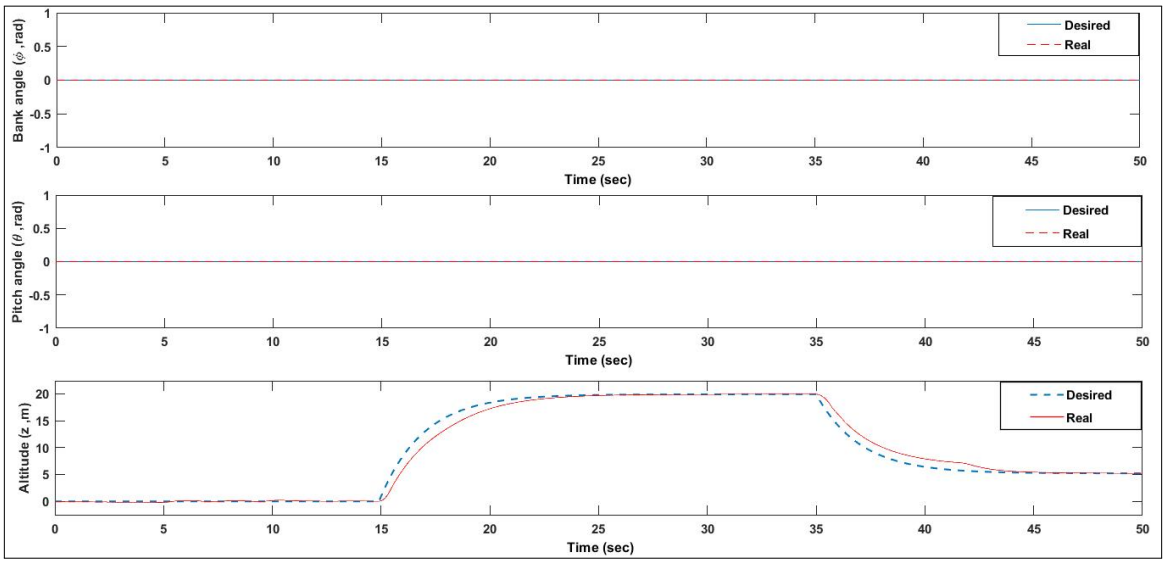


Figure 4.9: The ϕ, θ angles and altitude z outputs

4.2.3 Results

The figures above illustrate the estimation of system parameters and outputs, with a specific focus on achieving the desired output tracking rather than the convergence of parameter estimation. The primary objective is to determine the appropriate adaptation gain, represented by γ_i, for the yaw control law. Through iterative adjustments of the adaptation gain, the system

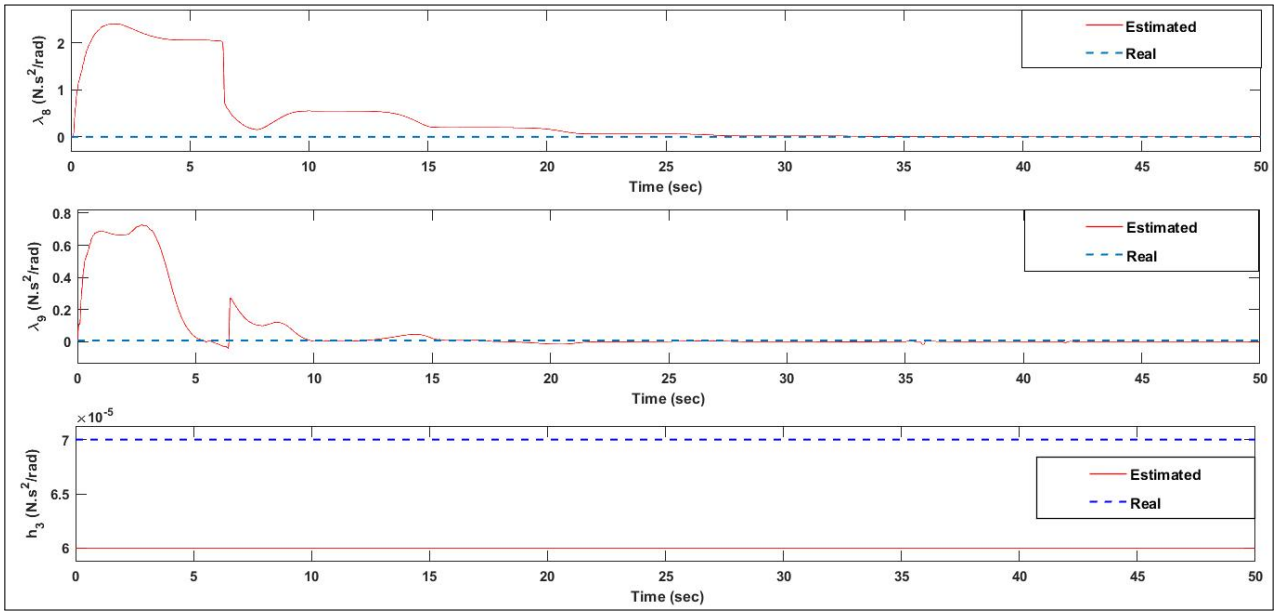


Figure 4.10: The λ_8, λ_9 and h_3 estimated outputs

aims to optimize the control law and improve the accuracy of the desired output tracking. Additionally, the figure demonstrates that the control law for ψ, which represents the heading angle, is deemed acceptable. This conclusion is drawn from observing the convergence of the error between the desired and actual heading angles, which steadily approaches zero. This outcome can be considered a positive result, indicating the effectiveness of the control law in achieving the desired tracking.

4.2.4 Implementation in the altitude

The drag coefficient $Kfdz$ is unknown. In such a scenario, the previous control law is inadequate and fails to achieve the desired tracking altitude. To address this issue, an adaptive approach law needs to be implemented in altitude control. The objective of this adaptive approach is to estimate the unknown parameters of the system and control law. By continuously updating and adapting the control parameters based on the estimated values, the adaptive approach law enables the system to better compensate for the unknowns, ultimately improving the accuracy and effectiveness of altitude control.

The block diagram of the Adaptive Backstepping flight control simulation model in Matlab Simulink applied to altitude control is shown in Figure 4.2. Adaptive flight control and

adaptation law equations related to altitude dynamic are given by:

$$U_z = \frac{1}{\cos x_1 \cos x_3}\left[-\hat{\lambda}_{11}x_{12} - c_{12}z_{12} - z_{11} + \hat{m}\left(\ddot{z}_d + g - c_{11}\left(z_{12} - c_{11}z_{11}\right)\right)\right]; \quad c_{11}, c_{12} > 0 \tag{4.36a}$$

$$\begin{pmatrix} \dot{\hat{\lambda}}_{11} \\ \dot{\hat{m}} \end{pmatrix} = \gamma_z \begin{pmatrix} 0 & x_{12} \\ 0 & -\left[\ddot{z}_d + g - c_{11}\left(z_{12} - c_{11}z_{11}\right)\right] \end{pmatrix} \begin{pmatrix} z_{11} \\ z_{12} \end{pmatrix} \tag{4.36b}$$

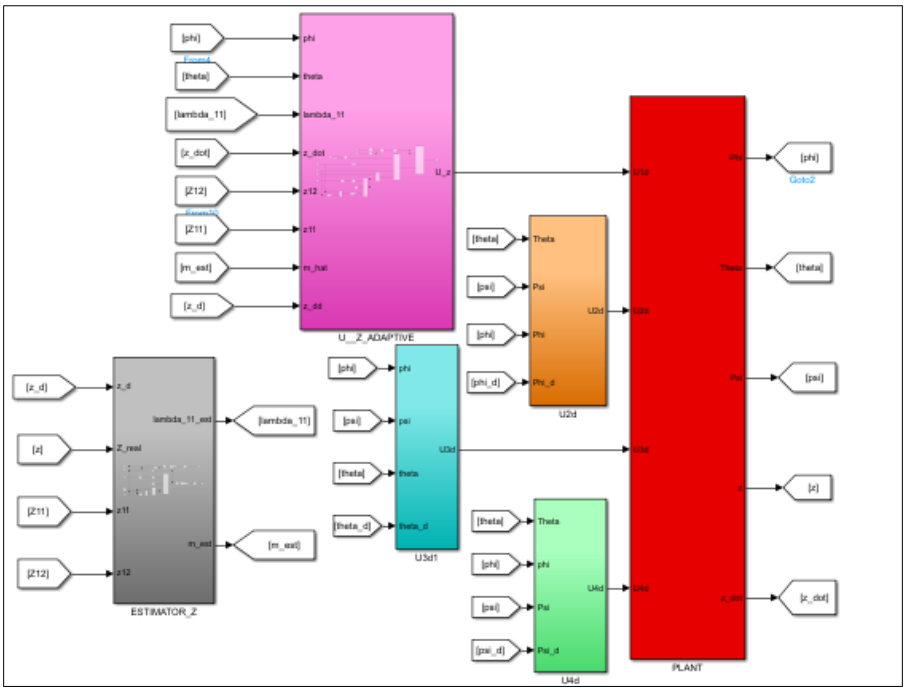


Figure 4.11: The block diagram of Adaptive Backstepping flight control applied to altitude control

4.2.5 Trajectory tracking and parameters estimation

In the simulation of the Adaptive Backstepping flight control, we will simulate the altitude z and the angles ϕ,θ and ψ to make the stabilization of the quadrotor.

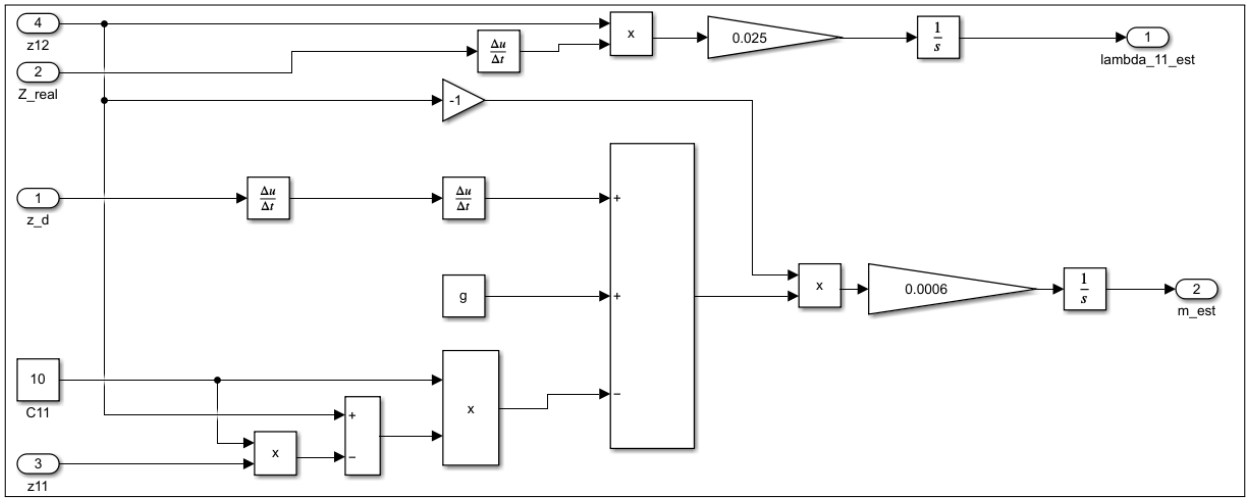


Figure 4.12: The estimator equation

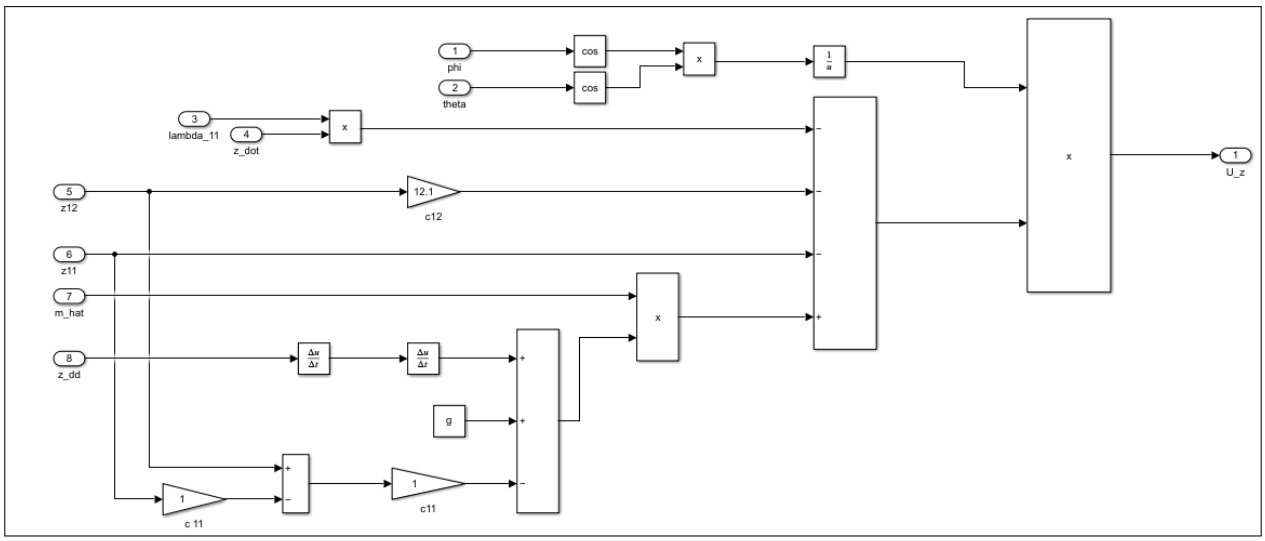


Figure 4.13: The U_1 command equation

We have set the duration of the mission to 50 seconds. The reference trajectory is given by the equations:

$$\phi_d(t) = \begin{cases} 0 \text{ rad} & \text{for } t \in [0; 50[\end{cases} \tag{4.37a}$$

$$\theta_d(t) = \begin{cases} 0 \text{ rad} & \text{for } t \in [0; 50[\end{cases} \tag{4.37b}$$

$$\psi_d(t) = \begin{cases} 0 \text{ rad} & \text{for } t \in [0; 50[\end{cases} \tag{4.37c}$$

And the z_d is a step input with a final value of 20 m, and step time 20 sec. And ϕ_d, θ_d and ψ_d have initial values $\frac{\pi}{3}$, $\frac{\pi}{6}$ and 1 respectively, to ensure the stability of hovering.

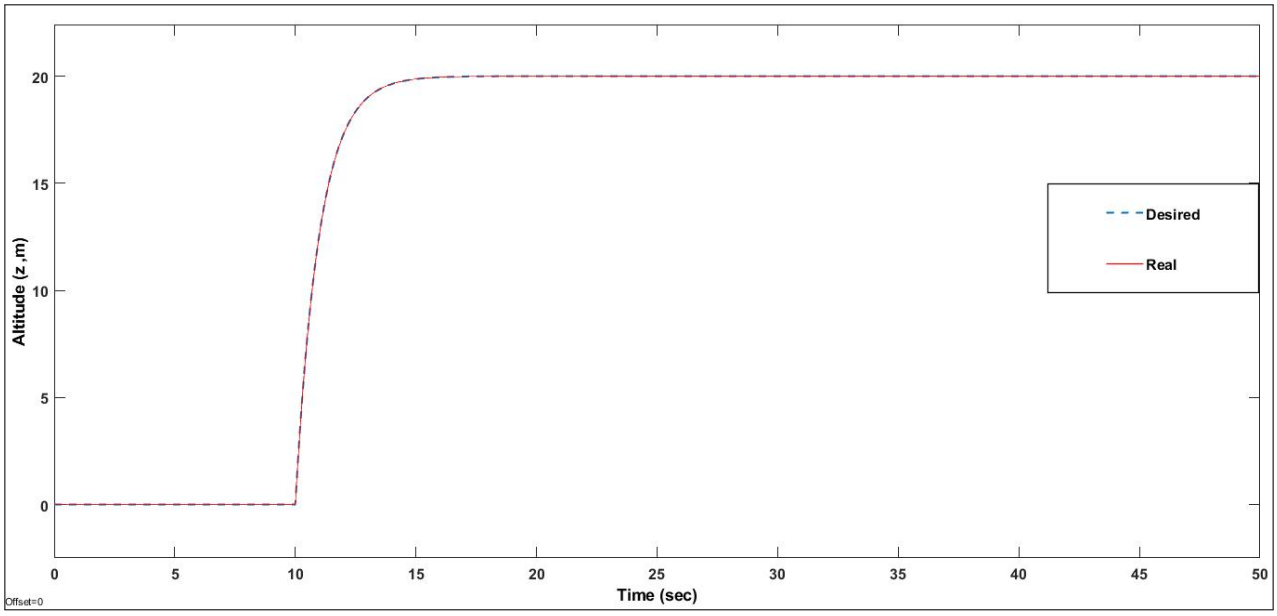


Figure 4.14: The altitude z_d and z outputs

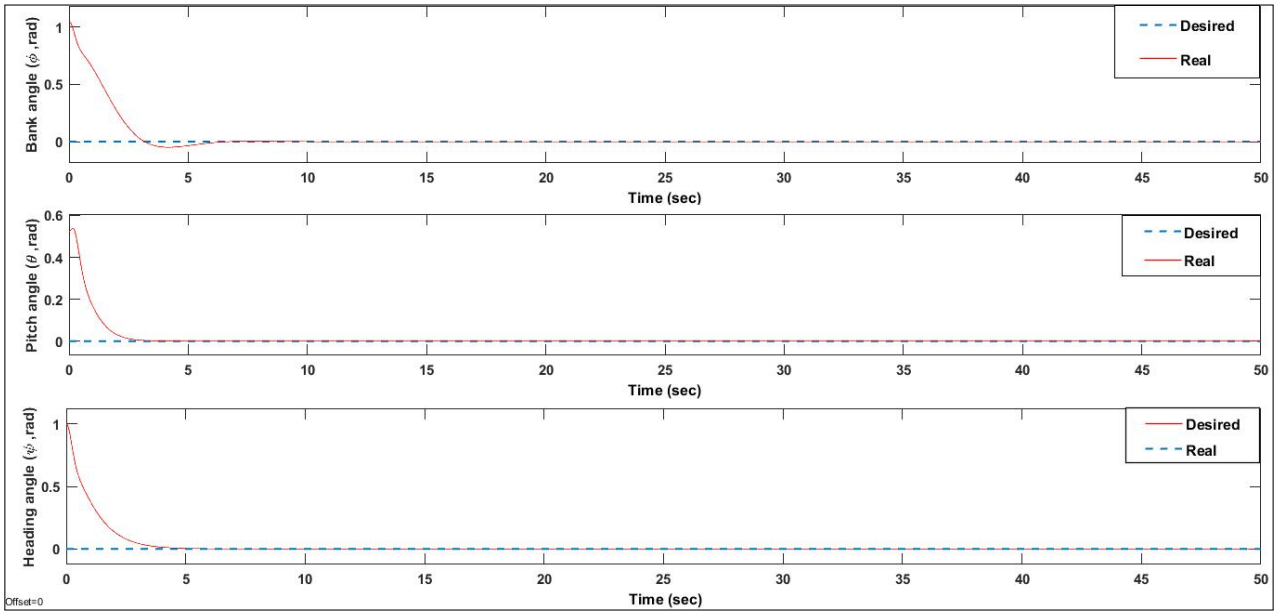


Figure 4.15: The ϕ, θ and ψ angles outputs

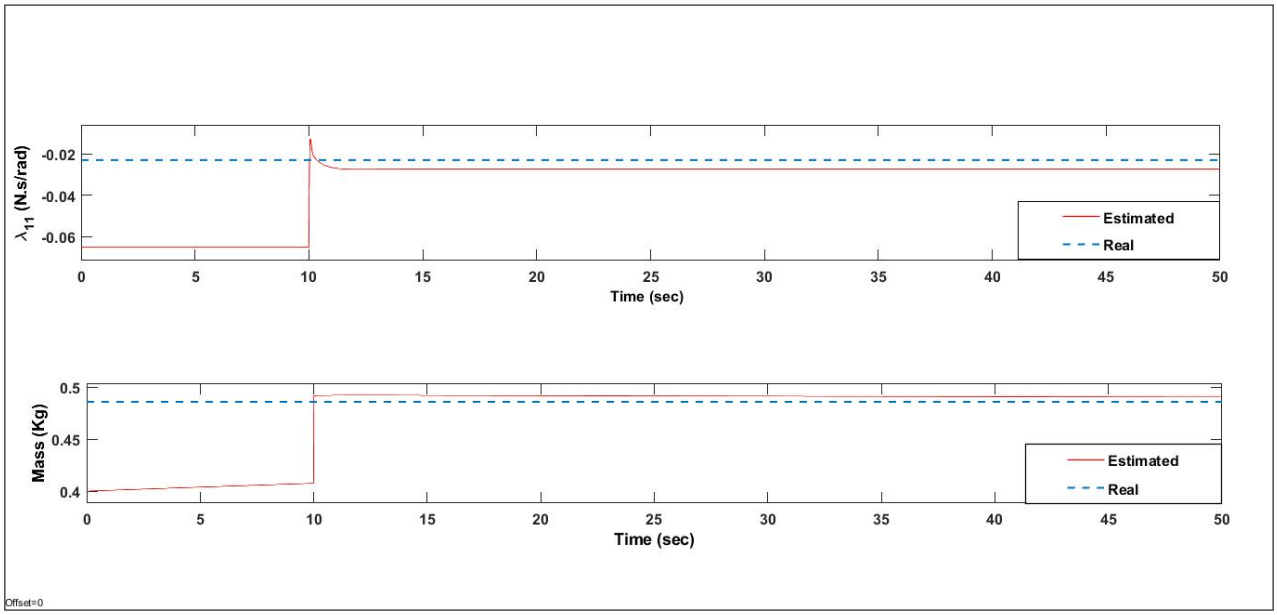


Figure 4.16: The λ_{11} and the mass m estimated outputs

4.2.6 Results

– Altitude Motion Tracking: The study involved tracking the motion of the quadrotor in terms of altitude or vertical displacement. This could refer to how well the quadrotor maintained a specific height or how accurately it followed a desired altitude profile.

– Stability of Orientation Angles: The study also examined the stability of the quadrotor's

orientation angles. This typically refers to the ability of the quadrotor to maintain a desired orientation (such as level flight) without excessive deviations or oscillations.

– Mass Variance and Unknown Parameters: The quadrotor had a variance in its mass due to the mentioned factors, and the drag coefficient in the Z direction was also unknown. Despite these uncertainties, the tracking position error was reported to be almost negligible. This suggests that the control law used in the study demonstrated robustness in the face of unknown parameters.

The results indicate that the adaptive control approach used in the study was effective in estimating the quadrotor's mass, even in the presence of variations and uncertainties. Additionally, the control law demonstrated robustness by achieving accurate altitude tracking, despite unknown parameters such as the mass and drag coefficient in the Z direction.

4.3 Conclusion

In conclusion, this chapter focused on the application of adaptive flight control to a quadrotor, considering the variations in system parameters such as mass and the inertia moment about the x and y axes. The development of adaptive control laws based on Lyapunov stability and the dynamic nature of parameters proved crucial in ensuring robust and accurate control of the quadrotor.

The adaptive control design employed backstepping techniques, allowing for the adjustment of control parameters in response to changing system dynamics. By continuously adapting the control laws to accommodate variations in the system, the quadrotor demonstrated improved tracking performance and stability in various flight scenarios.

The simulation results showcased the effectiveness of the proposed adaptive control scheme. The tracking error of the reference trajectory for both the z-axis and yaw angle converged to zero, indicating precise trajectory following. Moreover, the hovering stabilization was successfully achieved, further validating the efficacy of the adaptive control approach.

Overall, this study has demonstrated the importance and effectiveness of incorporating adaptive control techniques in quadrotor flight control. By accounting for varying system parameters, such as mass and inertia, the adaptive control laws presented in this chapter have the potential to enhance the performance and robustness of quadrotor systems in real-world applications. Further research and experimentation could explore the implementation of adaptive control in other flight maneuvers and evaluate its performance in more complex environments.

General conclusion and perspectives

In this work, we tackled the problem of stabilization and control for a multi-rotor of quadcopêtre type. The objective of the thesis was to synthesize simple and robust control laws, based on the results of recent work.

First, we identified the different components of the drone. Then, we did a bibliographic study on the control approaches applied to the quadrotor drone, particularly the recent Adaptive flight control. The literature on this subject is rich and the approaches proposed are diverse.

Before continuing our work, it was inevitable to establish a dynamic model. Based on the Newtonian approach, we presented the kinematic and dynamic modeling of the quadrotor to have a mathematical model close to reality by taking into account all the torques and forces acting on our drone. In addition, low-level modeling is presented to obtain the differential equations of the dynamics of the rotors to follow their rotation speeds during the flight.

A cascade control structure to control the dynamics of the six degrees of freedom quadrotor was used. Three control methods are used for stabilization and trajectory tracking. The first is a classic PID controller and the second is a non-linear Backstepping control and the third is an Adaptive approach. To control the rotational speeds of the different rotors, a dynamic inversion control based on Lyapunov's stability theory was used.

The results of the simulation of the control laws synthesized and applied to the model of the quadrotor and the dynamics of the rotors were presented in the last chapters. Different flight scenarios are planned to show the effectiveness of the proposed control techniques in stabilizing and following the desired trajectories.

For the rest of the work, we can list several perspectives:

– The control methods proposed in this work are efficient at the theoretical level in simulation. A promising prospect would be to implement them on board a quadrotor to carry out real experimental tests.

– To exploit the redundancy property of the quadrotor, the synthesis of a fault-tolerant control law would be beneficial to recover the drone in the event of a failure of one or more actuators.

– The choice of the gains of the various approaches is difficult to find a compromise between the various parameters of the six degrees of freedom.

– A group of drones can carry out missions more efficiently than a single drone In light of this idea, our work could consider the control of multi-robot systems.

Bibliography

[1] 3 axis gyroscope accelerometer, https://www.amazon.com/qkits-3-axis-gyroscope-accelerometer.

[2] Dhl express launches its first regular fully-automated and intelligent urban drone delivery service, https://www.prweb.com.

[3] Dhl express launches its first regular fully-automated and intelligent urban drone delivery service, https://www.prweb.com/.

[4] The dji mini 2 quadcopter drone, https://dronesnerd.com.

[5] Dji naza m v2 flight controller multi rotor system with compass gps, https://www.flyrobo.in/.

[6] Electronic speed controllers (esc's) , https://dronevolution.wordpress.com/.

[7] Flir wins additional 15.4m contract for black hornet nano-uav systems for u.s. army soldier borne sensor program.

[8] Frame drone qav 250 qav250 fibra de carbono fpv racer, https://lista.mercadolivre.com.br/.

[9] K60 pro gps drone avec caméra 6k hdr 120° grand angle, fonction suivez-moi, vole par trajectoire, equipé de vr, expérience 3d, transmission d'image haute définition 5g, https://www.amazon.fr.

[10] Moteur brushless a2212/10t 1400kv, https://heliantha.ma/.

[11] Phoenix 4 ag 10l agriculture quadcopter drone, https://avianaerospace.com.

[12] Turnigy high capacity 20000mah 6s 12c lipo pack w/xt90, https://hobbyking.com/.

[13] Xiro propeller blades for xplorer quadcopter (set of 4), https://www.bhphotovideo.com/.

[14] Andrea Alaimo, Valeria Artale, Cristina Lucia Rosa Milazzo, and Angela Ricciardello. PID controller applied to hexacopter flight. *Journal of Intelligent & Robotic Systems*, 73(1):261–270, January 2014.

[15] Rashid Ali, Yunfeng Peng, M. Touseef Iqbal, Rooh Ul Amin, Omer Zahid, and Omair Irfan Khan. Adaptive backstepping sliding mode control of coaxial octorotor unmanned aerial vehicle. *IEEE Access*, 7:27526–27534, 2019.

[16] Karima Benzaid, N. Mansouri, and Ouiddad Labbani-Igbida. Analyse comparative de différentes stratégies de commande d'un quadrirotor. 11 2013.

[17] Hakim Bouadi, S. Simoes Cunha, A. Drouin, and F. Mora-Camino. Adaptive sliding mode control for quadrotor attitude stabilization and altitude tracking. In *2011 IEEE 12th International Symposium on Computational Intelligence and Informatics (CINTI)*, pages 449–455, 2011.

[18] James Carvajal, Guanrong Chen, and Haluk Ogmen. Fuzzy pid controller: Design, performance evaluation, and stability analysis. *Information Sciences*, 123(3):249–270, 2000.

[19] Rita Cunha, David Cabecinhas, and Carlos Silvestre. Nonlinear trajectory tracking control of a quadrotor vehicle. 08 2009.

[20] History Galore. Mq-1 predator unmanned uav drone 2008, jun 27 2013.

[21] Arthur G. Green, Abdul-Rahim Abdulai, Emily Duncan, Alesandros Glaros, Malcolm Campbell, Rob Newell, Philip Quarshie, Krishna Bahadur KC, Lenore Newman, Eric Nost, and Evan D. G. Fraser. A scoping review of the digital agricultural revolution and ecosystem services: implications for canadian policy and research agendas. *FACETS*, 6:1955–1985, 2021.

[22] Md. Rifat Hazari, Effat Jahan, Md Ettaker Siraj, Md Tauhedull Islam Khan, and Ahmed Mortuza Saleque. Design of a brushless dc (bldc) motor controller. *2014 International Conference on Electrical Engineering and Information & Communication Technology*, pages 1–6, 2014.

[23] Arash Heidari, Nima Jafari Navimipour, Mehmet Unal, and Guodao Zhang. Machine learning applications in internet-of-drones: Systematic review, recent deployments, and open issues. *ACM Comput. Surv.*, 55(12), mar 2023.

[24] E. Slotine Jean-Jacques and Weiping Li. Applied nonlinear control. 1991.

[25] Byoung-Ju Jeon, Min-Guk Seo, Hyo-Sang Shin, and Antonios Tsourdos. Understandings of incremental backstepping controller considering measurement delay with model uncertainty. *Aerospace Science and Technology*, 109:106408, 2021.

[26] Anna Konert and Tomasz Balcerzak. Military autonomous drones (uavs) - from fantasy to reality. legal and ethical implications. *Transportation Research Procedia*, 59:292–299, 2021. 10th International Conference on Air Transport – INAIR 2021, TOWARDS AVIATION REVIVAL.

[27] Robert Mahony, Vijay Kumar, and Peter Corke. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robotics Automation Magazine*, 19(3):20–32, 2012.

[28] Ngo Phong Nguyen, Wonhee Kim, and Jun Moon. Super-twisting observer-based sliding mode control with fuzzy variable gains and its applications to fully-actuated hexarotors. *J. Frankl. Inst.*, 356:4270–4303, 2019.

[29] Michal Poliaĉik, Ján Murgaŝ, L'udovít Farkas, and Michal Blaho. A robust mrac modification and performance improvement in the presence of uncertainties. *IFAC Proceedings Volumes*, 43(10):186–190, 2010. 10th IFAC Workshop on the Adaptation and Learning in Control and Signal Processing.

[30] Abderahman Rejeb, Karim Rejeb, Steven J. Simske, and Horst Treiblmaier. Drones for supply chain management and logistics: a review and research agenda. *International Journal of Logistics Research and Applications*, 26(6):708–731, 2023.

[31] Max Schwenzer, Muzaffer Ay, Thomas Bergs, and Dirk Abel. Review on model predictive control: an engineering perspective. *The International Journal of Advanced Manufacturing Technology*, 117(5):1327–1349, Nov 2021.

[32] Gaurav Singhal, Babankumar Bansod, and Lini Mathew. Unmanned aerial vehicle classification, applications and challenges: A review, 11 2018.

[33] Martin Steinberger, Martin Horn, and Leonid Fridman. *Variable-Structure Systems and Sliding-Mode Control From Theory to Practice: From Theory to Practice*. 01 2020.

[34] Bailing Tian, Hanchen Lu, Zongyu Zuo, and Qun Zong. Multivariable uniform finite-time output feedback reentry attitude control for rlv with mismatched disturbance. *Journal of the Franklin Institute*, 355(8):3470–3487, 2018.

[35] Luca Trefiletti, Andrea Alaimo, V. Artale, C. Milazzo, and Angela Ricciardello. Fem analysis, modelling and control of a hexacopter. 10 2013.

[36] Sundarapandian Vaidyanathan and Ahmad Taher Azar. Chapter 1 - an introduction to backstepping control. In Sundarapandian Vaidyanathan and Ahmad Taher Azar, editors, *Backstepping Control of Nonlinear Dynamical Systems*, Advances in Nonlinear Dynamics and Chaos (ANDC), pages 1–32. Academic Press, 2021.

[37] Ngoc Vu, Duy Dang, and Tuan Le Dinh. Electric propulsion system sizing methodology for an agriculture multicopter. *Aerospace Science and Technology*, 90, 05 2019.

[38] Kun Yan and Qingxian Wu. Adaptive tracking flight control for unmanned autonomous helicopter with full state constraints and actuator faults. *ISA Transactions*, 128:32–46, 2022.

[39] Guangxue ZHANG and Haithem Taha. Adaptive back-stepping control applied on octocopter under recoil disturbance. *Journal of Engineering Science and Military Technologies*, 1(1):12–21, 2017.