

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

الجمهورية الجزائرية الديمقراطية الشعبية

MINISTRY OF HIGHER EDUCATION
AND SCIENTIFIC RESEARCH

HIGHER SCHOOL IN APPLIED SCIENCES
--T L E M C E N--



المدرسة العليا في العلوم التطبيقية
École Supérieure en
Sciences Appliquées

وزارة التعليم العالي والبحث العلمي

المدرسة العليا في العلوم التطبيقية
-تلمسان-

Mémoire de fin d'étude

Pour l'obtention du diplôme d'Ingénieur

Filière : Automatique
Spécialité : Automatique

Présenté par :
Anes DJAB
Younes Abdelaziz HASSANI

Thème

**Contrôle d'attitude d'un quad rotor à
l'aide de quaternions**

Soutenu publiquement, le 04 / 07 /2022, devant le jury composé de :

M. Fayssal ARICHI	Maître de conférences A	ESSA. Tlemcen	Président
M. Brahim CHERKI	Professeur	ESSA. Tlemcen	Directeur de mémoire
M. Mohammed Rida Mokhtari	Maître de conférences A	ESSA. Tlemcen	Co- Directeur de mémoire
Mme. Amel CHOUKCHOU BRAHAM	Professeur	UAB. Tlemcen	Examineur 1
M. Sidi Mohammed ABDI	Maître de conférences B	ESSA. Tlemcen	Examineur 2

Année universitaire : 2021 / 2022

Dedication

“

This work is dedicated,

*To my parents, my dear mother Fatima and my dear father
Sid Ahmed, for all their sacrifices, love, tenderness,
support and prayers all along my studies,*

*To my dear sister Syrine, and my dear brother Ismail, for
their constant encouragement and moral support,*

*To my dear grandparents, my dear grandmothers, and all
my family for their support all along my academic journey,*

To my friend Ismail, and all my friends and colleagues.

”

- Younes

“

It is with deep gratitude and sincere words that I dedicate this modest work to my dear parents who have sacrificed their lives for my success and who have enlightened my path by their judicious advices, may God grant them happiness and long life.

To my dear brothers for their support, strength, help and all the little moments of joy and happiness.

To my dear grandparents, my dear grandmothers, and all my family for their support all along my academic journey,

To my friends and colleagues.

”

- *Anes*

Acknowledgments

First of all, we would like to thank ALLAH the almighty and merciful, who helps us and who gave us the strength, courage and patience to accomplish this modest work.

We would like to thank our project supervisor Mr. Brahim CHERKI, Professor at the University of Aboubekr Belkaid, Tlemcen. It is with great luck that we had the honor and pleasure to work under his supervision. The work under his supervision, the meetings and discussions with him, were the reasons that made us learn a lot about the research world. Beside his know-how, Mr. CHERKI with his modesty, his patience, his good mood, his very open mind on the other, his encouragement and his permanent support, has been always setting example of the good attitude and interpersonal skills. We also thank our co-supervisor Dr. Mohammed Rida MOKHTARI for his patience, guidance and reviews.

Similarly, we would like to express our sincere thanks to Professor Amal CHOUKCHOU BRAHAM, and Mr the director of automation laboratory of the University of Tlemcen for allowing us to use the 3 DOF hover QUANSER platform.

Next, we would like to thank Dr.Fayssal ARICHI for accepting to be the president of the examiners. We would also like to thank Dr. Sidi Mohamed ABDI and Professor Amal CHOUKCHOU BRAHAM again, for doing us the honor of accepting to be a members of jury, of this project.

Résumé

Dans ce travail, nous nous intéressons à la modélisation d'un quadrotor-UAV en utilisant l'approche quaternion, qui offre une représentation globale non singulière grâce à l'utilisation d'un vecteur à quatre éléments. Nous nous intéressons également à la conception d'un contrôleur, le contrôleur actif de rejet des perturbations (ADRC). Il est principalement basé sur un observateur d'état étendu (ESO) qui estime les perturbations. L'avantage de ce contrôleur est qu'il s'agit d'un contrôleur sans modèle, donc toutes les erreurs et incertitudes du modèle sont considérées comme des perturbations. Cette loi de commande a été simulée à l'aide de Matlab/SIMULINK et testée dans des cas réels en utilisant le QUANSER 3DOF Hover. Les résultats expérimentaux ont montré l'efficacité de la méthode utilisée.

Mots clés : ADRC, quadrotor, QAV, VTOL, ESO, Problème de Wahba, Quanser 3DOF Hover, quaternion, Matlab/SIMULINK.

Abstract

In this work, we are interested in modelling a quadrotor-UAV using the quaternion-mathematics approach, which offers a non-singular overall representation through the use of four-element vector. We are also interested in studying a controller design, which is the Active Disturbance Rejection Controller (ADRC). It is mainly based on an extended state observer (ESO) that estimates the disturbances. The advantage of this controller is that it is a model-free controller, so all model errors and uncertainties are considered as disturbances. This control law has been simulated using Matlab/SIMULINK and tested in real-world cases using the QUANSER 3DOF Hover. The results of the experiment confirmed the method's effectiveness.

Keywords : ADRC, quadrotor, QUAV, VTOL, ESO, Wahba's problem, Quanser 3DOF Hover, quaternion, Matlab/SIMULINK.

ملخص

يرتكز هذا العمل اساسا على نمذجة الطائرات بدون طيار رباعية المحرك باستخدام نهج Quaternion و الذي يقدم وصفا كاملا لسلوك هذه الطائرات خاليا من اي تفرد, يهدف ايضا هذا العمل الى تصميم تحكم يضمن ثبات و استقرار الطائرات رباعية المحرك في ضل وجود الاضطرابات الخارجية و باخذ الاخطاء في النمذجة بعين الاعتبار بالاضافة الى ضمان متابعة دقيقة للمسارات المطلوبة. قمنا بمحاكاة قانون التحكم هذا باستخدام Matlab/SIMULINK واختباره في حالات العالم الحقيقي باستخدام Hover. 3DOF QUANSER أظهرت النتائج التجريبية فعالية الطريقة المستخدمة.

كلمات مفتاحية : وحدة التحكم ADRC رباعي المرواح, VTOL QUAV, Hover, 3DOF Quanser , quaternion Matlab/SIMULINK مشكل وهبة

Contents

Dedication	I
Acknowledgments	III
Résumé	IV
Abstract	V
VI	ملخص
Overall introduction	1
1 An overview of the aerial robotics systems	3
1.1 Introduction	4
1.2 Definition	4
1.3 Some historical points	4
1.4 Configuration UAV types	6
1.4.1 X Configuration Quadcopter	6
1.4.2 X Stretched Configuration Quadcopter	6
1.4.3 Quadcopter with plus (+) configuration	7
1.4.4 Y4 configuration Quadcopter	7
1.4.5 V-tail or A-Tail configuration Quadcopter	8
1.5 Sensors	9
1.5.1 IMU	9
1.5.2 Geolocation system	10
1.5.3 A barometric altimeter	10
1.5.4 Obstacle Detection	10
1.5.5 The vision sensor	11
1.6 Sensor fusion	11

1.7	Actuators and motor drivers	12
1.8	Models and control laws	12
1.9	Conclusion :	14
2	Generalities about the quaternions	15
2.1	Introduction	16
2.1.1	The quaternion discovery	16
2.1.2	Definition	16
2.2	Quaternion Algebra	16
2.2.1	Equality and Addition	16
2.2.2	Multiplication	17
2.2.3	The conjugate quaternion	18
2.2.4	Norm	18
2.2.5	Quaternion's inverse	18
2.2.6	Quaternion Differentiation	19
2.3	Quaternion Rotation Operator	20
2.4	Rotation matrix by quaternion	21
2.5	Quaternion to Euler Angle conversion	21
2.6	Conclusion	23
3	Quadrotor modeling and control law	24
3.1	Introduction	25
3.2	Basic concepts	25
3.2.1	quadrotor motion	26
3.3	Quadrotor model	28
3.3.1	Kinematic Model :	29
3.3.2	Dynamic Model :	29
3.3.3	State space representation	35
3.4	Proposed control scheme	36
3.4.1	Kinematics loop	36
3.4.2	Dynamic loop	39
3.5	Conclusion :	45
4	Quaternion attitude estimation	47
4.1	Introduction	48
4.2	Wahba's problem	48

4.3	Wahba's problem solutions	49
4.3.1	Static Attitude Determination Methods	49
4.3.2	Dynamic attitude estimation methods	51
4.4	Conclusion	53
5	Simulations Results	54
5.1	Quanser 3 DOF Hover [1]	55
5.1.1	3 DOF Hover model	55
5.2	Simulation	57
5.2.1	Hovering simulation	58
5.2.2	interpretation	62
5.3	Application on the Quanser 3 DOF Hover	62
5.3.1	Hovering	62
5.3.2	interpretation	64
5.3.3	Rotation around the z axis (YAW)(Without pertubation)	65
5.3.4	interpretation	67
5.4	Quaternion attitude estimation	67
5.4.1	MALAB implementation	68
5.5	Conclusion	70
	Conclusion and Prospects	71
	Bibliography	74

List of Figures

- 1.1 Detail of Leonardo’s ”aerial screw” 5
- 1.2 The QUAV ancestors 6
- 1.3 X Configuration Quadcopter shape [2] 7
- 1.4 X Stretched Configuration Quadcopter[2] 7
- 1.5 Quadcopter with plus (+) configuration[2] 8
- 1.6 Y4 configuration Quadcopter[2] 8
- 1.7 V-tail configuration Quadcopter [2] 9
- 1.8 Inertial Measurement Unit of a 9 DOF 9

- 2.1 Quaternion operation on vectors 20
- 2.2 XYZ rotation sequence 22

- 3.1 quadrotor structure 25
- 3.2 Influence of the angle of incidence of the blades on the lift 26
- 3.3 quadrotor configuration 26
- 3.4 Roll mouvement 27
- 3.5 Pitch mouvement 27
- 3.6 Yaw mouvement 28
- 3.7 frames used to describe the movement of the drone 29
- 3.8 proposed control scheme 36
- 3.9 Kinematic loop scheme 37
- 3.10 Dynamic loop scheme 39
- 3.11 Block diagram of the LADRC structure 45

- 5.1 3 DOF Hover 55
- 5.2 global schema representing system and controllers 58
- 5.3 perturbation 59
- 5.4 quaternions representing the system attitude 59

List of Figures

5.5	The system angular velocity	60
5.6	The observer output (disturbance estimation)	60
5.7	torques u generated by the controller	61
5.8	voltage delivered to the 4 motors	61
5.9	quaternions representing the system attitude	62
5.10	The system angular velocity	63
5.11	The observer output (disturbance estimation)	63
5.12	torques U generated by the controller	64
5.13	Voltage delivered to the 4 motors	64
5.14	quaternions representing the system attitude	65
5.15	The system angular velocity	65
5.17	torques U generated by the controller	66
5.16	The observer output (disturbance estimation)	66
5.18	Voltage delivered to the 4 motors	67
5.19	IMU-6050 chip	67
5.20	Arduino Nano3 chip	68
5.21	A snapshot from the Orientation Viewer	69
5.22	Quaternion-based attitude representation	70

List of Tables

- 1.1 Research projects on quadrotors 5
- 1.2 control strategies 14

- 5.1 System parameters 57
- 5.2 Initial conditions 58
- 5.3 Controllers and observer parameters 58

Overall introduction

By dint of recent technological advances in sensors, batteries and processing cards, which allow embarking on small vehicles all components necessary for autonomous flights at a reasonable cost, many fully autonomous aerial vehicles, also called drones, are existing today. These drones have been successfully used to respond to a wide variety of applications in several industries that requires robots to replace men in dangerous, boring or onerous situations, such as surveillance, air robotics, search and rescue operations, tactical reconnaissance, mapping and others.

Drones can be classified according to their sizes and also according to their aerodynamic functions. On this last point, we are then talking about fixed-wing drones that look like airplanes, providing high-speed travel voyage, and rotary wing drones or VTOLs (Vertical Take-Off and Landing) that can provide hovering and low-speed travel voyage.

Among the multicopter typology -which can be classified as a VTOL-UAV-, the four rotors, also called quadrotor, have been widely chosen by many researchers as a very promising vehicle for indoor and outdoor navigation [3].

Moreover, the nature of these vehicles as an under-actuated systems makes them very difficult to control and stabilize, furthermore if an unmodelled dynamic factors or/and external disturbances are introduced to these systems, then it will lead them towards instability.

To resolve the altitude and attitude issues, researchers have proposed a great deal of control laws and algorithms, however, there are still some limitations facing these solutions, like limited flight time, it can be limited by wind speed and gusts, and small payload capacity.

The work organisation

This manuscript is broken down into five chapters, which are briefly introduced in the following paragraph below

- **First chapter:** In this chapter the historical context around the development of the first rotary-wing drones will be briefly discussed . QUAV configurations such as the X-, X-Stretched-, plus-, Y4-, and V-tail- configurations will be mentioned. We

will describe a number of sensor technologies that make up a QUAV. Then we will show the current state of the art for control techniques for this sort of machine as discovered in the literature.

- **Second chapter:** In this chapter we will present some generalities about quaternions, including some basic mathematical formulas, then define the quaternion as a rotation operator, as well as the relationship between Euler angles and quaternions. The results from this chapter are going to be used as tools in the following chapters.
- **Third chapter:** This chapter presents a quaternion model for the quadrotor vehicle. The use of quaternions reduces undesirable system effects like gimbal-lock and discontinuities, which are major concerns with older methods. In addition, an ADRC control structure for a quadrotor will be proposed.
- **Fourth chapter:** In this chapter Wahba's problem will be discussed, several solutions have been proposed to this problem. As an example the solution proposed by J. L. Farrell and J. C. Stuelpnagel, Davenport's q method and the Kalman Filter will be cited.
- **Fifth chapter:** In this chapter we will test the ADRC law by using the Matlab/SIMULINK, and once again in real world application, on the QUANSER 3DOF Hover.

Chapter 1

An overview of the aerial robotics systems

- 1.1 Introduction 4
- 1.2 Definition 4
- 1.3 Some historical points 4
- 1.4 Configuration UAV types 6
- 1.5 Sensors 9
- 1.6 Sensor fusion 11
- 1.7 Actuators and motor drivers 12
- 1.8 Models and control laws 12
- 1.9 Conclusion : 14

1.1 Introduction

The aerial robotics is a term often used to define a class of highly intelligent, small machines, with a high level of mobility provided by flight. However, it is obvious that the range of systems and activities covered by the label aerial robotics could be much broader. In aerospace jargon, flying robots are usually referred to as unmanned aerial vehicles (UAVs), among this vehicles we mention the rotary wing UAVs which are mainly used for their ability of Vertical Take-off and Landing (VTOL), therefore they do not need a take-off/landing runway. In between this “Rotorcraft” we cite the quadrotor UAVs (QUAV).[4]

1.2 Definition

A quadrotor-type UAV is a helicopter with four rotors that propel it forward. Since adjacent blades revolve in opposite directions, a tail rotor is not necessary to offset the propellers’ angular momentum. As a dynamical system, changing the motor speed changes the position, for that, the forward (backward) motion is maintained by decreasing (increasing) the front (rear) rotor speed while concurrently increasing (decreasing) the rear (front) rotor speed, will change the pitch angle. Changing the roll angle in the same way allows for left and right motion. That system is under-actuated and dynamically unstable, as stated in [3].

1.3 Some historical points

Leonardo Da Vinci invented the Helical Air Screw in 1490, which is widely regarded as the first serious attempt to build a working helicopter (Figure 1.1). In 1863, Ponton d’Amécourt was the first to use the term ”Helicopter” (derived from two old Greek words: Helix and Pteron, screw and wing). He also described a coaxial helicopter and various steering methods.

29 September 1907 was the day that the first quad-copter flew, the Bréguet-Richet Gyroplane N°1 developed by Bréguet Aviation took-off vertically (0.6m), and because of a non-advanced control mechanism four guys were used to keep the structure stable, so it wasn’t a free flight [?] (Figure 1.2). Fifteen years after, two more projects were brought to the world, the Jerome-de Bothezat Flying Octopus (Figure 1.2) built by George de Bothezat for the United States Army Air Service and Ehmichen N°2 (Figure 1.2) by the french Etienne Ehmichen in the same year.

After these projects there was a lull in interest in the quad-copters development, it was until the mid-twentieth century that interest in this development resurfaced, thanks to projects funded by the United States, hence the Curtiss X-19 made in 1963 by Curtiss-Wright corporation, the Bell X-22A in 1966 by Bell Aircraft corporation and the fly vehicles of the Moller company [5, 6] (Figure 1.2). We note that all these projects were

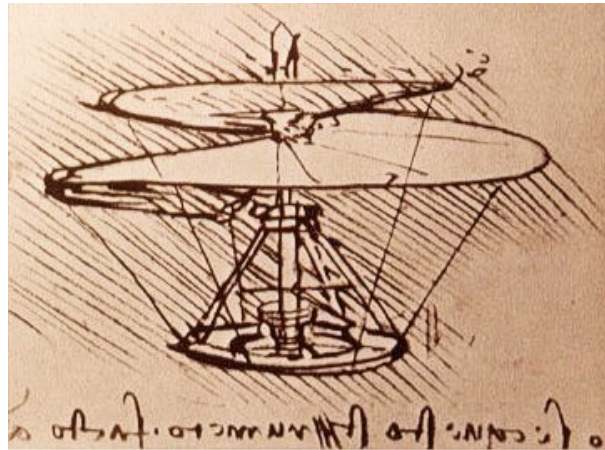


Figure 1.1: Detail of Leonardo's "aerial screw"

dependent on a pilot, in other words, they were manned vehicles, and the only unmanned quadrotor to leave ground effect was the Curtiss-Wright X-19. In the 1990s, the name "UAV" (Unmanned Aerial Vehicle) became popular for describing robotic aircraft, replacing the term "RPAV" (Remotely Piloted Aerial Vehicle) [6], because a pilot is not required on board. UAV systems have sparked attention since they allow the use of smaller aircraft with reduced power requirements. We present below a table (Table :1.1)of research laboratories and universities research and contributions:



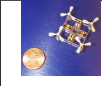








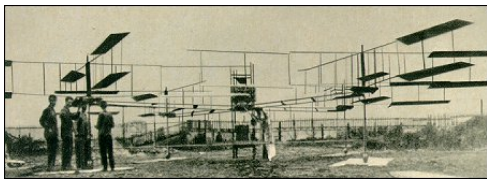
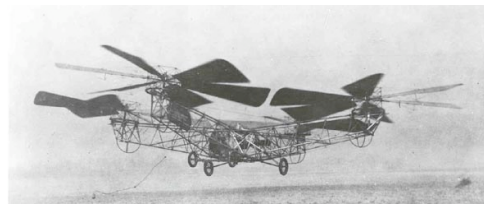
	University	Project	Year
	Dragan flyer	V Ti	1998
	Stanford	Mesicopter	2000
	Uni. Pennsylvania	E. Altuğ's thesis	2000
	Uni. Compiègne	P. Castillo's thesis	2003
	ANU	P. Pounds's thesis	2002
	Stanford	Starmac I	2004
	Stanford	Starmac II	2011
	EPFL	S.Bouabdallah's thesis	2007
	MIT	P. Tournier's thesis	2007
	CrazyFlie	CrazyFlie	2011

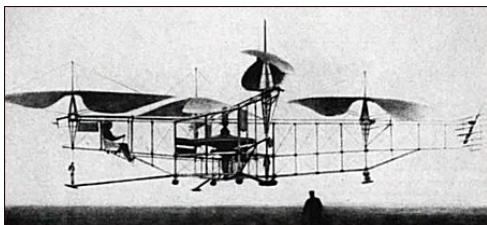
Table 1.1: Research projects on quadrotors



Gyroplane N°1



deBothezat Flying Octopus



Ehmichen N°2



Curtiss X-19



Bell X-22A

Figure 1.2: The QUAV ancestors

1.4 Configuration UAV types

1.4.1 X Configuration Quadcopter

By far the most common type of quadcopter drone design, because of its stability, the X configuration is frequently used for aerial photography and videography. This multirotor drone configuration is also popular among FPV¹ racers and acrobatic stunt pilots, it is also a very strong and stable design [2] (Figure 1.3).

1.4.2 X Stretched Configuration Quadcopter

The X stretched configuration a new version of the X configuration. This quadcopter drone design is primarily used for FPV multirotor racing. The idea is that the rear motors are moved further away from the front props, creating less turbulent air. The elongated design also improves pitch axis stability [2](Figure 1.4).

¹First Person View, or FPV, drone racing, is a sport where participants control "drones", equipped with cameras while wearing head-mounted displays showing the live stream camera feed from the drones

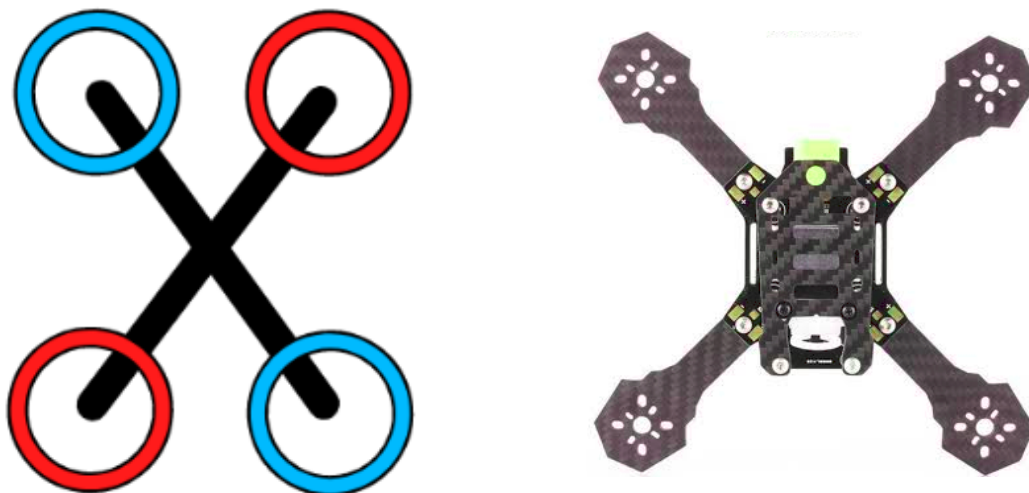


Figure 1.3: X Configuration Quadcopter shape [2]



Figure 1.4: X Stretched Configuration Quadcopter[2]

1.4.3 Quadcopter with plus (+) configuration

This quadcopter drone with the plus configuration that allows it to track its trajectories very well when flying straight. Many people believe the propellers are in a better aerodynamic position. The + configuration quadcopter is commonly used in acrobatic and FPV stunt flying because it flies more like an airplane and is easier to control than the X configuration quadcopter [2] (Figure 1.5).

1.4.4 Y4 configuration Quadcopter

This quadcopter drone appears like a tricopter, but it features a second brushless motor placed upon a first one instead of a servo. The two rear motors' speeds can be changed to manage yaw control. Y4 multirotors have higher lifting power and are more robust than tricopters [2] (Figure 1.6).

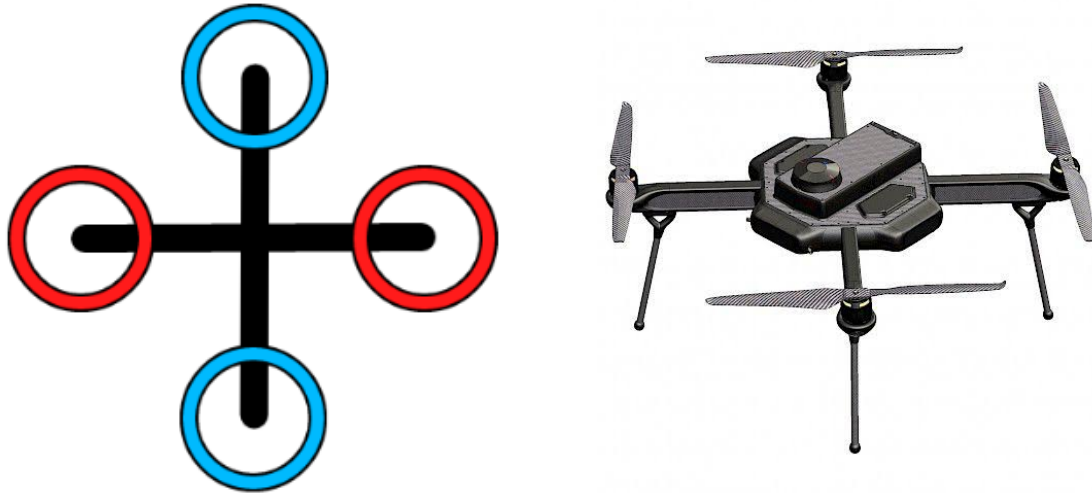


Figure 1.5: Quadcopter with plus (+) configuration[2]

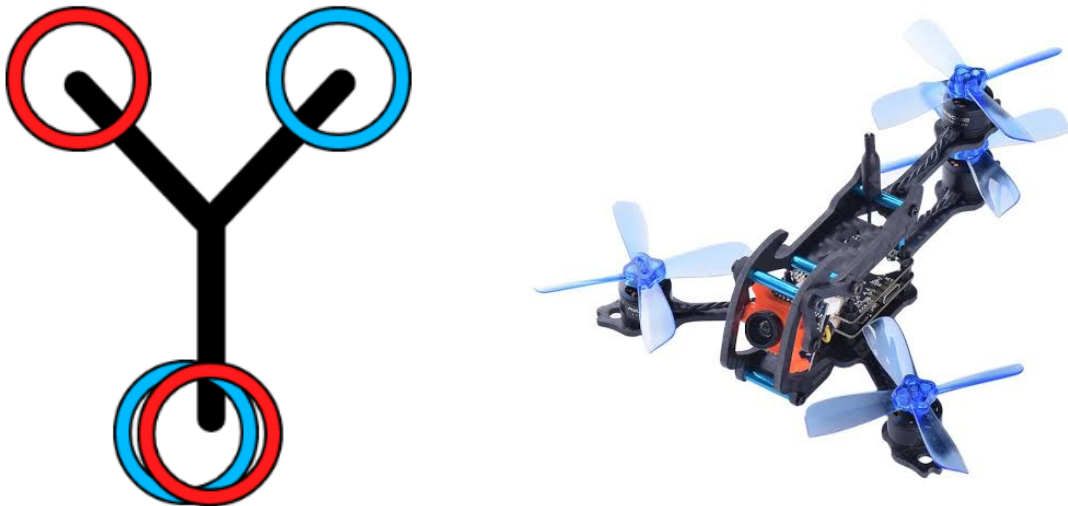


Figure 1.6: Y4 configuration Quadcopter[2]

1.4.5 V-tail or A-Tail configuration Quadcopter

It is a form of quadcopter drone having typical quadcopter arms on the front and angled at a vertical angle on the back. A V-Tail, also known as an A-Tail, is similar to a Y4 but has far greater Yaw authority since it uses thrust to turn rather than counter engine torque. Because it is less efficient than other quadcopter designs, this sort of multirotor drone is not particularly widespread [2] (Figure 1.7).

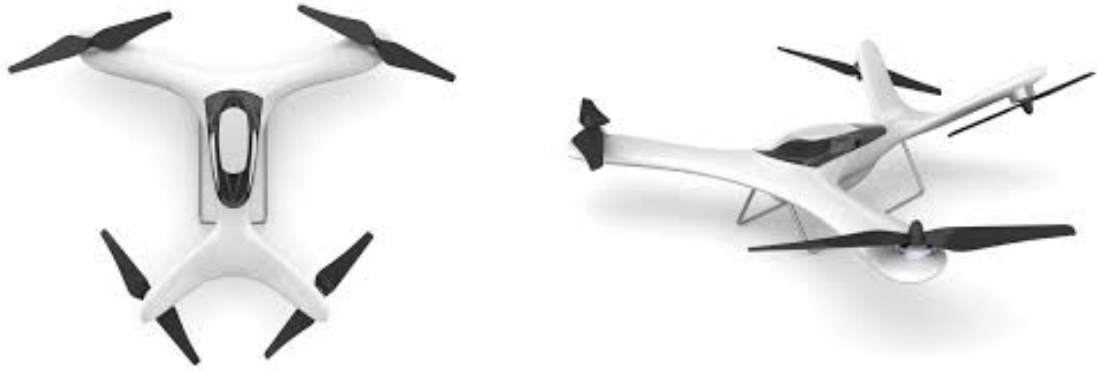


Figure 1.7: V-tail configuration Quadcopter [2]

Among all these configuration types, and in terms of stability, the x-configuration quadrotor is considered to be the most stable [7, 8].

1.5 Sensors

1.5.1 IMU

An IMU (Inertial Measurement Unit) is a sensor capable of providing us with the orientation (attitude) and the velocity and position vectors of a body through inertial sensors. This unit is considered as one of the most important parts of an UAV system (QUAV in our work), because of the necessity of its collected data in estimating the system's attitude and the performance of the control laws (Figure 1.8).

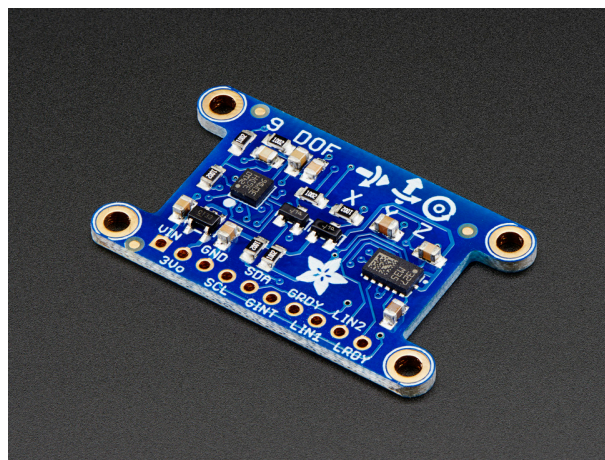


Figure 1.8: Inertial Measurement Unit of a 9 DOF

The main sensors mounted on an IMU are :

Triaxial accelerometer : Accelerometers are instruments that measure acceleration, or the rate at which an object's velocity changes. A three-axis accelerometer monitors three orthogonal axes' linear accelerations. An accelerometer's advantage is its ability to reveal a large amount of data (acceleration, speed, displacement, force...).

Triaxial gyroscope : The gyroscope measures the angular speeds caused by the rotation of the body-fixed frame around the inertial frame (roll, pitch and yaw speeds). The combination of the three measures determines the vehicle's attitude.

Magnetometer : A magnetometer, also known as a magnetic compass, is a sensor that measures the direction and intensity of magnetic fields and more specifically Earth's magnetic field by indicating the magnetic north.

1.5.2 Geolocation system

A satellite navigation system, often known as a satnav system, is a system that employs satellites to offer autonomous geospatial location. The system may be used to provide position, navigation, or track the location of something that has a receiver attached to it (satellite tracking). One of the many benefits of GPS is that it works in any weather, anywhere in the globe, at any time.

1.5.3 A barometric altimeter

As we know atmospheric pressure changes with respect to height from the sea level. This phenomenon is used in the barometer to determine height of Unmanned Aerial Vehicle. A barometric altimeter is a sensor that measures the altitude difference between the sensor level and the reference level. Its weakness is that climate change has a big impact on this measure [9].

1.5.4 Obstacle Detection

It's an active sensor for measuring the distance between an object and an obstacle, in purpose to avoid collision. The Ultrasonic (Figure 1.9b) and Laser sensors (Figure 1.9a) are the most used in the industry.



(a) Laser sensor



(b) Ultrasonic sensor

1.5.5 The vision sensor

The vision sensor is one of the information sources used to locate a vehicle in its environment. Stereo vision is a vision technology that employs several cameras, often two or three. The goal of this strategy is to blend several points of view. This enables the information to be enriched by comparing it to a single point of view.

We note that all these sensors are MEMS (Microelectromechanical systems) technology based, because of their lightweight and low cost.

1.6 Sensor fusion

Sensor fusion is a technique for combining signals from numerous sources. It enables the extraction of information from several inputs and combining it into a single signal or information. Sensors or other devices that allow perception or measurement of changing environments are frequently used as information sources. Sensor fusion or data fusion techniques are used to process information from many sensors. These algorithms are divided into three categories. First, there is probabilistic fusion, followed by least-squares fusion, and last, intelligent fusion. Bayesian reasoning, evidence theory, robust statistics, and recursive operators are examples of probabilistic model approaches. Kalman filtering, optimum theory, regularisation, and uncertainty ellipsoids are all least-squares approaches. The intelligent fusion methods are fuzzy logic, neural networks and genetic algorithms [10]. All this types can be exploited using one of the two following approach :

Centralized sensor fusion algorithms : this approach handles all measurements through a single central filter. Because all measurements must be computed in this situation, the computing burden is substantial, needing high-performance hardware. Furthermore, ensuring the system's resistance against sensor failure or temporary incorrect input data is difficult. Despite these drawbacks, it has the benefit of providing the optimal solution [11].

Distributed sensor fusion algorithms : this approach do not handle all measurements in a single central filter, but rather fuse all data in the main filter after processing each sensor's data in a separate sub-filter. In other words, they estimate state variables and covariance by running each data through a sub-filter. This estimate is sent to the main filter, which then estimates the overall optimal state variables [11].

Adding all this complex calculus to our system will yield many benefits, it can increase the quality of the data and their reliability, it can help to estimate the unmeasured states (it is important to recognise that unmeasured doesn't mean unmeasurable), and finally, it can increase the coverage area around our dynamic system.

We note that all the data are interpreted by the Micro Controller Unit (MCU).

1.7 Actuators and motor drivers

As with all dynamic systems, the QUAV needs actuators to produce action or motion. Much research has used the DC motor as an actuator [12], and more specifically, brushless DC motors (BLDC). High speed, acceleration, and efficiency are reasons for choosing the latter motor type, but the main reason is that the BLDC motors (no brushes to wear) have a longer lifetime compared with the brushed ones (brushes wear out). We add that the drivers are fundamental to integrating the Micro Controller Unit (MCU) outputs to the motors (connected to the propellers).

1.8 Models and control laws

To perform certain flight missions to a high quality, a control system must be designed and an accurate model must be established [13]. For that, the two widely used methodologies for modeling robots (the QUAV in our case) are **the Euler-Lagrange** formulation and **the Newton-Euler** formulation. In the Euler-Lagrange approach, the entire physical description of the manipulator is first included into the Lagrangian in terms of a set of generalised coordinates and velocities, and then the Lagrangian equations of motion are developed using a systematic technique. The Newton-Euler technique, on the other hand, applies Newton's law and Euler's equation for linear and angular motion to individual bodies [14].

Newton's technique is of particular relevance to us in this research.

A rigid body's rotating orientation (attitude) in three-dimensional Euclidean space is defined by three parameters. There are several parametrization approaches for mathematically representing the attitude transformation or rotation of a rigid body [15].

A **direction cosine matrix** is a transformation matrix that is made up of the direction cosine values between the initial and target coordinate systems, however, DCM is limited by the orthogonality requirement, which necessitates that all matrices have the same dimension for matrix operations [15].

Secondly there's **the Euler angles**, three sequential transformations about the body fixed axis can be used to represent the orientation of a rigid body in relation to an inertial coordinate system. The Euler angles are the three angles used in the successive transformation. Because they are reasonably simple to interpret, they are commonly employed for graphical displays of spacecraft orientation. However, describing the attitude by this method can lead to major problem : the singularity problem shown in the equation(1)

linking the angular velocity vector w and the Euler angles proven by [16], (as we notice when the pitch angle is near to 90° ($\pi/2$) fraction $1/\cos\theta$ is not defined), the physical interpretation of the this singularity is the gimbal lock [15].

Gimbal lock happens when the axes of two of the three gimbals are pushed into a parallel configuration, "locking" the system into rotation in a degenerate two-dimensional space, resulting in the loss of one degree of freedom in a three-dimensional, three-gimbal mechanism. Consider a level-sensing platform on a plane heading north with three gimbal axes that are all perpendicular to each other (i.e., roll, pitch and yaw angles each zero). When the aircraft pitches up 90 degrees, the yaw axis gimbal on the aircraft and platform becomes parallel to the roll axis gimbal, and yaw variations can no longer be compensated for.[17]

Also, both the DCM and Euler angles are methods that consume a huge computation by solving sine- cosine functions and using the Jacobian for system states.

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \frac{1}{\cos\theta} \begin{bmatrix} \cos\theta & \sin\phi \sin\theta & \cos\phi \sin\theta \\ 0 & \cos\phi \cos\theta & -\sin\phi \cos\theta \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \Omega \quad (1)$$

Problematic : how to get rid of those limitations (Singularities, the Gimble-Lock, the orthogonality requirement and the heavy computation) ?

Now we skip to the control laws and algorithms. As we mentioned before the QUAV system is by nature unstable, so many research have been conducted in recent decades on the control of rotary-wing flying machines. We present below some control strategies used in the literature on rotary wing flying models.

Model	Proposed control law/ algorithm	Limitations
Newton-Euler	Fuzzy variable structure control with sensor based Kalman filter design.[18]	While tracking, some translational and angular faults were discovered. Furthermore, there is a lag in detecting changes in physical characteristics.
Newton-Euler	Linear active disturbance with particle swarm optimization algorithm. [19]	Overshoots have been reported, as well as tracking inaccuracies in attitude angle.
Newton-Euler	Geometric control design with non- linear disturbance observer. [20]	This work cannot observe and estimate the non-smooth changes in unmodelled dynamic factors and produces the tracking errors.
Newton-Euler	Predictive optimal control design with disturbance observer.[21]	The tracking performance is robust, but only when the payload varies.
Quaternion-based	Backstepping control design with nonlinear reference model.[22]	For modest payload masses, there is a tiny steady-state error. This might lead to an increase in steady-state error.
Newton-Euler	Dual loop integral sliding mode control with linear extended state observer (LESO). [23]	The use of the hyperbolic function and integral form of SMC results in an unnecessary delay as well as a Zeno effect.
Newton-Euler	H-infinity control with robust compensator design.[24]	This has a Zeno effect, as well as a slight divergence off the trajectory.

Table 1.2: control strategies

1.9 Conclusion :

In this chapter, we briefly introduced the historical context associated with the emergence of the first rotary-wing drones. Different QUAV configurations were mentioned such as X-Configuration, X-Stretched Configuration, plus-configuration, Y4 configuration and the V-tail configuration. We have described a number of sensor technologies that make up a QUAV. Then, we have presented a state of the art concerning the control strategies found in the literature for this type of machine.

Chapter 2

Generalities about the quaternions

2.1	Introduction	16
2.2	Quaternion Algebra	16
2.3	Quaternion Rotation Operator	20
2.4	Rotation matrix by quaternion	21
2.5	Quaternion to Euler Angle conversion	21
2.6	Conclusion	23

2.1 Introduction

2.1.1 The quaternion discovery

In 1843 the term quaternion was introduced by the Irish mathematician William Rowan Hamilton as an extension of complex numbers.

One of Hamilton's purposes for pursuing three-dimensional complex numbers was to discover a description of rotation in space that correspond to the complex numbers, where a multiplication corresponds to a rotation and scaling in the plane.

While walking by the Royal Canal in Dublin on a Monday in October 1843, Hamilton realized that four numbers are needed to describe a rotation followed by a scaling which is the solution for the problem of multiplication that he faced when he supposed that an extended complex number is composed by only three components and written as : $q = s + xi + yj$.

Hamilton found a closed multiplication for four-dimensional complex numbers of the form $ix + jy + kz$, where $i^2 = j^2 = k^2 = ijk = -1$. So, he dubbed his four-dimensional complex numbers quaternions. This event is marked by a plaque at the exact location into the stone of the bridge [25].

2.1.2 Definition

Quaternions, which belong to the quaternion space \mathbb{H} , are hyper complex numbers that represent rotations in the 3D space, they are widely used instead of Euler angles because of their simple implementation also they avoid the gimbal blocking problem contrary to Euler angles.

A Quaternion is a 4x1 matrix which elements consists of a scalar part $q_s \in \mathbb{R}$ and a vector part $\bar{q} \in \mathbb{R}^3$. [15]

$$q = q_s + q_x i + q_y j + q_z k = \begin{pmatrix} q_s \\ \bar{q} \end{pmatrix} = \begin{pmatrix} q_s \\ q_x \\ q_y \\ q_z \end{pmatrix} \quad (2.1)$$

2.2 Quaternion Algebra

2.2.1 Equality and Addition

Two quaternions are equal if and only if they have exactly the same components [26], if $p = p_0 + p_1 i + p_2 j + p_3 k$ and $q = q_0 + q_1 i + q_2 j + q_3 k$ then $p = q$ if and only if :

$$\begin{aligned}
 p_0 &= q_0 \\
 p_1 &= q_1 \\
 p_2 &= q_2 \\
 p_3 &= q_3
 \end{aligned} \tag{2.2}$$

The sum of the two quaternions p and q is defined as follow :

$$p + q = (p_0 + q_0) + (p_1 + q_1)i + (p_2 + q_2)j + (p_3 + q_3)k \tag{2.3}$$

2.2.2 Multiplication

The multiplication of two quaternions q and p is non commutative and performed by a tensorial product named Kronecker product, denoted as \otimes , the outcome is presented in the following equation [27]:

$$q \otimes p = \begin{pmatrix} p_0q_0 - \bar{p} \cdot \bar{q} \\ q_0\bar{p} + p_0\bar{q} + \bar{q} \times \bar{p} \end{pmatrix} = \begin{pmatrix} p_0q_0 - q_1p_1 - q_2p_2 - q_3p_3 \\ q_1p_0 + q_0p_1 + q_2p_3 - q_3p_2 \\ q_2p_0 + q_0p_2 + q_3p_1 - q_1p_3 \\ q_3p_0 + q_0p_3 + q_1p_2 - q_2p_1 \end{pmatrix} \tag{2.4}$$

$$\begin{aligned}
 q \otimes p &= \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{pmatrix} = Q(q)p \\
 q \otimes p &= \begin{bmatrix} p_0 & -p_1 & -p_2 & -p_3 \\ p_1 & p_0 & -p_3 & p_2 \\ p_2 & p_3 & p_0 & -p_1 \\ p_3 & -p_2 & p_1 & p_0 \end{bmatrix} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} = \bar{Q}(p)q
 \end{aligned} \tag{2.5}$$

We state the following properties of quaternion multiplication [25]:

Let $p, q, r \in \mathbb{H}$ then :

$$(p \otimes q) \otimes r = p \otimes (q \otimes r)$$

$$r \otimes (p + q) = r \otimes p + r \otimes q$$

$$(p + q) \otimes r = p \otimes r + q \otimes r$$

The identity quaternion ,denoted q_I , which satisfy $q \otimes q_I = q_I \otimes q = q$, is $q_I = [1 \ 0 \ 0 \ 0]^T$ [28].

2.2.3 The conjugate quaternion

The conjugate denoted q^* is obtained by inverting the sign of the vector part [15].

$$q^* = \begin{pmatrix} q_s \\ -q_x \\ -q_y \\ -q_z \end{pmatrix} \quad (2.6)$$

From the definition we can deduce [29]:

$$(q^*)^* = q \quad (2.7)$$

$$q + q^* = [2q_0 \ 0 \ 0 \ 0]^T \quad (2.8)$$

$$q^* \otimes q = q \otimes q^* \quad (2.9)$$

Given two quaternions q and p :

$$(p \otimes q)^* = q^* \otimes p^* \quad (2.10)$$

2.2.4 Norm

The quaternion norm $\|q\| \in \mathbb{R}$ is defined in the same way as complex numbers and it's given by equation:

$$\|q\| = \sqrt{q \otimes q^*} = \sqrt{q_s^2 + q_x^2 + q_y^2 + q_z^2} \quad (2.11)$$

The norm of the product of two quaternions p and q is the product of the individual norms, for we have [29]:

$$\begin{aligned} |pq|^2 &= (pq)(pq)^* \\ &= pqq^*p^* \\ &= p|q|^2p^* \\ &= pp^*|q|^2 \\ &= |p|^2|q|^2 \end{aligned} \quad (2.12)$$

A Quaternion with the norm $|q| = 1$ is called unit quaternion. All quaternions for attitude representation are unit quaternions [15].

2.2.5 Quaternion's inverse

To obtain the inverse of a quaternion its conjugate is normalized.

$$q^{-1} = \frac{q^*}{|q|} \quad (2.13)$$

The invers is equal to the conjugate for unit quaternions.

2.2.6 Quaternion Differentiation

Let $q(t)$ be a unit quaternion function, and $w(t)$ a quaternion associated to the angular velocity of the same sequence of rotation denoted by q , the derivative of q is given by [29]:

$$\dot{q} = \frac{1}{2}q \otimes w \quad (2.14)$$

Proof. At $t + \Delta t$, the rotation is described by $q(t + \Delta t)$. This is after some extra rotation during Δt performed on the frame that has already undergone a rotation described by $q(t)$. This extra rotation is about the instantaneous axis $\hat{w} = w/\|w\|$ through the angle $\Delta = \|w\|\Delta t$. It can be described by a quaternion [29]:

$$\begin{aligned} \Delta q &= \cos \frac{\Delta\theta}{2} + \hat{w} \sin \frac{\Delta\theta}{2} \\ &= \cos \frac{\|w\|\Delta t}{2} + \hat{w} \sin \frac{\|w\|\Delta t}{2} \end{aligned} \quad (2.15)$$

So the rotation at $t + \Delta t$ can be described by the quaternion sequence $q(t)$, Δq

$$\Rightarrow q(t + \Delta t) = q(t) \otimes \Delta q \quad (2.16)$$

After setting this equations we are now ready to obtain the derivative $\dot{q}(t)$. First let us obtain the difference (by (2.15) and (2.16)):

$$\begin{aligned} q(t + \Delta t) &= \left(\cos \frac{\|w\|\Delta t}{2} + \hat{w} \sin \frac{\|w\|\Delta t}{2} \right) q(t) - q(t) \\ &= -2 \sin^2 \frac{\|w\|\Delta t}{4} q(t) + \hat{w} \sin \frac{\|w\|\Delta t}{2} q(t) \end{aligned} \quad (2.17)$$

The first term in the last equation above is of higher order than Δt , thus its ratio to Δt goes to zero as the latter does. Hence

$$\begin{aligned} \dot{q}(t) &= \lim_{\Delta t \rightarrow 0} \frac{q(t + \Delta t) - q(t)}{\Delta t} \\ &= \hat{w} \lim_{\Delta t \rightarrow 0} \frac{\sin \frac{\|w\|\Delta t}{2}}{\Delta t} q(t) \\ &= \hat{w} \frac{d}{dt} \sin \frac{\|w\|t}{2} \Big|_{t=0} q(t) \\ &= \hat{w} \frac{\|w\|}{2} q(t) \\ &= \frac{1}{2} q(t) \otimes w(t) \end{aligned} \quad (2.18)$$

□

The solution of this differential equation which represents the trajectory evolution of a rigid body in the three dimensional space is presented as [28] :

$$q(t) = q(t_0)e^{\int \frac{1}{2}w(t)dt} \quad (2.19)$$

To calculate $e^{\int \frac{1}{2}w(t)dt}$ we use the third order Taylor series expansion, we note $\delta = \int \frac{1}{2}w(t)dt$

$$e^\delta = 1 + \delta + \frac{1}{2}\delta^2 + \frac{1}{6}\delta^3 = \begin{pmatrix} 1 - \frac{1}{2}\delta^2 \\ \delta_1 - \frac{1}{6}\delta_1|\delta|^2 \\ \delta_2 - \frac{1}{6}\delta_2|\delta|^2 \\ \delta_3 - \frac{1}{6}\delta_3|\delta|^2 \end{pmatrix} \quad (2.20)$$

2.3 Quaternion Rotation Operator

As seen before a unit quaternion can describe a rotation or a sequence of rotations of a rigid body by a single rotation around a fixed axis, and it is defined in the form:

$$q = \begin{bmatrix} q_s \\ q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} \cos \frac{\theta}{2} \\ \vec{e} \cdot \sin \frac{\theta}{2} \end{bmatrix} \quad (2.21)$$

Where \vec{e} is the normalized rotational axis and θ the angle of rotation.

Multiplying two or more rotation quaternions produces another rotation quaternion that represents the total rotation. The three quaternion product can be used to rotate a 3D vector from one reference frame to another, but first vectors must be transformed in quaternions with a scalar part equal to zero [30].

$$\begin{bmatrix} 0 \\ v' \end{bmatrix} = L(q, v) = q \otimes \begin{bmatrix} 0 \\ v \end{bmatrix} \otimes q^*$$

where $v, v' \in \mathbb{R}^3$ and $\|q\| = 1$.

The operator $L(q, v)$ is linear as it does not change the magnitude of v .

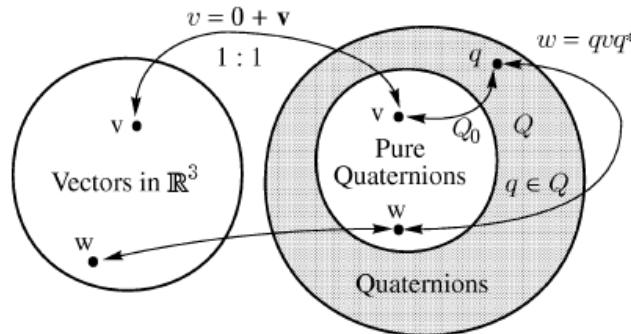


Figure 2.1: Quaternion operation on vectors

Note:

Every rotation can be expressed by two quaternions q and q^- where

$$q^- = \begin{bmatrix} -q_0 \\ -q_1 \\ -q_2 \\ -q_3 \end{bmatrix} = \begin{bmatrix} \cos \frac{2\pi-\theta}{2} \\ -\vec{e} \cdot \sin \frac{2\pi-\theta}{2} \end{bmatrix} = \begin{bmatrix} \cos(\pi - \frac{\theta}{2}) \\ -\vec{e} \cdot \sin(\pi - \frac{\theta}{2}) \end{bmatrix} \quad (2.22)$$

This becomes clear, if we imagine q^- as a rotation with angle of $2\pi - \theta$ and $-\vec{e}$ the opposite axis to q [15].

2.4 Rotation matrix by quaternion

The rotation matrix can be used to identify the orientation of any coordinate system in the three dimensional space. It provides the transformation from the inertial frame to the body fixed frame. The rotation matrix is obtained by transforming each axis from the inertial frame to the body-fixed frame, this result matrix is attained by multiplying the unit-vector which represent each axis by the quaternion vector that describe the rotation between the two frames.

$$R_x(q) = q \otimes \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \otimes q^* = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 \\ 2(q_1q_2 + q_0q_3) \\ 2(q_1q_3 - q_0q_2) \end{bmatrix}$$

$$R_y(q) = q \otimes \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \otimes q^* = \begin{bmatrix} 2(q_1q_2 - q_0q_3) \\ q_0^2 - q_1^2 + q_2^2 - q_3^2 \\ 2(q_3q_2 + q_0q_1) \end{bmatrix}$$

$$R_z(q) = q \otimes \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \otimes q^* = \begin{bmatrix} 2(q_1q_3 + q_0q_2) \\ 2(q_3q_2 - q_0q_1) \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

Rotation matrice expressed in quaternion is given as follow :

$$R(q) = [R_x(q) \quad R_y(q) \quad R_z(q)] \quad (2.23)$$

2.5 Quaternion to Euler Angle conversion

Because of their apparent physical explanation, Euler Angles are still widely used, they can be defined in many ways depending on the order of rotation.

Let assume that the sequence of rotation is given as XYZ where the first rotation about X axis is represented by the angle ϕ , then about Y axis by θ and finally about Z axis by

ψ . [16]

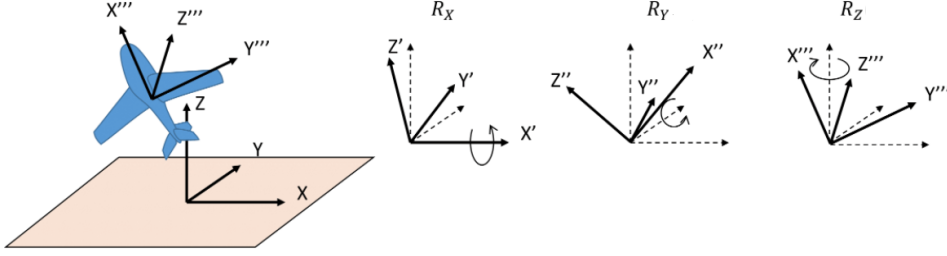


Figure 2.2: XYZ rotation sequence

Then the rotation matrix converting a vector from final frame (x, y, z) to original frame (X, Y, Z) is given by :

$$\begin{aligned}
 \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} &= R_\phi R_\theta R_\psi \begin{bmatrix} x \\ y \\ z \end{bmatrix} \\
 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \\
 &= \begin{bmatrix} \cos \theta \cos \psi & -\cos \theta \sin \psi & \sin \theta \\ \cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi & \cos \phi \cos \psi - \sin \phi \sin \theta \sin \psi & -\sin \phi \cos \theta \\ \sin \phi \sin \psi - \cos \phi \sin \theta \cos \psi & \sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.24}
 \end{aligned}$$

comparing element by element the two rotation matrices, the one expressed in quaternion and the second one expressed with the Euler angle we can find Euler angles from the quaternion[16]:

$$\phi = \tan^{-1} \left(\frac{-2(q_2 q_3 - q_0 q_1)}{q_0^2 - q_1^2 - q_2^2 + q_3^2} \right) \tag{2.25}$$

$$\theta = \sin^{-1} (2(q_0 q_2 + q_1 q_3)) \tag{2.26}$$

$$\psi = \tan^{-1} \left(\frac{-2(q_1 q_2 - q_0 q_3)}{q_0^2 + q_1^2 - q_2^2 - q_3^2} \right) \tag{2.27}$$

We can solve the preceding system of equations to get quaternion in terms of Euler angles, but it is tedious to solve. Instead, we can directly construct quaternion from the knowledge of Euler sequence. For the above sequence, we first represent the quaternions

for the three rotations as:

$$q_\phi = \begin{bmatrix} \cos(\frac{\phi}{2}) \\ \sin(\frac{\phi}{2}) \\ 0 \\ 0 \end{bmatrix}, q_\theta = \begin{bmatrix} \cos(\frac{\theta}{2}) \\ 0 \\ \sin(\frac{\theta}{2}) \\ 0 \end{bmatrix}, q_\psi = \begin{bmatrix} \cos(\frac{\psi}{2}) \\ 0 \\ 0 \\ \sin(\frac{\psi}{2}) \end{bmatrix}$$

Compounding the rotation as before, i.e., post-multiplying the local rotations, we get :

$$q = q_\phi \otimes q_\theta \otimes q_\psi$$

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos(\phi/2) \cos(\theta/2) \cos(\psi/2) - \sin(\phi/2) \sin(\theta/2) \sin(\psi/2) \\ \cos(\phi/2) \sin(\theta/2) \sin(\psi/2) + \sin(\phi/2) \cos(\theta/2) \cos(\psi/2) \\ \cos(\phi/2) \cos(\psi/2) \sin(\theta/2) - \sin(\phi/2) \cos(\theta/2) \sin(\psi/2) \\ \cos(\phi/2) \cos(\theta/2) \sin(\psi/2) + \cos(\psi/2) \sin(\theta/2) \sin(\phi/2) \end{bmatrix} \quad (2.28)$$

The above expression can be verified with the NASA documentation [31].

2.6 Conclusion

In this chapter we have presented some generalities about quaternions where we have given some basic mathematical formulas, then we have defined the quaternion as an operator for rotation also the relation between Euler angles and quaternions, the results found in this chapter will be used as tools for the following chapters.

Chapter 3

Quadrotor modeling and control law

3.1	Introduction	25
3.2	Basic concepts	25
3.3	Quadrotor model	28
3.4	Proposed control scheme	36
3.5	Conclusion :	45

3.1 Introduction

This section is devoted to modeling and control design for the quadricopter. The modeling is very important because it enables to describe the behavior of the helicopter and how it moves according to its inputs. The dynamic model of a quadrotor is built with the Newton-Euler formalism, it permits to predict the positions and velocities reached by the helicopter by studying only the four rotors speed. Modeling is important, but at the same time is not an easy task owing to the complexity of aerodynamic phenomena and the sensitivity of UAVs to external disturbances. The dynamic models are far from being exact, leading to difficult problems in the control of the UAV's.

To tackle these problems we have chosen the active disturbance rejection control (ADRC) which can handle the two main problems in the quadrotor dynamic description that are external disturbance and model uncertainty.

3.2 Basic concepts

Here we are interested in the cross configuration, it has six degrees of freedom, three for the translation, and three for the rotation movement. This structure is robust owing to its thines and lightness, and the way the four motors are connected [32].



Figure 3.1: quadrotor structure

Aerodynamic characterization of rotary wings: Each rotor possess of a number of propellers. The propeller is the most influential element most influential element in the dynamics of rotating wings. Each propeller is related to the motor through the reduction gears. All the propellers axes of rotation are fixed and parallel. Furthermore, they have fixed-pitch blades and their air flows points downwards (to get an upward lift). When the blade is perfectly horizontal (zero incident), the pressure difference between the upstream and downstream of the profile is zero. The increase in the angle of incidence of the airfoil leads to an increase in the overall pressure and consequently the lift that is created also increases. When the profile incidence reaches a certain value, called the stall incidence, the depression field on the upper surface suddenly decreases and the bypass becomes turbulent. These different configurations are illustrated by the figure bellow.

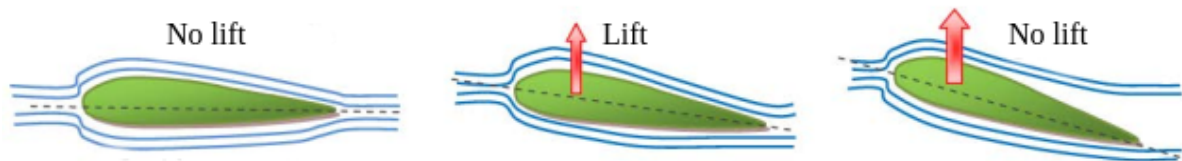


Figure 3.2: Influence of the angle of incidence of the blades on the lift

3.2.1 quadrotor motion

Forward/backward left/right motions are achieved just by pitching or rolling in the desired direction, so to control the motion of a quadcopter it suffices to control the roll, pitch and yaw motion, these three movements are obtained through a differential control strategy of the thrust generated by each rotor, In order to avoid the yaw drift due to the reactive torques the front and the rear propellers rotate counter-clockwise, while the left and the right ones turn clockwise [33].

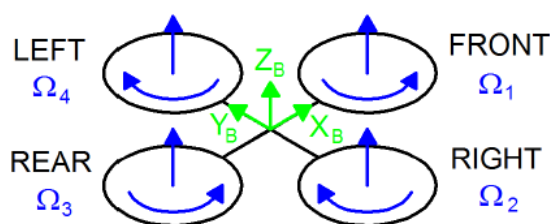


Figure 3.3: quadrotor configuration

Hovering

In hover flight, the four propellers rotate at the same speed, a speed that guarantees the creation of thrust that counterbalance the acceleration due to gravity.

Roll

to achieve this motion left propeller speed must be increased (or decreased) and the right one decreased (or increased), this difference between the propellers speed creates a torque with respect to x_b axis which makes the quadrotor turn. The roll motion leads to a translation motion along y_b axis.

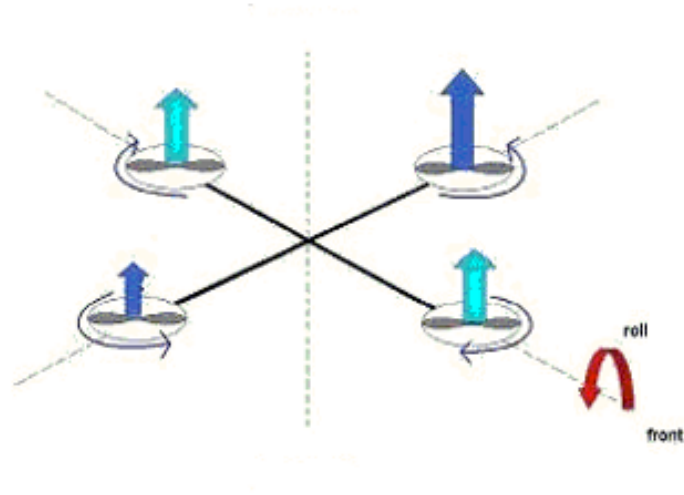


Figure 3.4: Roll movement

Pitch

similarly to the roll, to achieve pitch motion rear propeller speed must be increased (or decreased) and the front one decreased (or increased), this difference between the propellers speed creates a torque with respect to y_b axis which makes the quadrotor turn. The pitch motion leads to a translation motion along x_b axis (forward/backward).

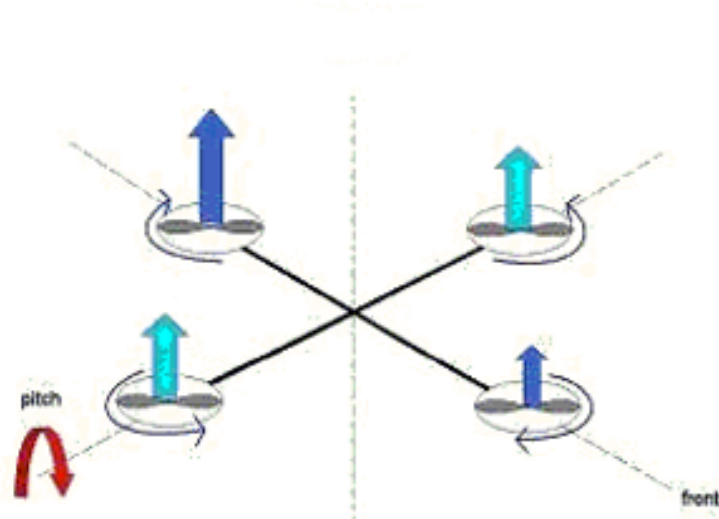


Figure 3.5: Pitch movement

Yaw

In order to have a rotation around the z_b axis, we increase the speed of the two propellers that turn in the direction that we want make the quadrotor turn (clockwise or counter-clockwise), while decreasing the speed of the other two that turn in the opposite direction.

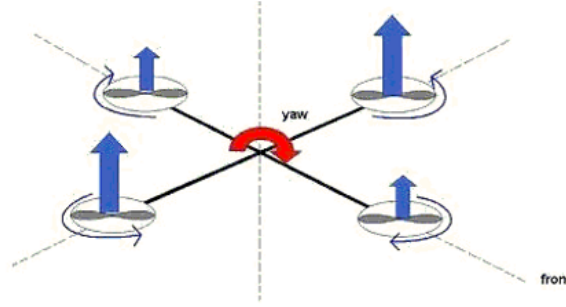


Figure 3.6: Yaw mouvement

3.3 Quadrotor model

Answering the problematic mentioned in the first chapter, a quaternion mathematics approach is used to build our system's model. Using this approach will results also, in turning the algebraic structure easier.

For modeling the attitude of the quadrotor the structure has been considered to be rigid and symmetrical, the centre of gravity and the body fixed frame origin coincide, the propellers are stiff, and only the differential forces caused by the propellers have an influence on rotation.

It is known in the literature the orientation of two reference frames may be described as a single rotation θ , along a unitary axis $\vec{e} \in \mathbb{R}^3$, such rotation is encoded by a unit quaternion and can be described as

$$q = \begin{pmatrix} q_0 \\ \bar{q} \end{pmatrix} = \begin{pmatrix} \cos \theta/2 \\ \vec{e} \cdot \sin \theta/2 \end{pmatrix} \quad (3.1)$$

Remark. As it's certified that (q_0, \bar{q}) and $(-q_0, -\bar{q})$ refer to the same physical point. Therefore, q_0 is chosen to be always positive ($0 \leq q_0 \leq 1$).

In the following, q_i^j signifies the quaternion holding the rotation (θ, \vec{e}) that makes the reference frame i coinciding with the frame j when applied to it.

In order to study the movement of a quadcopter two frames have to be defined:

- the inertial frame (I-frame) $E:(O_e; x_e, y_e, z_e)$ which can be treated as an inertial reference where x_e points to the north, y_e points to the East and z_e points upwards.
- the body-fixed frame (B-frame) $B:(O_b; x_b, y_b, z_b)$ which is fixed to the quadrotor body, with origin O_b placed in its centre of gravity (CoG), by convention x_b is the longitudinal

axis pointing towards the front of the vehicle, y_b defines the lateral axis and points to the right of the vehicle and z_b defines the vertical axis of the vehicle and points upwards. Because the body-fixed frame translates and rotates with quadrotor, its position and attitude relative to the Earth-fixed frame determines the flight position and attitude of the quadrotor [34], both frames are depicted in figure 3.7

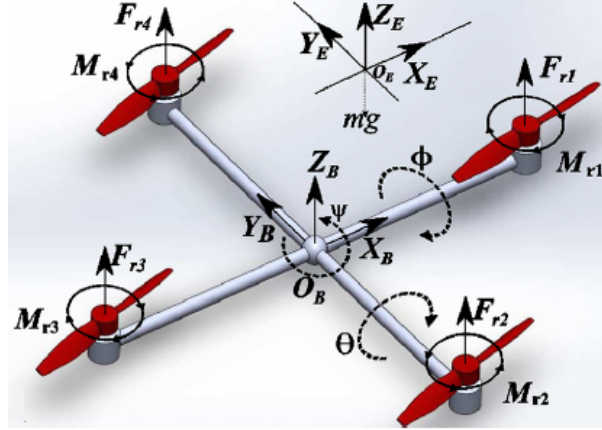


Figure 3.7: frames used to describe the movement of the drone

3.3.1 Kinematic Model :

The quadrotor kinematics can be described with the quaternion \dot{q}_I^B , (3.2) that represent the rotation from the inertial frame I to the body-fixed frame B :

$$\dot{q}_I^B = \frac{1}{2} q_I^B \otimes w_b \quad (3.2)$$

where $w_b = (0, \Omega_b)^T \in \mathbb{R}^4$ is a quaternion associated to the body angular velocity Ω_b . The quaternion product \otimes can be expressed in the following matrix form :

$$q \otimes w_b = \begin{pmatrix} -\bar{q}^{-T} \\ q_0 \cdot I_{3 \times 3} + S(\bar{q}) \end{pmatrix} \Omega \quad (3.3)$$

Where $S(\cdot)$ is the skew-symmetric operator :

$$S(\bar{q}) = \begin{pmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{pmatrix} \quad (3.4)$$

3.3.2 Dynamic Model :

Newton's second law can describe the QUAUV rotational dynamics, which is given in its simplest form by [35]

$$\dot{\Omega} = I_{cm}^{-1} \cdot \tau(t) \quad (3.5)$$

Where $\Omega \in \mathbb{R}^3$ is the angular velocity, $\tau \in \mathbb{R}^3$ is the total torque acting on the vehicle and I_{cm} is the inertia matrix, which is considered to be diagonal.

The applied forces on the QUAUV

The weight of the quadrotor Known as :

$$P = mg \quad (3.6)$$

Where m is the total mass of the QUAUV and g is the acceleration's gravity .

the lift force Lift is the component of a force that is exerted on an object by a fluid flowing around it. It is perpendicular to the oncoming flow direction. Lift conventionally acts in an upward direction in order to counter the force of gravity, but it can act in any direction at right angles to the flow. It is described by the sum of the thrust forces generated by each rotor :

$$F_l = b(w_{r_1}^2 + w_{r_2}^2 + w_{r_3}^2 + w_{r_4}^2) \quad (3.7)$$

Where b is the lift coefficient, it depends on the shape and number of the blades and the density of the air. For that we use the rotation matrix R :

$$F_l = b(w_{r_1}^2 + w_{r_2}^2 + w_{r_3}^2 + w_{r_4}^2)R \quad (3.8)$$

The drag force The drag force is a force that acts in the opposite direction of the relative motion of any object moving in relation to a fluid. This force mainly depends on velocity. It's expressed by :

$$F_d = K_d \cdot v \quad (3.9)$$

Where K_d is a constant matrix called aerodynamic translation coefficient, and v is the QUAUV velocity.

Other forces called disturbance forces, can affects our system :

The Coriolis force : It is an inertial force acting perpendicularly on the direction of motion of a moving body in a uniform rotating reference frame.

Ground effect : This effect is the consequence of the interaction between the earth and the air flow circulating through the blades. The ground effect increases as the drone gets closer to the ground.

According to Newton's second law :

$$\frac{d(m \cdot v)}{dt} = \Sigma F_{outside}$$

and (3.6), (3.6), (3.6) we get :

$$m \cdot \dot{v} = P + F_l + F_d + F_{dist} \quad (3.10)$$

The applied torques on the QUAV

The thrust torque The combined impact of the gear ratio with our propeller converts the torque provided by the engine into thrust. It is expressed by a relation between the lift force F_l , and l the distance from the propeller's center to the gravity center of the QUAV.

Rotation around the x-axis: It is due to the difference in thrust forces between rotors r_2 and rotor r_4 . This moment is given by the following relation:

$$M_x = l(F_{r_4} - F_{r_2}) = lb(w_{r_4}^2 - w_{r_2}^2) \quad (3.11)$$

Rotation around the y-axis: It is due to the difference in thrust forces between rotors r_1 and rotor r_3 . This moment is given by the following relation:

$$M_y = l(F_{r_1} - F_{r_3}) = lb(w_{r_1}^2 - w_{r_3}^2) \quad (3.12)$$

The drag torque The combined impact of the gear ratio with our propeller converts the torque provided by the engine into thrust. It is expressed by a relation between the lift force F_l , and l the distance from the propeller's center to the gravity center of the QUAV.

Rotation around the z-axis: It is due to the drag torques created by each propeller. This moment is given by the following relation :

$$M_z = d(w_{r_1}^2 - w_{r_2}^2 - w_{r_3}^2 + w_{r_4}^2) \quad (3.13)$$

where d is the drag coefficient, it depends on the construction of the propellers.

Torque resulting from aerodynamic frictions: it is the moment resulting from aerodynamic friction, it is determined by :

$$M_{aero} = K_{aero}\Omega^2 \quad (3.14)$$

where K_{aero} is the coefficient of aerodynamic friction and Ω is the angular velocity of the QUAV.

The gyroscopic torque The simultaneous rotation around two perpendicular axes generates a third rotation around the axis perpendicular to the previous two. This is a disturbance moment that induces unwanted rotations of the machine. Our system contains two types of gyroscopic torques :

Gyroscopic torque of the propellers :

$$M_{gh} = \sum_{i=1}^4 \Omega \wedge I_r \begin{pmatrix} 0 \\ 0 \\ (-1)^{i+1} w_{r_i} \end{pmatrix} \quad (3.15)$$

where I_r is the rotors inertia matix

Gyroscopic torque due to the QUAUV motion :

$$M_{gm} = \Omega \wedge I_{cm} \Omega \quad (3.16)$$

According to Newton's second law :

$$\frac{d(I_{cm} \Omega)}{dt} = \sum M_{outside} = \tau(t)$$

we get :

$$\tau = M_x + M_y + M_z - M_{gm} - M_{gh} - M_{aero} \quad (3.17)$$

Translational motion equations

Transnational motion can be described by the following equations :

$$\begin{cases} \dot{r} = v \\ m \cdot \dot{v} = P + F_l + F_d + F_{dist} \end{cases} \quad (3.18)$$

Reformulating the forces equations, for the adaptation with our system :

The weight force :

$$P = \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} \quad (3.19)$$

The lift force :

$$F_l = R \cdot \begin{pmatrix} 0 \\ 0 \\ \sum_{i=1}^4 F_i \end{pmatrix} \quad (3.20)$$

with $F_i = b w_{r_i}$

The drag force :

$$F_d = \begin{pmatrix} -K_{d_x} & 0 & 0 \\ 0 & -K_{d_y} & 0 \\ 0 & 0 & -K_{d_z} \end{pmatrix} \dot{r} \quad (3.21)$$

Replacing those equations in the system 3.18 we get :

$$m \begin{pmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{pmatrix} = R(q) \begin{pmatrix} 0 \\ 0 \\ \sum_{i=1}^4 F_i \end{pmatrix} - \begin{pmatrix} K_{d_x} \\ K_{d_y} \\ K_{d_z} \end{pmatrix} \begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} \quad (3.22)$$

with $R(q)$ is the rotation matrix :

$$R(q) = [R_x(q) \quad R_y(q) \quad R_z(q)] = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_3q_2 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_3q_2 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix}$$

After that we obtain the differential equations that define the translational motion :

$$\begin{cases} m\ddot{x} = 2(q_1q_3 + q_0q_2)(\sum_{i=1}^4 F_i) - K_{d_x}\dot{x} \\ m\ddot{y} = 2(q_3q_2 - q_0q_1)(\sum_{i=1}^4 F_i) - K_{d_y}\dot{y} \\ m\ddot{z} = q_0^2 - q_1^2 - q_2^2 + q_3^2(\sum_{i=1}^4 F_i) - K_{d_z}\dot{z} \end{cases} \quad (3.23)$$

Rotational motion equations

Rotational motion can be described by the following equations :

$$I_{cm}\dot{\Omega} = \Omega \wedge I_{cm}\Omega + M_{xyz} - M_{aero} - M_{gh} \quad (3.24)$$

Reformulating the torques equations, for the adaptation with our system :

Thrust and drag torque:

$$M_{xyz} = \begin{pmatrix} l(F_{r_4} - F_{r_2}) \\ l(F_{r_1} - F_{r_3}) \\ d(w_{r_1}^2 - w_{r_2}^2 - w_{r_3}^2 + w_{r_4}^2) \end{pmatrix} \quad (3.25)$$

Gyroscopic torque:

$$M_{gh} = \begin{pmatrix} I_r w_r \dot{\theta} \\ I_r w_r \dot{\phi} \\ 0 \end{pmatrix} \quad (3.26)$$

with $w_r = w_{r_1} - w_{r_2} - w_{r_3} + w_{r_4}$

Torque resulting from aerodynamic frictions:

$$M_{gh} = \begin{pmatrix} K_{r_x} \dot{\phi}^2 \\ K_{r_y} \dot{\theta}^2 \\ K_{r_z} \dot{\psi}^2 \end{pmatrix} \quad (3.27)$$

Replacing those equations in the system 3.24 we get :

$$\begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \wedge \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} - \begin{bmatrix} K_{r_x} \dot{\phi}^2 \\ K_{r_y} \dot{\theta}^2 \\ K_{r_z} \dot{\psi}^2 \end{bmatrix} - \begin{bmatrix} I_r w_r \dot{\theta} \\ I_r w_r \dot{\phi} \\ 0 \end{bmatrix} + \begin{bmatrix} lb(w_{r_4}^2 - w_{r_2}^2) \\ lb(w_{r_1}^2 - w_{r_3}^2) \\ d(w_{r_1}^2 - w_{r_2}^2 - w_{r_3}^2 + w_{r_4}^2) \end{bmatrix}$$

This equation leads us to differential equations that define the rotational motion :

$$\begin{cases} I_x \ddot{\phi} = \dot{\theta} \dot{\psi} (I_z - I_y) - K_{r_x} \dot{\phi}^2 - I_r w_r \dot{\theta} + lb(w_{r_4}^2 - w_{r_2}^2) \\ I_y \ddot{\theta} = \dot{\phi} \dot{\psi} (I_z - I_x) - K_{r_y} \dot{\theta}^2 - I_r w_r \dot{\phi} + lb(w_{r_1}^2 - w_{r_3}^2) \\ I_z \ddot{\psi} = \dot{\phi} \dot{\theta} (I_y - I_x) - K_{r_z} \dot{\psi}^2 + d(w_{r_1}^2 - w_{r_2}^2 - w_{r_3}^2 + w_{r_4}^2) \end{cases} \quad (3.28)$$

According to the equations that define the translation and rotation of the system, we can describe the dynamics of the QUAUV as follows :

$$\begin{cases} \ddot{\phi} = \dot{\theta} \dot{\psi} \frac{(I_z - I_y)}{I_x} - \frac{K_{r_x}}{I_x} \dot{\phi}^2 - \frac{I_r w_r}{I_x} \dot{\theta} + \frac{l}{I_x} u_2 \\ \ddot{\theta} = \dot{\phi} \dot{\psi} \frac{(I_z - I_x)}{I_y} - \frac{K_{r_y}}{I_y} \dot{\theta}^2 - \frac{I_r w_r}{I_y} \dot{\phi} + \frac{l}{I_y} u_3 \\ \ddot{\psi} = \dot{\phi} \dot{\theta} \frac{(I_y - I_x)}{I_z} - \frac{K_{r_z}}{I_z} \dot{\psi}^2 + \frac{1}{I_z} u_4 \\ \ddot{x} = \frac{1}{m} u_x u_1 - \frac{K_{d_x}}{m} \dot{x} \\ \ddot{y} = \frac{1}{m} u_y u_1 - \frac{K_{d_y}}{m} \dot{y} \\ \ddot{z} = \frac{1}{m} u_z u_1 - \frac{K_{d_z}}{m} \dot{z} \end{cases}$$

with

$$u_x = 2(q_1 q_3 + q_0 q_2)$$

$$u_y = 2(q_3 q_2 - q_0 q_1)$$

$$u_z = q_0^2 - q_1^2 - q_2^2 + q_3^2$$

We define the command vector U :

$$U = [u_1 \quad u_2 \quad u_3 \quad u_4]$$

where

$$u_1 = b(w_{r_1}^2 + w_{r_2}^2 + w_{r_3}^2 + w_{r_4}^2)$$

$$u_2 = b(w_{r_4}^2 - w_{r_2}^2)$$

$$u_3 = b(w_{r_1}^2 - w_{r_3}^2)$$

$$u_4 = d(w_{r_1} - w_{r_2} - w_{r_3} + w_{r_4})$$

We can rewrite those equations in a matrix form :

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -b & 0 & 0 \\ b & 0 & -b & 0 \\ d & -d & -d & d \end{bmatrix} \begin{bmatrix} w_{r_1}^2 \\ w_{r_2}^2 \\ w_{r_3}^2 \\ w_{r_4}^2 \end{bmatrix} \quad (3.29)$$

3.3.3 State space representation

We take the following vector as a state vector :

$$\begin{aligned} \xi &= \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} \\ x_{11} & x_{12} \\ \phi & \dot{\phi} & \theta & \dot{\theta} & \psi & \dot{\psi} & x & \dot{x} & y & \dot{y} \\ z & \dot{z} \end{bmatrix} \\ &= \begin{bmatrix} \phi & \dot{\phi} & \theta & \dot{\theta} & \psi & \dot{\psi} & x & \dot{x} & y & \dot{y} \\ z & \dot{z} \end{bmatrix} \end{aligned}$$

That leads us to the following state space representation :

$$f(\xi, U) = \dot{\xi}$$

and :

$$f(\xi, U) = \begin{bmatrix} x_2 \\ a_1 x_4 x_6 + a_2 w_r x_4 + b_1 u_2 - c_1 x_2^2 \\ x_4 \\ a_3 x_2 x_6 + a_4 w_r x_2 + b_2 u_3 - c_2 x_4^2 \\ x_2 \\ a_5 x_2 x_4 + b_3 u_4 - c_3 x_6^2 \\ x_8 \\ \frac{1}{m} u_x u_1 - c_4 x_8 \\ x_{10} \\ \frac{1}{m} u_y u_1 - c_5 x_{10} \\ x_{12} \\ \frac{1}{m} u_z u_1 - c_6 x_{12} \end{bmatrix}$$

with

$$\begin{aligned} a_1 &= \frac{(I_z - I_y)}{I_x}; a_2 = \frac{(I_r)}{I_x}; a_3 = \frac{(I_z - I_x)}{I_y}; a_4 = \frac{(I_r)}{I_y}; a_5 = \frac{(I_y - I_x)}{I_z} \\ b_1 &= \frac{l}{I_x}; b_2 = \frac{l}{I_y}; b_3 = \frac{l}{I_z} \end{aligned}$$

$$c_1 = \frac{K_{r_x}}{I_x}; c_2 = \frac{K_{r_y}}{I_y}; c_3 = \frac{K_{r_z}}{I_z}; c_4 = \frac{K_{d_x}}{m}; c_5 = \frac{K_{d_y}}{m}; c_6 = \frac{K_{d_z}}{m};$$

3.4 Proposed control scheme

To control the quadrotor attitude a control scheme consists of a cascade connection between two loops is proposed, where the outer loop also known as kinematic loop which takes as input the output of the dynamic loop and ensure the trajectory tracking where the vehicle orientation converge to the desire orientation, and generates a reference angular velocity that must be tracked by the dynamic loop which in its turn fulfils this task due to the controller used which is the active disturbance rejection control. The two control loops are depicted in figure(3.8) and developed next.

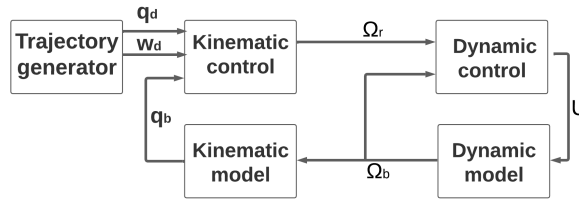


Figure 3.8: proposed control scheme

The dynamic loop must be faster (with a larger bandwidth) than the kinematic loop in order to make the connection possible between the two loops. The control parameters for that purpose must then be chosen.

3.4.1 Kinematics loop

The outer loop comprises essentially two elements that are the kinematic model which provides the quaternion q_I^B noted q_b , representing the orientation of the vehicle, based on the angular velocity Ω_b coming from the dynamic loop and satisfying:

$$\dot{q}_I^B = \dot{q}_b = \frac{1}{2} q_b \otimes w_b \quad (3.30)$$

where $w_b = (0; \Omega_b)^T$.

The second part of the outer loop is the kinematic control that guarantees the trajectory tracking. This outer loop is presented in the following scheme.

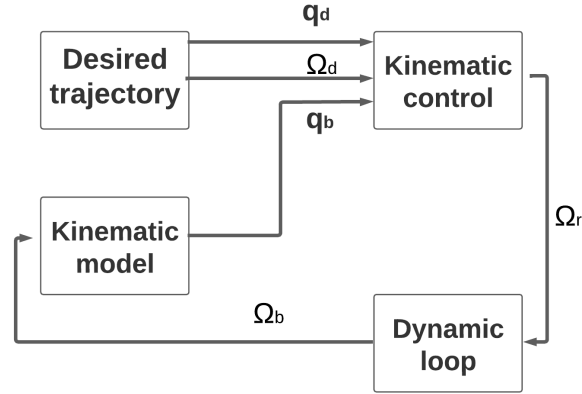


Figure 3.9: Kinematic loop scheme

Problem

Given a reference frame, B , in an arbitrary initial orientation, which can be rotated with an angular velocity Ω_b satisfying (3.30), and the desired orientation of the vehicle is given by a feasible trajectory q_I^D (noted q_d) of a virtual desired reference frame D , which also must satisfy:

$$\dot{q}_I^D = \dot{q}_d = \frac{1}{2} q_d \otimes w_d \quad (3.31)$$

where $w_d = (0, \Omega_d)^T$, and Ω_d is the angular velocity of the desired reference frame. find a control law $\Omega_b(q_b, q_d, \Omega_d)$ such that the error between q_d and q_b noted q_e is driven to $q_e = (1, 0, 0, 0)$.

q_e known also as q_D^B is defined as the relative orientation error between B and D, and can be obtained as:

$$q_D^B = (q_I^D)^* \otimes q_I^B = \begin{pmatrix} q_{e,0} \\ \bar{q}_e \end{pmatrix} \quad (3.32)$$

Differentiating (3.32) leads to:

$$\dot{q}_D^B = \frac{1}{2} q_D^B \otimes \tilde{w} \quad (3.33)$$

with $\tilde{w} = (0, \tilde{\Omega}) = w_b - (q_D^B)^* \otimes w_d \otimes q_D^B$

As a solution for the problem above and to ensure a trajectory tracking where q_b converge to q_d i.e. q_e tends to $[1 \ 0 \ 0 \ 0]^T$ a non-linear quaternion-based control law is designed. It takes q_d , w_d , q_b as inputs, and generates a reference angular velocity, Ω_r , for the dynamic control system. this angular velocity is the one that make the vehicle behave as we want him to do.

The pure attitude kinematic control problem neglects the system dynamics, it is assumed that Ω_b tracks Ω_r instantly ($\Omega_b = \Omega_r$).

Knowing q_b, q_d and ω_d (or \dot{q}_d), the kinematic control law is given as [36] :

$$\begin{aligned} w'_d &= (q_e)^* \otimes w_d \otimes q_e = (0, \Omega'_d)^T \\ \Omega_b &= -K_k \bar{q}_e + \Omega'_d \end{aligned} \quad (3.34)$$

w'_d is obtained from the differentiation of equation (3.32) and represents the transformation of the vector \vec{w}_d from the reference frame D to the body fixed frame B . Using the controller presented in (3.34), the attitude error q_e converge exponentially to $(1, 0, 0, 0)$

Proof. [36]

As $w_b = (0, \Omega_b)^T$ and $w'_d = (q_e)^* \otimes w_d \otimes q_e = (0, \Omega'_d)^T$, then $w_b - w'_d = (0, \Omega_b - \Omega'_d)^T = (0, \tilde{\Omega})^T$.

Therefore $\tilde{\Omega} = \Omega_b - \Omega'_d$. In order to demonstrate that q_e converges to $[1 \ 0 \ 0 \ 0]^T$ the following Lyapunov candidate function is considered :

$$V = (1 - q_{e,0})^2 + (\bar{q}_e)^T \bar{q}_e \geq 0 \quad (3.35)$$

Where $V = 0$ if, and only if, $q_{e,0} = 1$ and $\bar{q}_e = 0$. With the quaternion product matrix representation(3.4), and model (3.33) its time derivative takes the form:

$$\begin{aligned} \dot{V} &= (1 - q_{e,0})(\bar{q}_e)^T \tilde{\Omega} + (\bar{q}_e)^T (q_{e,0} I_3 + S(\bar{q}_e)) \tilde{\Omega} \\ \dot{V} &= (\bar{q}_e)^T \tilde{\Omega} \end{aligned}$$

Taking $\tilde{\Omega} = -K_k \bar{q}_e$:

$$\dot{V} = -K_k (\bar{q}_e)^T \bar{q}_e \leq 0 \quad (3.36)$$

So under $\tilde{\Omega} = -K_k \bar{q}_e \Leftrightarrow \omega_b = -K_k \bar{q}_e + \Omega'_d$, it can be concluded that the system is asymptotically stable. Now, lets prove the exponential convergence. Since $0 \leq q_{e,0} \leq 1$ and $\|q\| = 1$, V and \dot{V} can be expressed as :

$$V = 2(1 - q_{e,0}) \leq 2 \quad (3.37)$$

$$\dot{V} = -K_k(1 - q_{e,0}^2) \quad (3.38)$$

With equation (3.37),(3.38) the Lyapunov function can be written in the form :

$$\dot{V} = -K_k V + K_k \frac{V^2}{4} = -K_k V + g(V)$$

$g(V)$ is upper bounded by $(K_k/2)V$ in the region $V \leq 2$, so:

$$\dot{V} = -K_k V + g(V) \leq -K_k V + \frac{-K_k}{2} V \leq \frac{-K_k}{2} V$$

Therefore

$$V(t) \leq V(0) \cdot e^{-\left(\frac{-K_k}{2}\right)t} \quad (3.39)$$

and the exponential convergence is proved. \square

3.4.2 Dynamic loop

The inner loop controller has been proposed to keep track of the target angular velocity w_r , delivered by the outer loop, where it takes as input the reference angular velocity w_r , also the measured w_b and generates the command u (torques), as illustrated in the figure (3.10), that force the dynamic model to follow the reference velocity which will lead to track the desired trajectory, but the dynamics of UAV is very complex and presents non linearities and uncertainties in its model also external disturbances which can come into play, to deal with these problems and difficulties several control laws have been adopted, However, by examining the current solutions for quadrotor control, the following limitations can be identified. PID controllers' tolerance to model uncertainty and unknown disturbances are frequently limited. In addition, the LQR and MPC approaches are potential choices, but their effectiveness is greatly dependent on the correctness of the plant model. Furthermore, robust control such as sliding-mode control and backstepping provide a solution for reducing the effect of external disturbances, but their complex nature makes them difficult to put into reality.

The indicated control methods for quadrotors may not demonstrate a suitable level of resilience and adaptability in circumstances of highly nonlinear and uncertain quadrotor models, strong couplings, and influence of variable disturbances. As a result, a robust control method based on active disturbance rejection control (ADRC) can be employed to handle the difficult challenge of quadrotor control. [34]

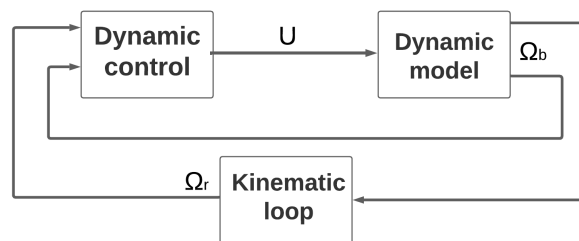


Figure 3.10: Dynamic loop scheme

In what follows the Active Disturbance rejection control is explained in depth.

Active Disturbance Rejection Control

ADRC was established by J. Han in 1998 for the aim of enriching the essence of PID control and sheds its limitations. The concept behind ADRC is to estimate in real time the total disturbance, which lumps parameter uncertainty, nonlinear dynamics, and external disturbances, using a dedicated observer [extended state observer (ESO)] then mitigates this total disturbance in the feedback control inputs. without the needs for a rigorous

mathematical model the control paradigm is shifted from model-centric to control-centric.

Roughly speaking, the ADRC is composed of three parts. The first part is the tracking differentiator (TD) which extracts the derivative of reference signal, note that in the PID controller used in most of industrial control systems, the derivative action “D” is seldom used because of it’s sensitive to high frequency noise. In ADRC, TD serves not only as the derivative extractor, but also as a transient profile that the output of plant can reasonably follow to avoid setpoint jump in PID. The second part, the most important part, is the extended state observer (ESO). As a generalization of the classical state observer in control theory, the ESO provides estimates of both state and total disturbance in terms of output. The last part of ADRC is the TD and ESO based output feedback control which achieve output tracking, which specializes to system stabilization when the reference signal is zero [37].

Tracking differentiator

First Let’s start with introducing the TD.

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a locally Lipschitz continuous function, with $f(0) = 0$. Suppose that the zero-equilibrium state of the following reference-free system is globally asymptotically stable:

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = x_3(t) \\ \vdots \\ \dot{x}_n(t) = f(x_1(t), x_2(t), \dots, x_n(t)) \end{cases} \quad (3.40)$$

For any given initial value. If the reference signal $v(t)$ is differentiable and satisfies

$\sup_{t \in [0, \infty)} |v^{(n+1)}(t)| < \infty$, then tracking differentiator is designed as follow [37]:

$$\begin{cases} \dot{z}_{1R}(t) = z_{2R}(t) \\ \dot{z}_{2R}(t) = z_{3R}(t) \\ \vdots \\ \dot{z}_{nR}(t) = R_n f(z_{1R}(t) - v(t), \frac{z_{2R}(t)}{R}, \dots, \frac{z_{nR}(t)}{R^{(n-1)}}) \end{cases} \quad (3.41)$$

TD converges in the sense that for every $a > 0$, $\lim_{R \rightarrow \infty} |z_{1R} - v(t)| = 0$ uniformly on $[a, \infty)$. This result was first proved in [38].

in control practice, The setpoint $v(t)$ is frequently presented as a step function, which is inappropriate for most dynamics systems because it asks the output, and therefore the control signal, to make an abrupt jump [39]. Tracking differentiator comes as a solution to this problem where the output track $z_{1R}(t)$ instead of $v(t)$. Other variables produced from TD (3.41) are considered as the derivatives of $v(t) : Z_{iR}(t) \approx v^{(i-1)}(t)$ in the sense of generalized derivative.

Extended State Observer

An ESO is an observer that can estimate the system's uncertainties as well as its states, allowing disturbance rejection or compensation. The ESO considers all system elements, including parameter uncertainty, nonlinear dynamics, and external disturbances, as the total disturbance and treat it as an extended state; this is why it is named extended state observer. Its advantage consists in the independence of the system's mathematical model, and the good performance and simplicity of the implantation.[40] The idea of ESO can

be demonstrated in the following single-input and single-output system:

$$\begin{cases} \dot{x}^{(n)}(t) &= f(x^{(n-1)}(t), x^{(n-2)}(t), \dots, x(t), d(t), t) + bu(t) \\ y &= x(t) \end{cases} \quad (3.42)$$

where n is the order of the plant, y is the output, u is the input, b is a constant, $d(t)$ is the external disturbance, $f(\cdot)$ is an unknown function which can be viewed as the total uncertainties or disturbances of the system, both internal and external. Introduce $h = \frac{df}{dt}$, if the function f is nonsmooth, h denotes the generalized derivative of $f(\cdot)$.

The extended form of the system (3.42), where the uncertainty f as an extended state and noted as X_{n+1} , is given as:

$$\begin{cases} \dot{x}_1(t) &= x_2(t) \\ &\vdots \\ \dot{x}_{n-1}(t) &= x_n(t) \\ \dot{x}_n(t) &= x_{n+1}(t) + bu(t) \\ \dot{x}_{n+1}(t) &= h \end{cases} \quad (3.43)$$

The ESO that estimate both the states and total disturbance (the extended state) is defined as follow [41]:

$$\begin{cases} \dot{z}_1(t) &= z_2(t) - \beta_1 fal(e_1, \alpha_1, \delta) \\ &\vdots \\ \dot{z}_{n-1}(t) &= z_n(t) - \beta_{n-1} fal(e_1, \alpha_{n-1}, \delta) \\ \dot{z}_n(t) &= z_{n+1}(t) - \beta_n fal(e_1, \alpha_n, \delta) + bu \\ \dot{z}_{n+1}(t) &= -\beta_{n+1} fal(e_1, \alpha_{n+1}, \delta) \end{cases} \quad (3.44)$$

Where $e_1 = z_1 - x_1$ the observer error and β_i ($i = 1, \dots, n + 1$) the gain observer.

The function fal is defined as follow:

$$fal(e, \alpha, \delta) = \begin{cases} |e|^\alpha sign(e), & |e| > \delta \\ e/\delta^{1-\alpha}, & otherwise \end{cases} \quad (3.45)$$

Where $0 \leq \alpha \leq 1$ and $\delta > 0$

ESO (3.44) is designed to have the property $z_i(t) \rightarrow x_i(t)$ ($i = 1, \dots, n + 1$), it should be noted that when $\alpha_i = 1$ ($i = 1, \dots, n + 1$), (3.44) take the form of the classical Luenberger Observer. On the other hand when $\alpha_i = 0$ (3.44) is consistent with the sliding mode observer.

The linear form of ESO which is the linear extended state observer (LESO) is given as :

$$\begin{cases} \dot{z}_1(t) = z_2(t) - \beta_1 e_1 \\ \vdots \\ \dot{z}_{n-1}(t) = z_n(t) - \beta_{n-1} e_1 \\ \dot{z}_n(t) = z_{n+1}(t) - \beta_n e_1 + bu \\ \dot{z}_{n+1}(t) = -\beta_{n+1} e_1 \end{cases} \quad (3.46)$$

In [41] it was proved that if f is differentiable with respect to t and $h = \dot{f}$ is bounded, then the LESO can estimate $f(t)$ with bounded error.

Since $y = x_1$ is available, only the estimation of $x_i, i \geq 2$ are needed. Hence, we will use the reduced-order ESO [42] :

$$\begin{cases} \dot{z}_2 = \begin{cases} -\beta_1 z_2 - \beta_1^2 x_1 - \beta_1 bu(t), & \text{if } n = 1, \\ -\beta_1 z_2 + z_3 + (\beta_2 - \beta_1^2) x_1, & \text{if } n > 1 \end{cases} \\ \dot{z}_3(t) = -\beta_2 z_2 + z_4 + (\beta_2 - \beta_1 \beta_3) x_1 \\ \vdots \\ \dot{z}_n(t) = -\beta_{n-1} z_2 + z_{n+1} + (\beta_n - \beta_1 \beta_{n-1}) x_1 + bu(t) \\ \dot{z}_{n+1}(t) = -\beta_n z_2 - \beta_1 \beta_n x_1 \\ \hat{x}_i = z_i + \beta_{i-1} x_1, i = 2, \dots, n+1 \end{cases} \quad (3.47)$$

The characteristic polynomial of (3.47) is given as :

$$s^n + \beta_1 s^{n-1} + \dots + \beta_n = s^n + \sum_{i=1}^n \beta_i s^{n-i} \quad (3.48)$$

Proof. To prove the precedent result we assume that $n > 1$ in (3.47), taking the first line $\dot{z}_2 = -\beta_1 z_2 + z_3 + (\beta_2 - \beta_1^2) x_1$, differentiating \dot{z}_2 leads to:

$$z_2^{(2)} = -\beta_1 \dot{z}_2 + \dot{z}_3 + (\beta_2 - \beta_1^2) \dot{x}_1$$

$$z_2^{(2)} = -\beta_1 \dot{z}_2 + \dot{z}_3 + f(\dot{x}_1)$$

Replacing \dot{z}_3 with its expression from (3.47) gives :

$$z_2^{(2)} = -\beta_1 \dot{z}_2 - \beta_2 z_2 + z_4 + f(\ddot{x}_1, \dot{x}_1)$$

Differentiation \dot{z}_2 for the second time and replacing \dot{z}_4 with its expression from (3.47):

$$z_2^{(3)} = -\beta_1 z_2^{(2)} - \beta_2 \dot{z}_2 - \beta_3 z_2 + z_5 + f(x_1^{(3)}, \ddot{x}_1, \dot{x}_1)$$

Repeating the same procedure n times, i.e differentiates \dot{z}_2 i times and in each time we derive \dot{z}_2 we replace \dot{z}_{i+1} by its expression, we obtain :

$$z_2^{(n)} = -\beta_1 z_2^{(n-1)} - \beta_2 z_2^{(n-2)} - \dots - \beta_n z_2 + f(x_1, \dot{x}_1, \dots, x_1^{(n)}) \quad (3.49)$$

By doing the Laplace transform of the equation (3.49) we get:

$$s^n z_2 = -\beta_1 s^{n-1} z_2 - \beta_2 s^{n-2} z_2 - \dots - \beta_n z_2 + f(x_1)$$

$$z_2(s^n + \beta_1 s^{n-1} + \beta_2 s^{n-2} + \dots + \beta_n) = f(x_1)$$

x_1 is an input to the observer, then

$$\frac{z}{f(x_1)} = \frac{1}{s^n + \beta_1 s^{n-1} + \beta_2 s^{n-2} + \dots + \beta_n} \quad (3.50)$$

□

The parameters β_i are chosen in such way as $s^n + \beta_1 s^{n-1} + \dots + \beta_n = (s + w_0)^n$, where w_0 denotes the bandwidth of the LRESO (3.47).

Note that the inputs to ESO are the system output y and the control signal u , and the output of the ESO provides the important information $f(t)$ representing the total disturbance.

output feedback control

The third and the last link of ADRC is to design an extended state observer-based output feedback control:

$$U(t) = -B^{-1}(\hat{x}_{n+1} + U_0(t, \hat{x}_1, \dots, \hat{x}_n)) \quad (3.51)$$

It is seen that the term \hat{x}_{n+1} in controller (3.51) is used to cancel the total disturbance, and $U_0(t, \hat{x}_1, \dots, \hat{x}_n)$ is the control law which can achieve the satisfactory performance for the canonical system (3.43) [43].

LADRC applied on quadrotor control

Even though the dynamics of uav is completely described in (3.24), it should be noted that none of its terms is accurately known. for example, the dependence of matrix M_{aero} on aerodynamic coefficients that are very difficult to model, the external disturbance $p(t)$. Even the Coriolis term is uncertain because it depends on the inertia matrix which is uncertain too. In the light of the arguments raised above, it is important to use a control that does not depend on the model and that takes into account its uncertainties and unmodeled dynamics, and for this we have chosen the Linear active disturbance rejection control. From (3.24) the quadrotor dynamic model can be written in general form as:

$$\dot{\Omega}_b = d(t) + f(\Omega, t) + Bu(t) \quad (3.52)$$

where $d(t) = -I_{cm}^{-1}p(t)$, $B = -I_{cm}^{-1}M$ and f contains the rest of nonlinear and unknown dynamics.

In purpose of tracking the reference signal $\Omega_r(t)$ and for control design we define the error

model x_1 as $x_1 = \Omega_b - \Omega_r$. Differentiating x_1 leads to:

$$\dot{x}_1 = \dot{\Omega}_b - \dot{\Omega}_r = f(\Omega, t) + Bu(t) + d(t) - \dot{\Omega}_r \quad (3.53)$$

$\dot{\Omega}_r$ is considered as an unknown disturbance and thus $\tilde{d}(t) = d(t) - \dot{\Omega}_r$. (3.53) becomes :

$$\dot{x}_1 = f(\Omega, t) + Bu(t) + d(t) + \tilde{d}(t) \quad (3.54)$$

The model (3.54), which retains the information of the certainly known dynamics, will be used for control design purposes.

First we begin by establishing the extend state observer, Let us define $\bar{B} = \text{diag}(\bar{b}_1; \bar{b}_2; \bar{b}_3)$ as an estimation of B. Since $B; \bar{B}$ are diagonal, the system (3.54) can be seen as three second-order subsystems. The attitude model is rewritten as:

$$\begin{cases} \dot{\Omega}_{bx} = b_2 U_2 + f_1(\Omega, t) + \tilde{d}_x \\ \dot{\Omega}_{by} = b_3 U_3 + f_2(\Omega, t) + \tilde{d}_y \\ \dot{\Omega}_{bz} = b_4 U_4 + f_3(\Omega, t) + \tilde{d}_z \end{cases} \quad (3.55)$$

Therefore, the design of the LADRC can be reduced to a single axis, the results being equal for the other two. For the axis $i = x; y; z$ and including the unknown disturbances as an extended state $x_{2,i}$, the model (3.54) becomes:

$$\begin{cases} \dot{x}_{1,i} = x_{2,i} + \bar{b}_i u_i \\ \dot{x}_{2,i} = h_i(\Omega_{b,i}, u_i, t) \\ y_{m,i} = x_{1,i} = \Omega_{b,i} - \Omega_{r,i} \end{cases} \quad (3.56)$$

Where $x_{1,i} \in \mathbb{R}$, $x_{2,i} = f_i(\Omega_{b,i}, t) + \tilde{d}_i(t) + (b_i - \bar{b}_i)u \in \mathbb{R}$ and $h_i(\Omega_{b,i}, u_i, t)$ the time derivative of f .

As the angular velocity Ω_b is measured and the reference velocity Ω_r is known, $x_{1,i}$ is available and only the estimation of $x_{2,i}$ is needed. So the following linear reduced extended state observer will be used:

$$\begin{cases} \dot{z}_{2,i} = -\beta z_{2,i} - \beta^2 x_{1,i} - \beta \bar{b}_i u_i(t) \\ \hat{x}_{2,i} = z_{2,i} + \beta x_{1,i} \end{cases} \quad (3.57)$$

Since $n=1$ and as it has mentioned above in (3.48), the characteristic polynomial of (3.57) is $(s + \beta)$ being $\beta > 0$ the bandwidth of the RLESO. The initial condition must be chosen to force $\hat{x}_{2,i}(0) = 0$, so $z_{2,i}(0) = -\beta x_{1,i}(0)$.

As the total disturbance has been estimated in (3.57), we move to the third element in ADRC which is the output feedback control, it is designed in such way that the total disturbance is canceled and the error model $x_{1,i}$ converges exponentially to zero. For that aim u_i is given as :

$$u_i = \frac{-k_d x_{1,i} - \hat{x}_{2,i}}{\bar{b}_i} \quad (3.58)$$

where $k_d > 0$ is the feedback control gain.

If we replace (3.58) in (3.56) we get :

$$\dot{x}_{1,i} = -k_d x_{1,i} \implies x_{1,i} = x_{1,i}(0)e^{-k_d t} \quad (3.59)$$

From (3.59) when t is sufficiently big $x_{1,i}$ tends to zero, as $x_1 = \Omega_b - \Omega_r$, $x_{1,i} \rightarrow 0 \implies \Omega_b \rightarrow \Omega_r$. Then using the active disturbance rejection control the convergence of the system dynamics to the desired dynamics is ensured and model uncertainty and external perturbations are rejected.

We give the ADRC scheme bellow (figure 3.11):

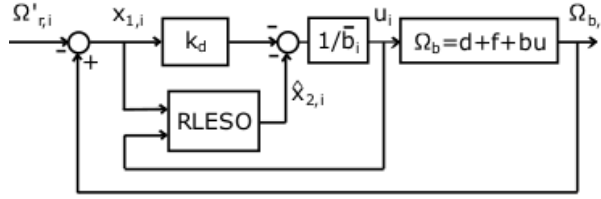


Figure 3.11: Block diagram of the LADRC structure

To avoid overshoot in system response and the setpoint jump i.e to ask a system to do a sudden jump and rather than tracking differentiator (TD) we use a low pass filter given by the following transfer function :

$$F(s) = \frac{10}{s + 10} \quad (3.60)$$

Instead of tracking a step the system will track the output of the low pass filter $y(t)$ calculated as:

$$F(s) = \frac{Y(s)}{U(s)} = \frac{10}{s + 10} \implies Y(s) = \frac{10U(s)}{s + 10}.$$

where U is a step with amplitude A , $U(s) = \frac{A}{s}$.

$$Y(s) = \frac{10A}{s(s + 10)} = \frac{A}{s} - \frac{A}{s + 10}$$

$$y(t) = A - Ae^{-10t} = A(1 - e^{-10t})$$

The term e^{-10t} prevents the output from doing a sudden jump from 0 to A and it tends to 0 making the output $y(t) = A$.

3.5 Conclusion :

A quaternion model for the quadrotor vehicle is presented in this chapter. The usage of quaternions eliminates undesired effects on the system such as the gimbal-lock or discontinuities, which are common problems using traditional approaches. In addition, ADRC control structure has been proposed for a quadrotor. The key contribution is a cascade control with an external loop managing the quaternion-based quadrotor kinematics. The internal loop, constructed utilising ADRC principles, takes into account the quadrotor's

uncertain dynamic model and rejects external disturbances. Overall, it enables quadrotor flight control without singularities.

Chapter 4

Quaternion attitude estimation

4.1	Introduction	48
4.2	Wahba's problem	48
4.3	Wahba's problem solutions	49
4.4	Conclusion	53

4.1 Introduction

Numerous spacecraft missions need the capacity to estimate the spacecraft's attitude, and many methods for estimating attitude based on sensor data exist. Sensor data from a magnetometer, a sun sensor, a star camera, a horizon-crossing indication, and a three-axis rate gyro, for example, can be employed (As we mentioned in the first chapter). To process the data, many algorithms may be utilised. All three-axis attitude estimation issues are nonlinear by definition. There are several strategies for dealing with generic nonlinear estimates as well as the specific non-linearities connected with attitude estimation. However, there is no one algorithm that is guaranteed to find the globally optimal attitude estimate for all possible nonlinear attitude determination problems. There is a special attitude determination problem for which a globally optimal nonlinear solution exists: Wahba's problem [44].

Instead of filtering approaches that use knowledge about spacecraft dynamics, this study will look at algorithms for predicting spacecraft attitude using vector measurements acquired at a single moment, sometimes known as "single-frame" or "point" methods. Almost all single-frame algorithms are based on a problem that Grace Wahba proposed in 1965 [45, 46].

4.2 Wahba's problem

The Wahba problem, seeks the proper orthogonal matrix (attitude matrix) A with determinant $+1$ which minimizes the loss function [47] :

$$L(A) = \frac{1}{2} \sum_i a_i |b_i - Ar_i|^2 \quad (4.1)$$

where b_i is a set of unit vectors measured in a spacecraft's body frame; r_i are the corresponding unit vectors in a reference frame, and a_i are non-negative weights. In order to relate Wahba's problem to Maximum Likelihood Estimation, the weights are selected to be inverse variances, $a_i = \sigma_i^{-1}$. This departs from Wahba's and many other authors' assumptions, which assumed the weights are normalised to unity. According to [46], the loss function can be written as :

$$L(A) = \lambda_0 - tr(AB^T) \quad (4.2)$$

with $\lambda_0 \equiv \sum_i a_i$
and $B \equiv \sum_i a_i b_i r_i^T$

Consequently, $L(A)$ is minimized when the trace, $tr(AB^T)$, is maximized. This is similar to the orthogonal Procrustes problem, which is to identify the orthogonal matrix A that is closest to B in the Frobenius norm [46].

$$\|M\|_F^2 \equiv \sum_{ij} M_{ij}^2 = \text{tr}(MM^T) \quad (4.3)$$

Now

$$\|A - B\|_F^2 = \|A\|_F^2 + \|B\|_F^2 - 2\text{tr}(AB^T) = 3 + \|B\|_F^2 - 2\text{tr}(AB^T) \quad (4.4)$$

The choice of the A 's determinant equal to +1 has lead Wahba' problem to be equivalent to the orthogonal Procrustes problem.

We present below some Wahba's problem solutions.

4.3 Wahba's problem solutions

Attitude determination methods are roughly divided into two categories: static and dynamic. Static methods are time independent, all measurements (e.g. sun vector, magnetic field vector, etc.) are made simultaneously or close enough in time that spacecraft motion between observations can be ignored or easily compensated. This is a deterministic method that does not require information about past states. The problem then is to resolve the geometry of these measurements in the body frame and compare them with the corresponding known descriptions in the inertial frame.[48]

Dynamic attitude determination methods not only consider motion and are therefore time-dependent, but also they do not treat the measurement as a deterministic process, but as a process in the presence of random noise, which requires statistical methods. Filtering techniques such as the Kalman filter are used to organize information from past and actual measurements, knowledge of spacecraft motion, and possible errors in the system dynamics model. Due to the statistical nature of these methods, the results are best called attitude "estimates" rather than "determinations". [48]

4.3.1 Static Attitude Determination Methods

First Solution of Wahba's Problem

J. L. Farrell and J. C. Stuelpnagel presented the first solutions of Wahba's problem. Farrell and Stuelpnagel noted that any real square matrix, including B , has the polar decomposition

$$B = WR \quad (4.5)$$

R is symmetric and positive semidefinite, while W is orthogonal. R may then be diagonalized by

$$R = VDV^T \quad (4.6)$$

where V is orthogonal and D is diagonal with elements arranged in decreasing order. The

optimal attitude estimate is then given by

$$A_{opt} = WV \text{diag} [1 \quad 1 \quad \det W] V^T \quad (4.7)$$

In most cases, $\det W$ is positive and $A_{opt} = W$, but this is not guaranteed. For that R. H. Wessner proposed the alternative solution

$$A_{opt} = (B^T)^{-1}(B^T B)^{1/2} = B(B^T B)^{-1/2} \quad (4.8)$$

This equation requires the B matrix to be non-singular, this implies that at least three vectors must be observed, however it is commonly known that determining the attitude requires only two vectors.

Davenport's q Method [46]

Davenport gave the first practical solution to Wahba's problem of determining spacecraft attitude. He used a unit quaternion to parameterize the attitude matrix. (as mentioned in 2)

As

$$A(q) = (q_s^2 - \|\bar{q}\|^2)I_{3 \times 3} + 2\bar{q}\bar{q}^T - 2q_s S(\bar{q}) \quad (4.9)$$

Since $A(q)$ is a homogeneous quadratic function of q , we can write

$$\text{tr}(AB^T) = q^T K q \quad (4.10)$$

Where K is the symmetric traceless matrix

$$K = \begin{pmatrix} S - I \text{tr}(B) & z \\ z^T & \text{tr}(B) \end{pmatrix} \quad (4.11)$$

with

$$S \equiv B + B^T \quad \text{and} \quad z \equiv \begin{pmatrix} B_{23} - B_{32} \\ B_{31} - B_{13} \\ B_{12} - B_{21} \end{pmatrix} = \sum_i a_i b_i \times r_i$$

The quaternion maximising right side of equation (4.10) represents the optimal attitude. The optimal quaternion is equal to the normalised eigenvector of K with the largest eigenvalue, i.e. the solution of

$$K q_{opt} \equiv \lambda_{max} q_{opt} \quad (4.12)$$

With equations (4.2) and (4.10), we get the optimized loss function

$$L(A) = \lambda_0 - \lambda_{max} \quad (4.13)$$

The eigenvalues of the K matrix, $\lambda_{max} \equiv \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \lambda_4 \equiv \lambda_{min}$, are related to the singular values by [46]

$$\begin{aligned}\lambda_1 &= s_1 + s_2 + s_3, \quad \lambda_2 = s_1 - s_2 - s_3, \\ \lambda_3 &= -s_1 + s_2 - s_3, \quad \lambda_4 = -s_1 - s_2 + s_3\end{aligned}$$

Because K is traceless, the eigenvalues sum to zero. If the two largest eigenvalues of K are equal, or $S_2 + S_3 = 0$, there is no unique solution. This does not indicate a failure of the q method; rather, it indicates that the data are insufficient to establish the attitude uniquely.

The disadvantage of this method is the necessity of calculating the eigenvectors and eigenvalues of a four by four matrix, which may be demanding in terms of computational resources [48].

4.3.2 Dynamic attitude estimation methods

Kalman Filter (KF)

The Kalman Filter (KF) is an iterative algorithm that distinguishes between noise and useful information. KF can be used to obtain better estimates from noisy measurements. For this, the algorithm needs additional information, some from knowledge about the process, for example in the form of equations of motion, and some from other observations/measurements.

One major assumption in KF is that process and measurement uncertainties are white noises, i.e., Gaussian distributions with known covariances and mean values equal to zero. Another important assumption is that the process, transition, or system dynamic model is linear and takes the form [48] :

$$x_{k+1} = \eta_k x_k + \mu_k \tag{4.14}$$

Where x_k is the state vector; η_k is the linear transition matrix and μ_k is the noise vector that represents the uncertainty in the transition model, it has a zero mean value and Q_k as covariance matrix.

$$E[\mu_k] = 0; \quad E[\mu_k \mu_i^T] = \begin{cases} Q_k, & i = k \\ 0, & i \neq k \end{cases}$$

The measurement or observation model is considered to be linear as well, and it may be described as follows:

$$z_{k+1} = H_k x_k + n u_k \tag{4.15}$$

Where z_k is the observation (measurement) vector, H_k is a linear observation matrix that shows the relation between the state vector and the observation performed. μ_k is

the noise vector that represents the uncertainty in the measurement model, it has a zero mean value and R_k as covariance matrix.

$$E[\nu_k] = 0; \quad E[\nu_k \nu_i^T] = \begin{cases} R_k, & i = k \\ 0, & i \neq k \end{cases}$$

Covariances Q_k and R_k are assumed to be known and noises μ_k and ν_k are independent in relation to each other.

The outcome of the KF algorithm is an estimate of the state vector \hat{x}_k for each time step k and its pertaining probability distribution expressed in terms of the estimate covariance matrix P_k .

The estimate covariance matrix, also known as the estimate error covariance matrix P , indicates how reliable the estimate is. It can be described as :

$$\begin{aligned} e_{r_k} &= x_k - \hat{x}_k \\ P_k &= E[e_{r_k} e_{r_k}^T] \end{aligned}$$

Where e_{r_k} is the estimate error.

The Kalman filter has two stages: the propagation or prediction stage, which employs process equations to anticipate the state for the next time step, and the update stage, which trims the estimates using measurement data. The order of these two steps is arbitrary because this is an iterative process. Only the information from the previous time step is required because it is a recursive procedure.

The iterations of the Kalman loop are shown below, along with their corresponding equations.

0. Best initial estimate:

$$\hat{x}_0^- ; P_0^-$$

1. Compute the Kalman gain:

$$K_k = P_0^- H_k^T (H_k P_0^- H_k^T + R_k)^{-1}$$

2. Update estimate with the measurement z_k :

$$\hat{x}_0 = \hat{x}_0^- + K_k (z_k - H_k \hat{x}_0^-)$$

3. Update the error covariance estimate:

$$P_k = (I - K_k H_k) P_0^-$$

4. Project ahead the estimate :

$$\hat{x}_{k+1}^- = \eta_k \hat{x}_k$$

5. Project ahead the error covariance:

$$P_{k+1}^- = \eta_k P_k \eta_k^T + Q_k$$

6. Return to step 1

4.4 Conclusion

In this chapter the cruciality of the attitude estimation in the control loop was presented. This attitude estimation was expressed in a mathematical way under the name of **Wahba's problem**, by Grace Wahba in 1965. This problem has many solutions over the years. In this study, **J. L. Farrell** and **J. C. Stuelpnagel** solution was presented, however this solution has a major drawback, which is singularities. For that **Davenport's q method** has been introduced. both of these two solutions are called **Static Attitude Determination Methods**. On the other hand, the **Kalman Filter**, which is a **Static Attitude Determination Method** has been presented. This latter method is known as one the most utilised algorithms in attitude estimation.

Chapter 5

Simulations Results

- 5.1 Quanser 3 DOF Hover [1] 55
- 5.2 Simulation 57
- 5.3 Application on the Quanser 3 DOF Hover 62
- 5.4 Quaternion attitude estimation 67
- 5.5 Conclusion 70

5.1 Quanser 3 DOF Hover [1]

the quanser 3 DOF is a simulation platform which allows to test the commands developed to pilot a quadrotor and to visualize the performances of this command used, before applying it on a real QUAV.

The 3 DOF Hover consists of a planar round frame with four propellers. The frame is mounted on a three degrees of freedom pivot joint that enables the body to rotate about the roll, pitch and yaw axes. The propellers are driven by four DC motors that are mounted at the vertices of the frame. The propellers generate a lift force that can be used to directly control the pitch and roll angles. Two of the propellers are counter-rotating, so that the total torque in the system is balanced when the thrust of the four propellers is approximately equal.



Figure 5.1: 3 DOF Hover

5.1.1 3 DOF Hover model

The 3 DOF Hover modeling conventions used are:

- The 3 DOF Hover is horizontal (i.e., parallel with the ground) when the pitch and roll angles are zero, $\theta_p = 0$ and $\theta_r = 0$.
- Yaw angle increases positively, $\dot{\theta}_y(t) > 0$ when the body rotates in the counter-clockwise (CCW) direction.
- Pitch angle increases positively, $\dot{\theta}_p > 0$ when rotated CCW.
- roll angle increases positively, $\dot{\theta}_r > 0$ when rotated CCW.

A positive thrust force is generated when a positive voltage is delivered to any motor, which causes the matching propeller assembly to rise. The thrust force generated by

the front, back, right, and left motors are denoted by F_f , F_b , F_r , and F_l , respectively. The front and back motors' thrust forces mostly control motions around the pitch axis, while the right and left motors mainly move the hover around its roll axis. Notice that the pitch angle increases when the thrust force from the front motor is larger than back motor $F_f > F_b$. The roll angle increases when the thrust force from the right motor is larger than the left motor, $F_r > F_l$.

The dynamics for each axis can be described by the general equation:

$$J_i \ddot{\theta}_i = L \Delta F \quad (5.1)$$

where θ_i (i=roll, pitch, yaw) is the angle of the pivot, L is the distance between the propeller motor and the pivot on the axis, J is the moment of inertia about the axis, and ΔF is the differential thrust-force.

The 3 DOF Hover model is given as follow :

$$\begin{cases} J_r \ddot{\theta}_r & = LK_f(V_r - V_l) \\ J_p \ddot{\theta}_p & = LK_f(V_f - V_b) \\ J_y \ddot{\theta}_y & = K_t(V_r + V_l) - K_t(V_f + V_b) \end{cases} \quad (5.2)$$

Where V_f , V_b , V_r , V_l are the front, back, right, left motor voltage respectively and K_f is the thrust-force constant, K_t is the thrust-torque constant.

The state-space representation is given by :

$$\begin{cases} \dot{x} & = Ax + Bu \\ y & = Cx + Du \end{cases} \quad (5.3)$$

we define the state vector : $x^T = [\theta_r \ \theta_p \ \theta_y \ \dot{\theta}_r \ \dot{\theta}_p \ \dot{\theta}_y]$.

and the control vector : $u^T = [V_f \ V_b \ V_r \ V_l]$.

Using the equations of motion given (5.4), the corresponding 3 DOF Hover state-space matrices are:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{LK_f}{J_r} & -\frac{LK_f}{J_r} \\ \frac{LK_f}{J_p} & -\frac{LK_f}{J_p} & 0 & 0 \\ -\frac{K_t}{J_y} & -\frac{K_t}{J_y} & \frac{K_t}{J_y} & \frac{K_t}{J_y} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

in our case we control the drone attitude by controlling the three angular velocities $[\dot{\theta}_r \ \dot{\theta}_p \ \dot{\theta}_y]$. so we will use only the last three rows of the matrix A and B, the state-space representation become : $\dot{x} = B'V$ where $V^T = [V_f \ V_b \ V_r \ V_l]$ and

$$B' = \begin{bmatrix} 0 & 0 & \frac{LK_f}{J_r} & -\frac{LK_f}{J_r} \\ \frac{LK_f}{J_p} & -\frac{LK_f}{J_p} & 0 & 0 \\ -\frac{K_t}{J_y} & -\frac{K_t}{J_y} & \frac{K_t}{J_y} & \frac{K_t}{J_y} \end{bmatrix}$$

The control proposed in the previous section gives as command the three torques u_1, u_2, u_3 which have a proportional relation with the voltages V_f, V_b, V_r, V_l .

$$\begin{cases} u_1 & = LK_f(V_r - V_l) \\ u_2 & = LK_f(V_f - V_b) \\ u_3 & = K_t(V_r + V_l) - K_t(V_f + V_b) \end{cases} \quad (5.4)$$

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = B' \begin{bmatrix} V_f \\ V_b \\ V_r \\ V_l \end{bmatrix}$$

To obtain the voltage V from the torques u we must invers the matrix B', as B' is not a square matrix its pseudo inverse will be calculated :

$$V = (B')^+ u$$

5.2 Simulation

To test the effectiveness of the proposed control scheme, a series of simulations were performed on the Quanser 3DOF Hover model given previously. The simulation was done using Matlab/Simulink.

The system parameters and initial conditions are detailed in the table below:

Name	parameter	Value	Units
Moment of Inertia about x axis	J_r	0.0552	$kg.m^2$
Moment of Inertia about y axis	J_p	0.0552	$kg.m^2$
Moment of Inertia about z axis	J_y	0.1104	$kg.m^2$
Propeller Force-Thrust Constant	K_f	0.1188	N/V
Propeller Torque-Thrust Constant	K_t	0.0036	N.m/V
Mass	m	2.85	Kg
Gravitational Constant	g	9.81	m/s^2

Table 5.1: System parameters

Initial condition:

Initial condition	Value
qb_0	1
qb_1, qb_2, qb_3	0
wb_x, wb_y, wb_z	0

Table 5.2: Initial conditions

The global schema of the simulation is presented in the following figure :

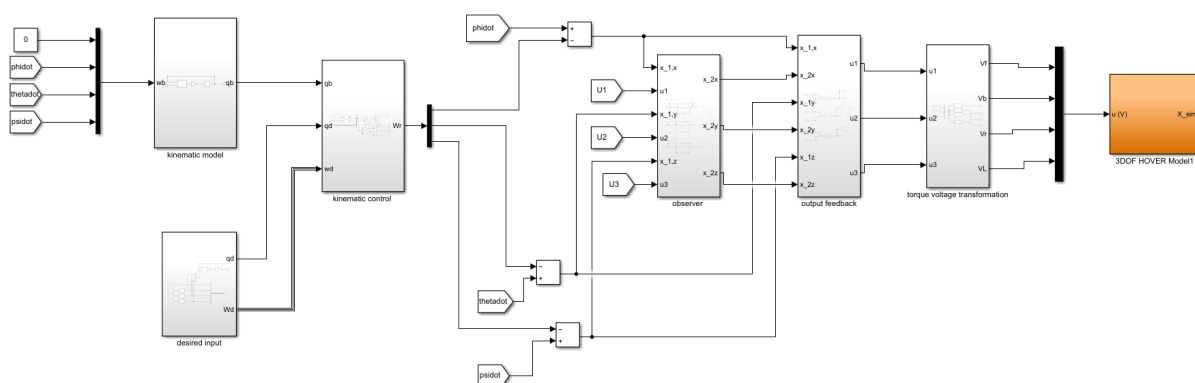


Figure 5.2: global schema representing system and controllers

The controllers and observer parameters are given in the following table :

Parameter	Value
K_d	100
K_k	1
β	1

Table 5.3: Controllers and observer parameters

K_d is chosen larger than K_k to make the connection between the two loops dynamic and kinematic possible, the dynamic loop must be faster than the kinematic. Also the observer gain is chosen with a small value because for biggest values the observer become too sensitive so the smallest perturbation destabilizes the system.

5.2.1 Hovering simulation

To simulate the hovering, the angular velocity of the desired reference frame is given as $\omega_d = (0 \ 0 \ 0)^T$ and $q_d = (1 \ 0 \ 0 \ 0)^T$ which corresponds to a rotational desire angle $\theta_d = 0$ (knowing that $q_d = \begin{bmatrix} \cos(\theta_d/2) \\ \vec{e} \cdot \sin(\theta_d/2) \end{bmatrix}$).

A perturbation is applied on the x axis (roll) at time $t=1s$ and $t=3.5s$ and given in the following form :

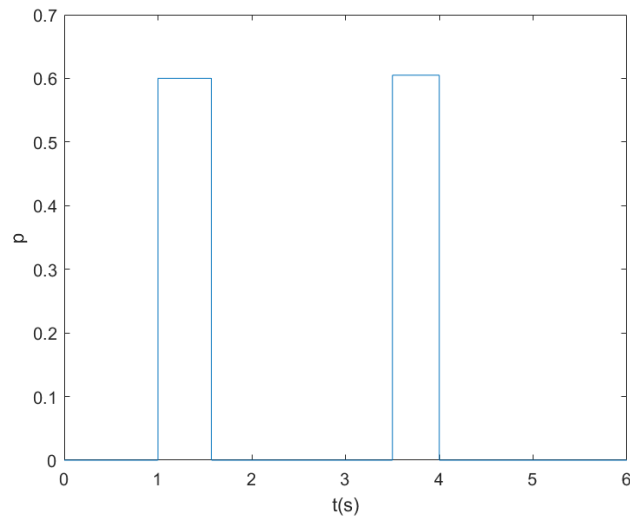


Figure 5.3: perturbation

The simulation results are given as follow :

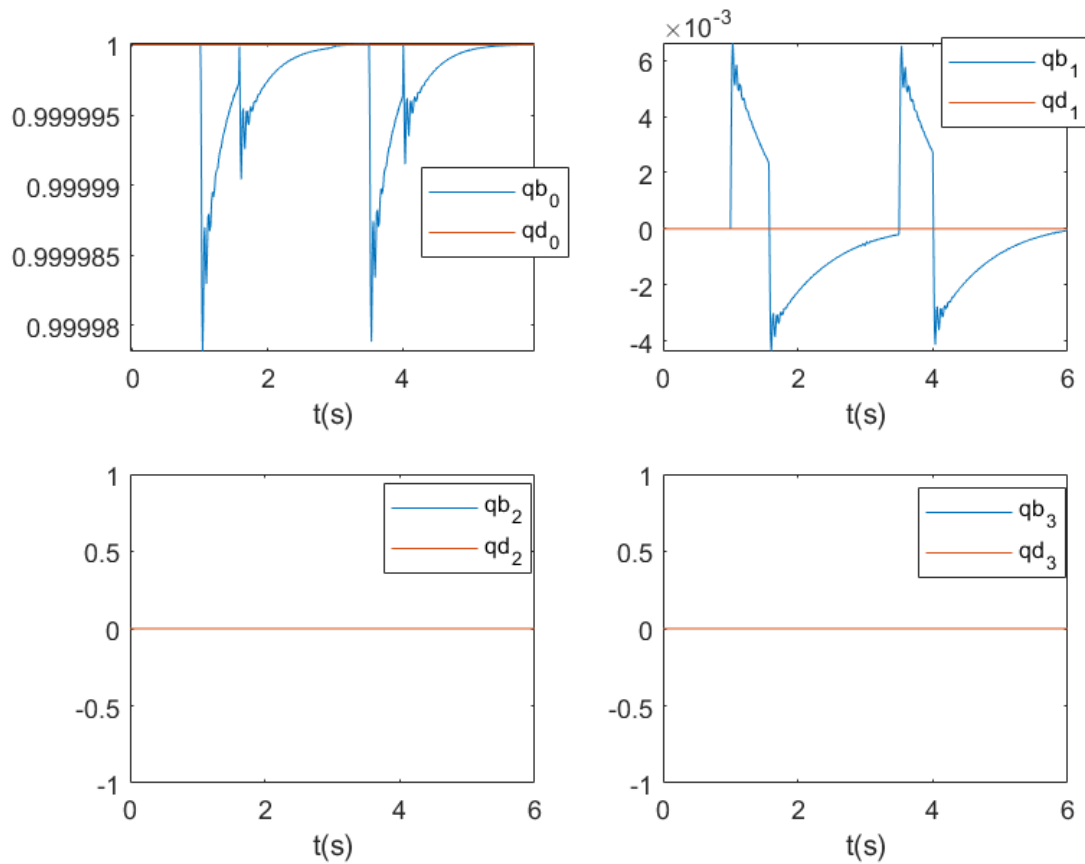


Figure 5.4: quaternions representing the system attitude

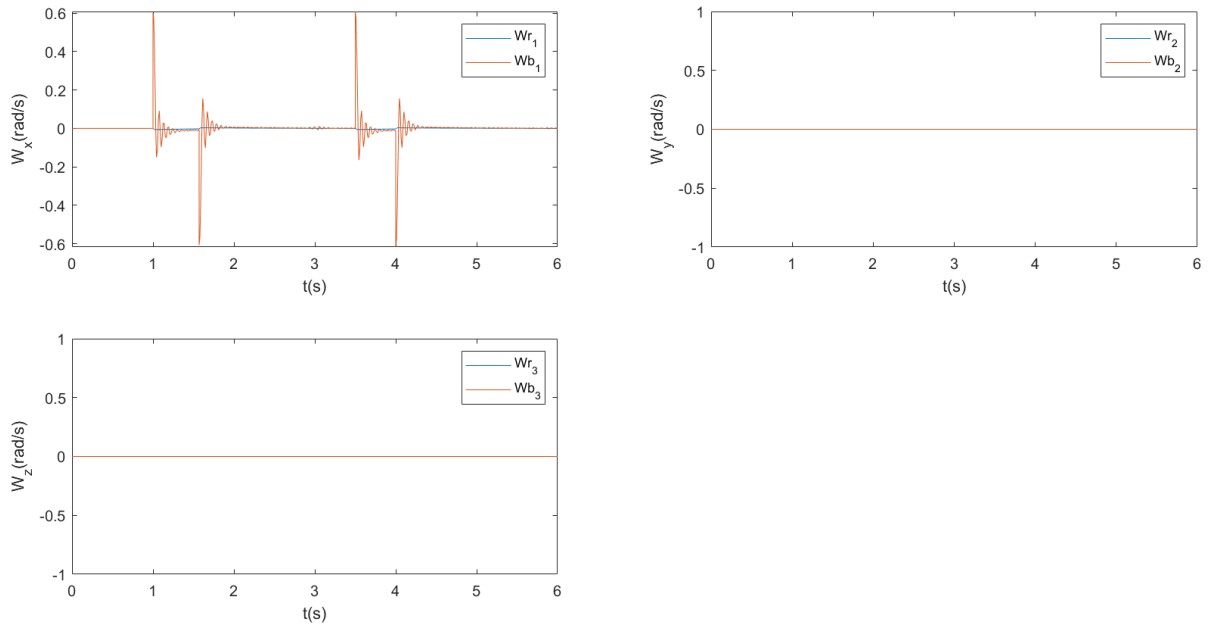


Figure 5.5: The system angular velocity

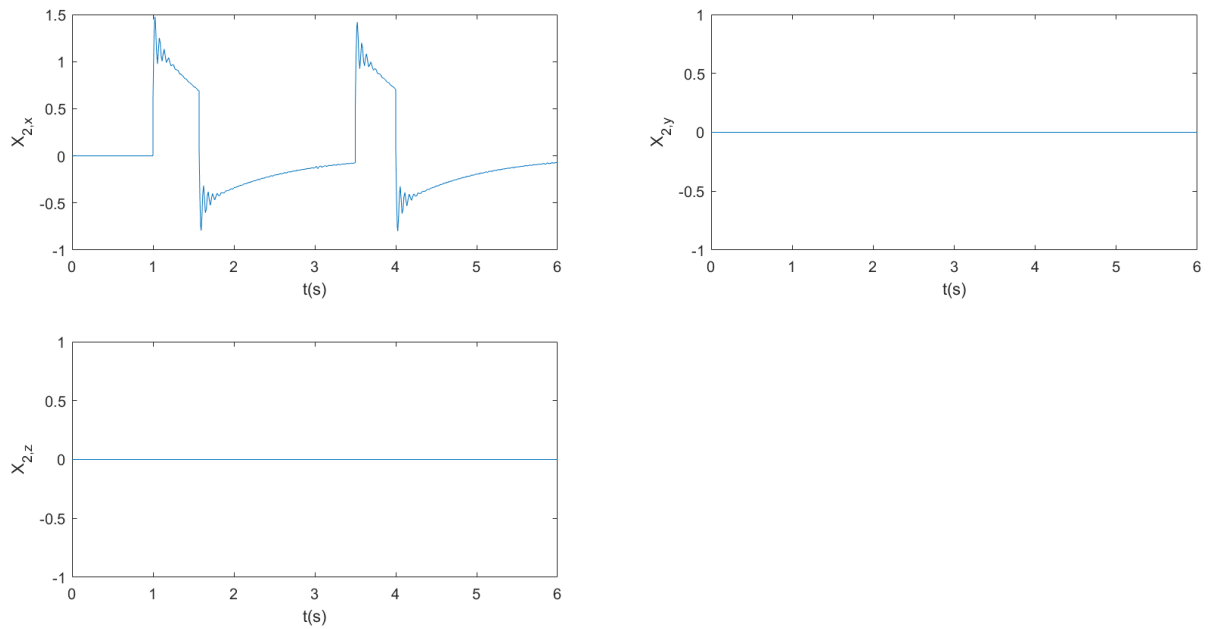


Figure 5.6: The observer output (disturbance estimation)

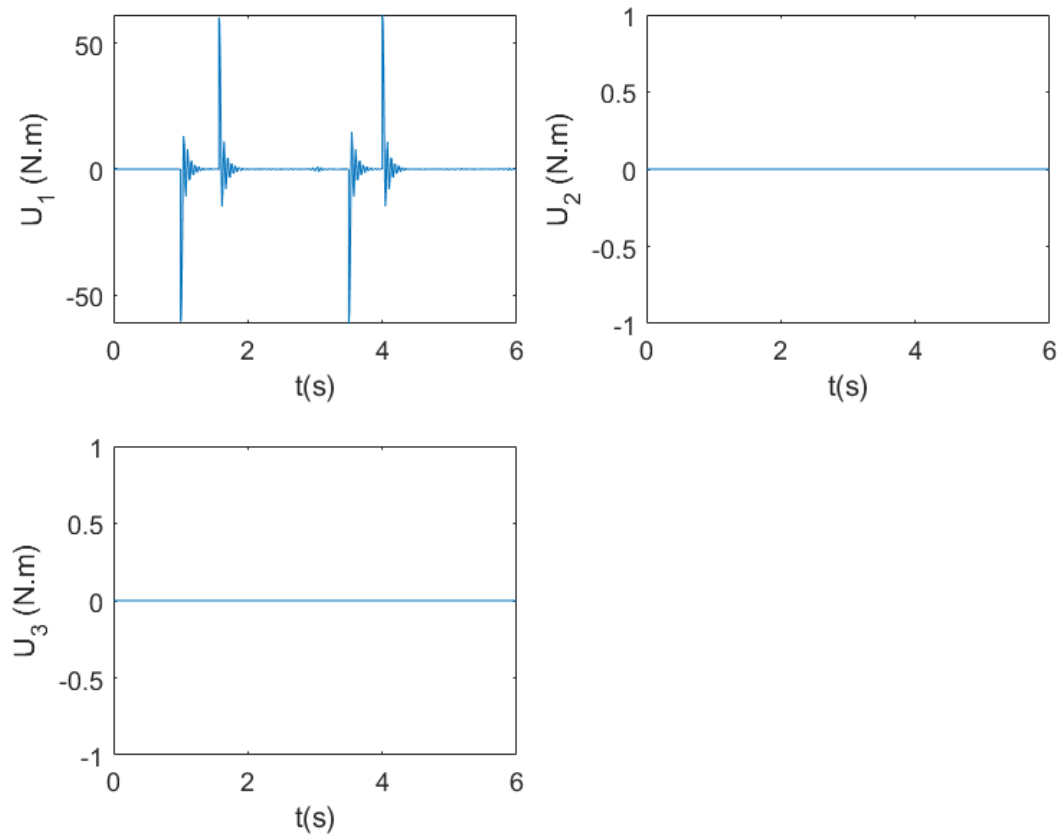


Figure 5.7: torques u generated by the controller

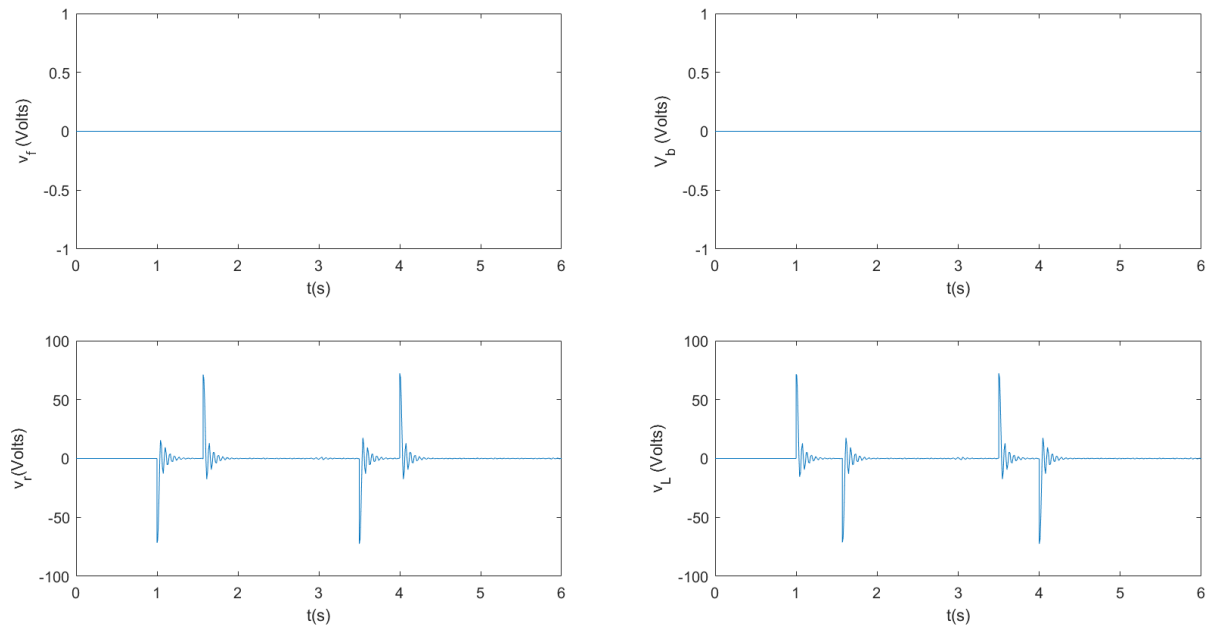


Figure 5.8: voltage delivered to the 4 motors

5.2.2 interpretation

We can note that the system remains stable in spite of the presence of perturbation thing that is reflected in the figure 5.4 where the attitude measured is almost the same as the desired one $q_b \approx q_d = [1 \ 0 \ 0 \ 0]$, the stability of the system is ensured due to the good estimation of the observer, where we can see in the figure 5.6 that the shape of the disturbance is almost the same as the observer has given. The disturbances are compensated in the output feedback control (figure 5.7).

In the figure 5.8 we can see that only the right and left motors that are operated and that two motors are the responsible of the roll stabilisation.

5.3 Application on the Quanser 3 DOF Hover

In this section the ADRC law is applied on the Quanser 3 DOF Hover to test its real world robustness. (It's should be mentioned that the attitude estimation is made by using the Euler angles)

5.3.1 Hovering

To test the stability on the hovering position in the presence of real disturbances , the angular velocity of the desired reference frame is given as $\omega_d = (0 \ 0 \ 0)^T$.

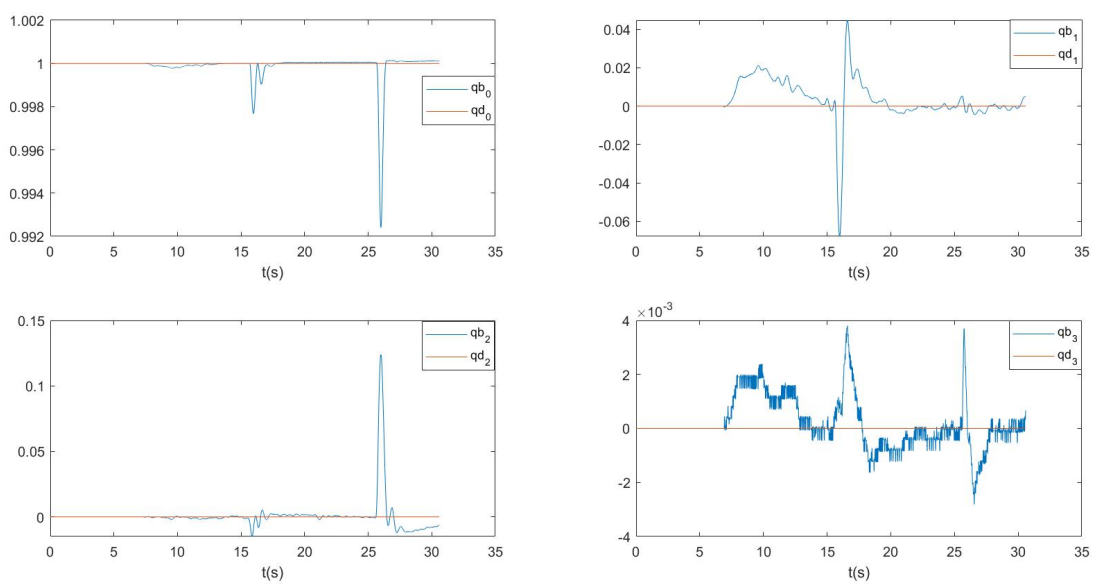


Figure 5.9: quaternions representing the system attitude

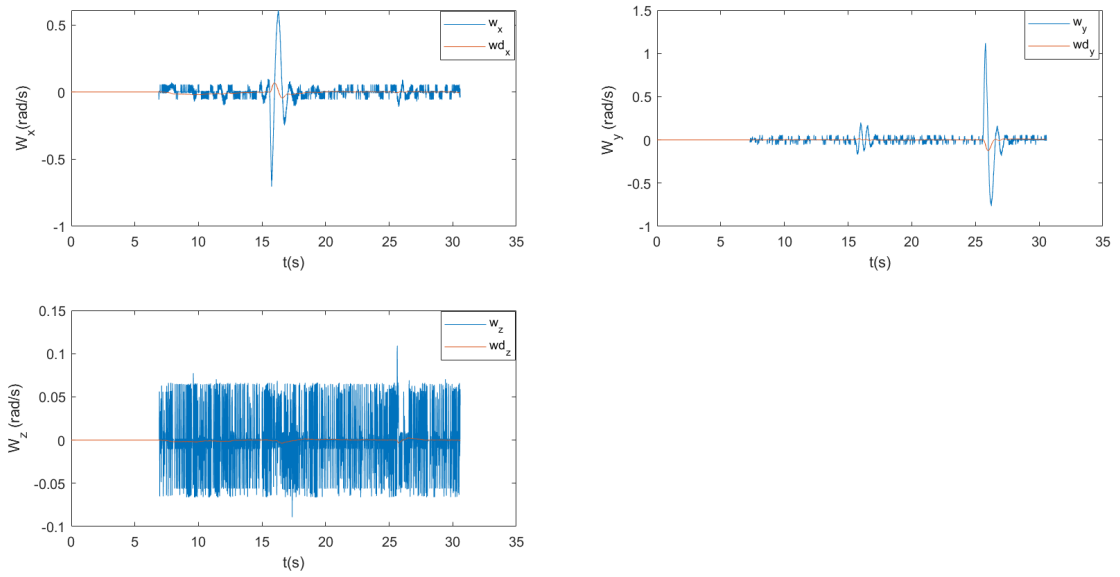


Figure 5.10: The system angular velocity

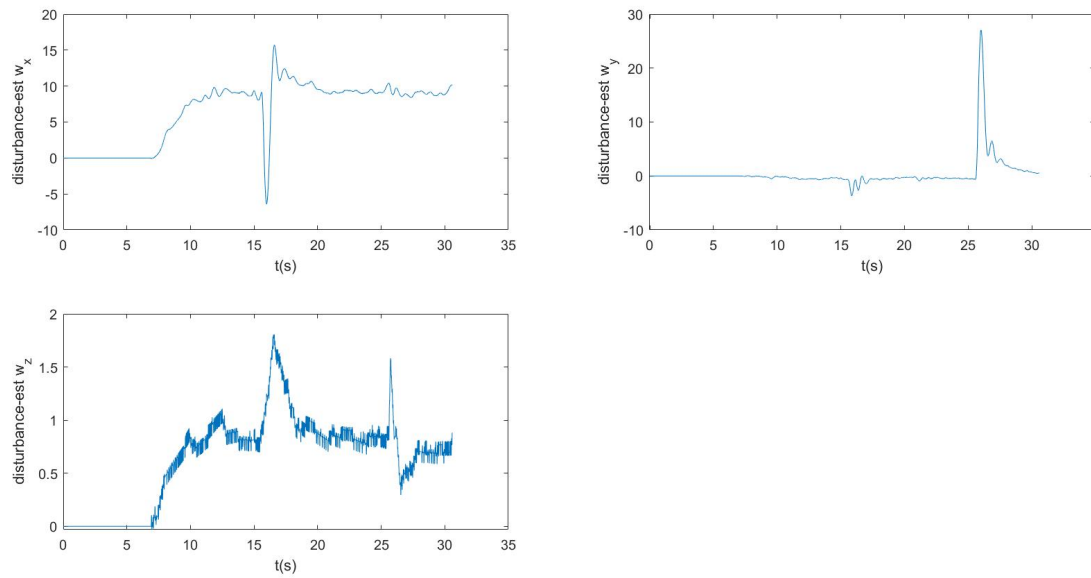


Figure 5.11: The observer output (disturbance estimation)

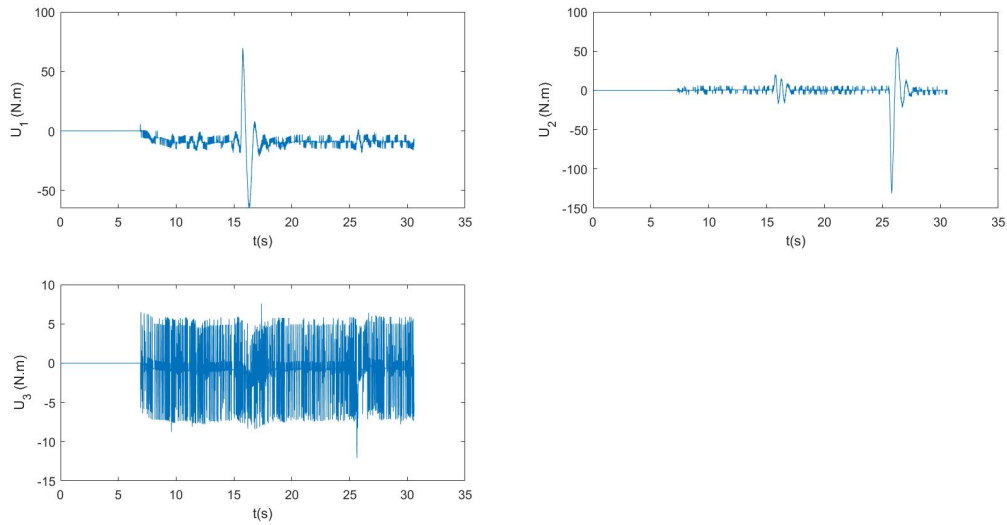


Figure 5.12: torques U generated by the controller

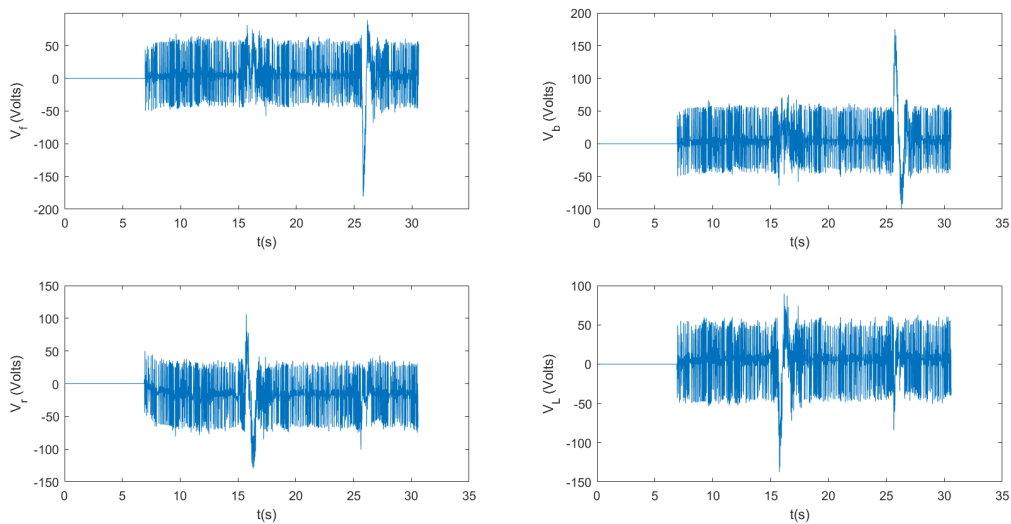


Figure 5.13: Voltage delivered to the 4 motors

5.3.2 interpretation

In figures (5.9, 5.10) we see that our system is identical with the desired trajectory ($\omega_d = (0 \ 0 \ 0)^T$), until a disturbance is included (real disturbance made by hand) and what we notice is that the system returns to the desired path immediately, that what demonstrate the robustness of not only the control law that rejects the disturbances, but also the high performance of the observer error estimation.

5.3.3 Rotation around the z axis (YAW)(Without perturbation)

To test the ability of the control law to make the system track a desired trajectory we chose the rotation around the z axis, the angular velocity of the desired reference frame is given as $\omega_d = (0 \ 0 \ 0.1)^T$ and the desired attitude is given by the quaternion $q_d = [\cos(\theta_d/2) \ 0 \ 0 \ \sin(\theta_d/2)]^T$ where θ_d is the desired rotational angle.

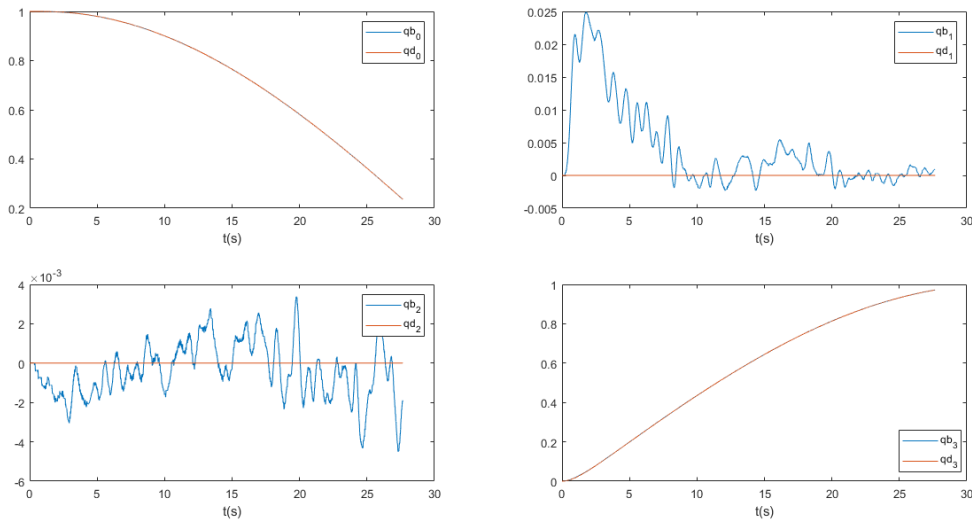


Figure 5.14: quaternions representing the system attitude

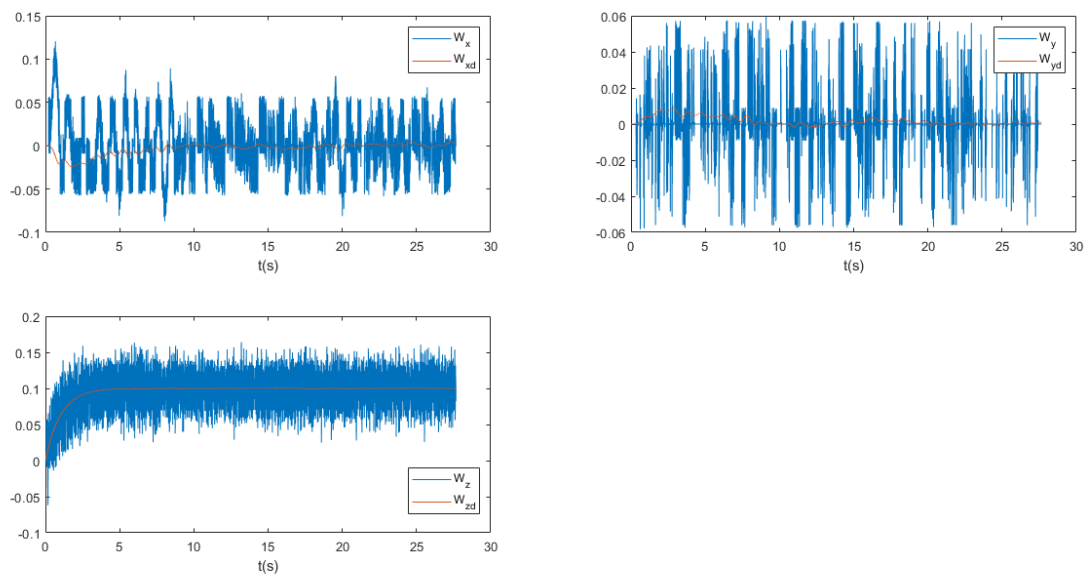


Figure 5.15: The system angular velocity

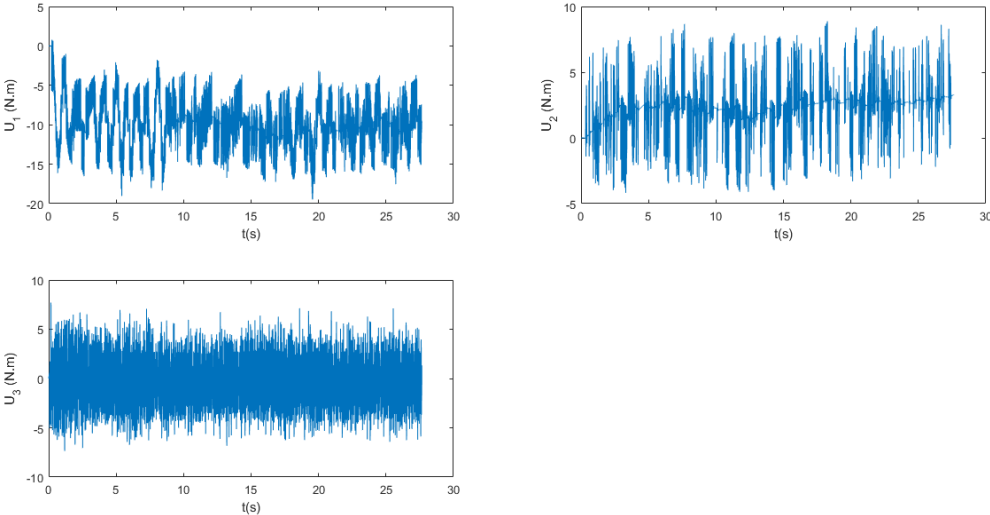


Figure 5.17: torques U generated by the controller

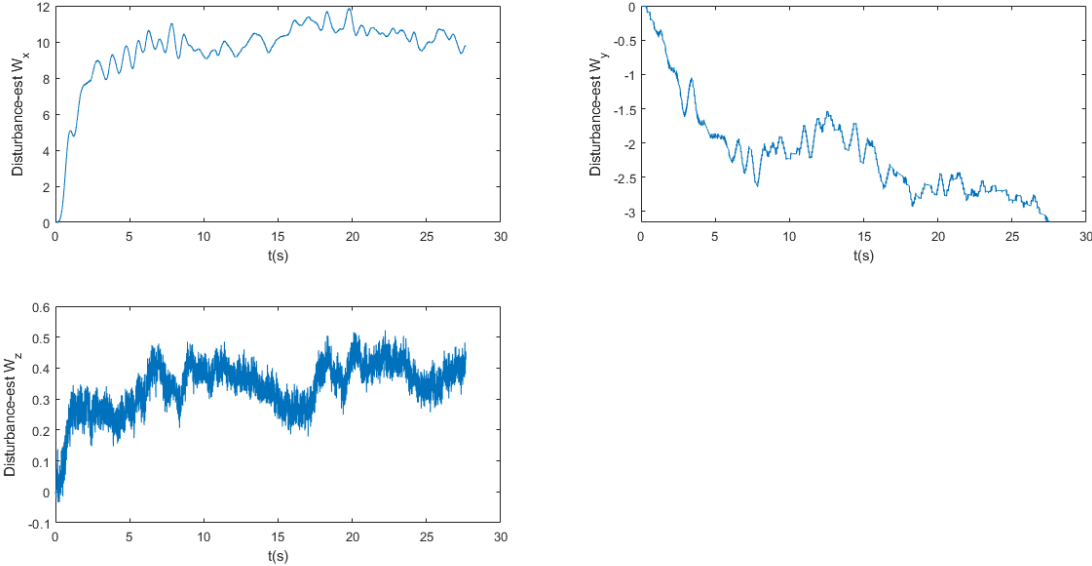


Figure 5.16: The observer output (disturbance estimation)

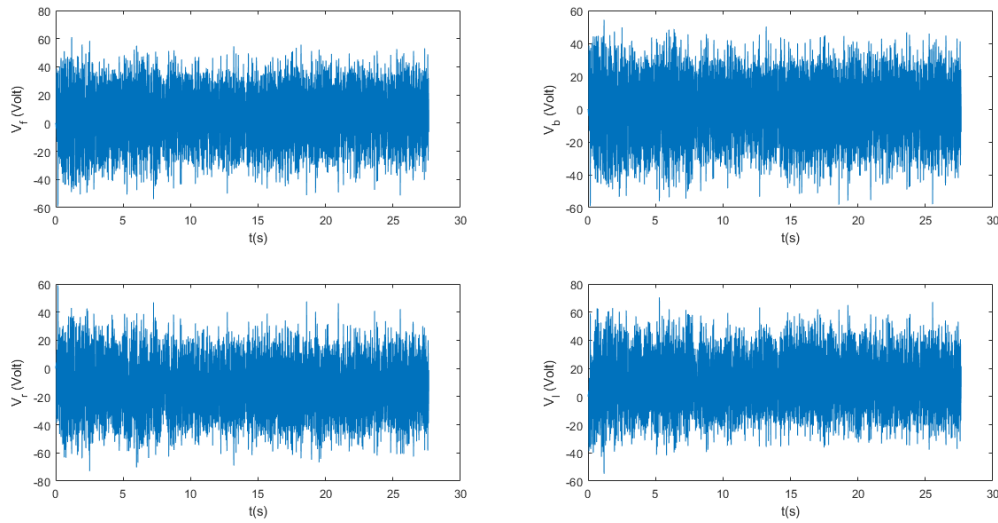


Figure 5.18: Voltage delivered to the 4 motors

5.3.4 interpretation

We note that the system track the desired trajectory where we can see in figure 5.14 the measured attitude quaternion q_b is almost the same as the desired one, by analyzing q_b we mark that the rotational axe is $\vec{e} = [0 \ 0 \ 1]^T$ ($q_b = [\cos(\theta/2) \ \vec{e} \cdot \sin(\theta/2)]^T$), also the desire angular velocity is respected where we can see in figure 5.15 that $w_b = w_d = [0 \ 0 \ 0.1]^T$. in figure (5.16) we see that even with the absence of the disturbances the observer is still showing an estimate of errors and that due to the existence of an offset in the QUANSER's controller, and modeling errors.

5.4 Quaternion attitude estimation

As it's mentioned in the previous section, the attitude estimation requires sensors data (a.k.a Sensor fusion), for that we are using an IMU-6050 micro-chip.

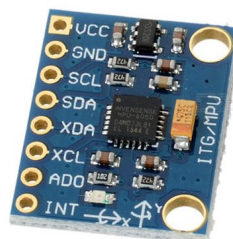


Figure 5.19: IMU-6050 chip

The IMU-6050 is integrated 6-axis MotionTracking device that combines a 3-axis gyroscope, 3-axis accelerometer, and a Digital Motion Processor.

This device needs a control unit to work on, in our case an arduino Nano3 is used.



Figure 5.20: Arduino Nano3 chip

5.4.1 MALAB implementation

To run our experience, we are using MATLAB Arduino package, and Sensor Fusion and Tracking Toolbox. Sensor fusion algorithms used in this experience use North-East-Down(NED) as an Inertial frame coordinate system. In the NED reference frame, the X-axis points north, the Y-axis points east, and the Z-axis points down.

Since our IMU doesn't include a magnetometer, we are using the **imufilter** and **complementaryFilter** System objects; those system objects fuse accelerometer and gyroscope data.

The **imufilter** uses an internal error-state Kalman filter and the **complementaryFilter** uses a complementary filter. The filters are capable of removing the gyroscope's bias noise, which drifts over time.

Listing 5.1: Estimating Orientation code

```

1 a = arduino('COM7', 'Nano3', 'Libraries', 'I2C');
2 fs = 100; % Sample Rate in Hz
3 imu = mpu6050(a,'SampleRate',fs,'OutputFormat','matrix');
4
5 GyroscopeNoiseMPU9250 = 3.0462e-06; % GyroscopeNoise (variance)
   in units of rad/s
6 AccelerometerNoiseMPU9250 = 0.0061; % AccelerometerNoise (
   variance) in units of m/s^2
7 viewer = HelperOrientationViewer('Title',{'IMU Filter'});
8 FUSE = imufilter('SampleRate',imu.SampleRate, 'GyroscopeNoise',
   GyroscopeNoiseMPU9250,'AccelerometerNoise',
   AccelerometerNoiseMPU9250);
9 stopTimer=100;
10
11 tic;
12 while(toc < stopTimer)
13     [accel,gyro] = imu.read;
14     rotators = FUSE(accel,gyro);
15     for j = numel(rotators)
16         viewer(rotators(j));
17     end
18 end

```

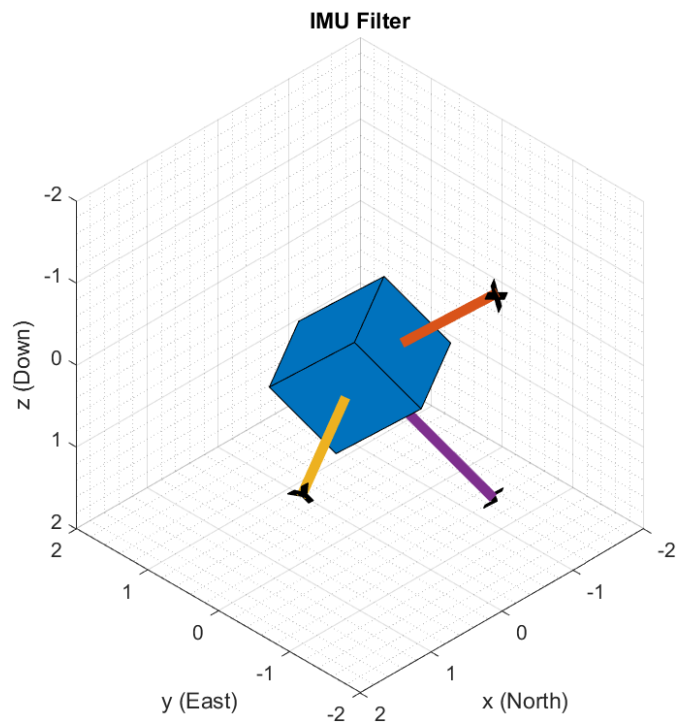


Figure 5.21: A snapshot from the Orientation Viewer

The $FUSE = imu\ filter$ returns fusion of accelerometer and gyroscope data to estimate device orientation. This data is represented in quaternion, therefore we can use them in our control loop by placing the IMU in the center of the Quanser Hover.

```
0.28809 + 0.95071i - 0.09443j - 0.065108k
0.28939 + 0.94981i - 0.098262j - 0.06678k
0.28663 + 0.94995i - 0.10485j - 0.066631k
0.28364 + 0.95013i - 0.11142j - 0.066248k
0.2785 + 0.95094i - 0.11828j - 0.064451k
0.27215 + 0.95212i - 0.1249j - 0.061606k
0.26583 + 0.9532i - 0.13149j - 0.058825k
0.26533 + 0.95297i - 0.13527j - 0.056093k
0.27281 + 0.95139i - 0.13217j - 0.054387k
0.27791 + 0.94988i - 0.13298j - 0.053026k
```

Figure 5.22: Quaternion-based attitude representation

In the figure (5.22), we see a table of quaternions that represents the IMU attitude, we notice that there's an update of this estimation (we can see only the attitude of the last ten samples)

5.5 Conclusion

In this chapter, the ADRC law has been tested in two different ways. By simulation where we used Matlab/SIMULINK, this control law has proved its robustness by rejecting all the introduced disturbances, estimated by ESO observer. In the real world we applied it on the QUANSER 3DOF Hover, and the results were approximately identical with the simulation ones (approximately because there was permanent disturbances which is an offset).

Conclusion and Prospects

Overall conclusion

The main objective of this work was to develop a quadrotor attitude control while taking into account the external disturbance and model errors and uncertainties, also two principal axes were treated, which are the attitude estimation, in particular the solution of Whaba's problem, and the use of quaternions as a solution to the gimbal lock singularity.

To understand the behaviour of the quadcopter and the physical laws that govern such systems, a dynamic model was introduced based on the Newton-Euler formalism. Thus, it is almost impossible to build a complete mathematical model of an aerial vehicle, capturing all the aerodynamic effects, Therefore to achieve desired trajectories and suitable positions it becomes necessary to synthesize a controller that does not depend on the QUAV model, for that purpose we have chosen the active disturbance rejection control, based on this command the unmodelled dynamics and external disturbance will be treated as total disturbance that is estimated using a dedicated observer ESO and compensated in the feedback output allowing to acquire satisfactory performances in terms of stabilization and trajectory tracking.

The quadrotor kinematics is modilized by the quaternion, thing that enables to avoid the famous problem of Euler angle that is the gimble lock singularity where the system loses a degree of freedom, using the quaternion as a rotation operator all possible positions can be reached.

In case of real flight, there is no sensor that can measure the position of quadrotor in space so the problem of attitude estimation is introduced under the name of Wahba problem and solved by many methods, we set Davenport's q method and Kalman filter, as two solution of that problem. These two methods use what is called sensor fusion technique, which is about fusing two, or more, sensors' data to estimate the attitude.

Finally these theoretical results were tested in simulation using Matlab/SIMULINK, where we witnessed the robustness of the ADRC Law; next we applied this control law on the QUANSER 3DOF Hover where we saw performance of this law in real world application.

Prospects

As all work is called to be improved and enriched, ours is no exception. Prospects of evolution can be envisaged in the following ways :

- Using the IMU (with the Kalman filter algorithm) to get the estimation of the QUANSER's attitude.

Conclusion and Prospects

- Exploitation of the data gathered from the IMU, in the control loop.
- In our work we set the setpoint as an angular velocity, another work can set it as a desired position.

Bibliography

- [1] J Apkarian and M Lévis. Laboratory guide: 3 dof hover experiment for matlab. *Simulink Users*, 2013.
- [2] the different types of quadcopter drones today for industrial ... -horusrc. www.horusrc.com/en/blog/type-of-quadcopte-drone/ –Accessed on 12.04.2022.
- [3] Redouane Dargham, Adil Sayouti, and Hicham Medromi. Euler and quaternion parameterization in vtol uav dynamics with test model efficiency. *coordinates*, 2(3):4, 2015.
- [4] Eric Feron and Eric N Johnson. Aerial robotics., 2008.
- [5] S Norouzi Ghazbi, Yasaman Aghli, M Alimohammadi, and Ali Akbar Akbari. Quadrotors unmanned aerial vehicles: A review. *International journal on smart sensing and Intelligent Systems*, 9(1), 2016.
- [6] Jaime Moreno, Jesus Cruz, and Edgar Dominguez. White- donkey: Unmanned aerial vehicle for searching missing people. *International Journal of Advanced Computer Science and Applications*, 7(7):574–581, 2016.
- [7] Rejane Cavalcante Sá, Guilherme A Barreto, André Luiz C de Araújo, and Antonio T Varela. Design and construction of a quadrotor-type unmanned aerial vehicle: Preliminary results. In *2012 Workshop on Engineering Applications*, pages 1–6. IEEE, 2012.
- [8] Omkar Harkare and Rohan Maan. Design and control of a quadcopter. *International Journal of Engineering and Technical Research*, 10:257, 05 2021.
- [9] Rf wireless world. www.rfwireless-world.com/Terminology/Drone-Sensors.html – Accessed on 12.04.2022.
- [10] Jurek Z Sasiadek. Sensor fusion. *Annual Reviews in Control*, 26(2):203–228, 2002.
- [11] Taehwan Cho, Changho Lee, and Sangbang Choi. Multi-sensor fusion with interacting multiple model filter for improved aircraft position accuracy. *Sensors*, 13(4):4122–4137, 2013.
- [12] Tammaso Bresciani. Modelling, identification and control of a quadrotor helicopter. *MSc theses*, 2008.

- [13] Sen Yang, Leiping Xi, Jiaxing Hao, Yuefei Zhao, Ye Yang, and Wenjie Wang. Aerodynamic parameters identification and adaptive ladrc attitude control of quad-rotor model. 2018.
- [14] Bhaskar Dasgupta and Prasun Choudhury. A general strategy based on the newton–euler approach for the dynamic formulation of parallel manipulators. *Mechanism and machine theory*, 34(6):801–824, 1999.
- [15] Karsten Großekatthöfer and Zizung Yoon. Introduction into quaternions for spacecraft attitude representation. *TU Berlin*, 16, 2012.
- [16] Hardik Parwana and Mangal Kothari. Quaternions and attitude representation. *arXiv preprint arXiv:1708.08680*, 2017.
- [17] Gimbal lock, April 2022. Page Version ID: 1082887600.
- [18] Chih-Lyang Hwang, Jui-Yu Lai, and Zih-Siang Lin. Sensor-fused fuzzy variable structure incremental control for partially known nonlinear dynamic systems and application to an outdoor quadrotor. *IEEE/ASME Transactions on Mechatronics*, 25(2):716–727, 2020.
- [19] Baomei Xu, Zhixin Cheng, Rui Zhang, Changhong Gong, and Linhao Huang. Pso optimization of ladrc for the stabilization of a quad-rotor. In *2020 12th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, pages 437–441. IEEE, 2020.
- [20] Manmohan Sharma and Indrani Kar. Nonlinear disturbance observer based geometric control of quadrotors. *Asian Journal of Control*, 23(4):1936–1951, 2021.
- [21] Yuan Wang, Hongming Cai, Junmiao Zhang, and Xubo Li. Disturbance attenuation predictive optimal control for quad-rotor transporting unknown varying payload. *IEEE Access*, 8:44671–44686, 2020.
- [22] Qi Lu, Beibei Ren, and Siva Parameswaran. Uncertainty and disturbance estimator-based global trajectory tracking control for a quadrotor. *IEEE/ASME Transactions on Mechatronics*, 25(3):1519–1530, 2020.
- [23] Kai Wang, Changchun Hua, Jiannan Chen, and Manjun Cai. Dual-loop integral sliding mode control for robust trajectory tracking of a quadrotor. *International Journal of Systems Science*, 51(2):203–216, 2020.
- [24] Erdal Kayacan and Reinaldo Maslim. Type-2 fuzzy logic trajectory tracking control of quadrotor vtol aircraft with elliptic membership functions. *IEEE/ASME Transactions on Mechatronics*, 22(1):339–348, 2016.
- [25] Dam B Erik, Koch Martin, and Lillholm Martin. Quaternions, interpolation and animation. *Technical Report DIKU-TR-98/5*, 1998.
- [26] Jack B Kuipers. *Quaternions and rotation sequences: a primer with applications to orbits, aerospace, and virtual reality*. Princeton university press, 1999.

- [27] Emil Fresk and George Nikolakopoulos. Full quaternion based attitude control for a quadrotor. In *2013 European control conference (ECC)*, pages 3864–3869. IEEE, 2013.
- [28] Sandeep Gupta. Linear quaternion equations with application to spacecraft attitude propagation. In *1998 IEEE Aerospace Conference Proceedings (Cat. No. 98TH8339)*, volume 1, pages 69–76. IEEE, 1998.
- [29] Yan-Bin Jia. Quaternions. *Course Com S*, 477:577, 2019.
- [30] Jossué Carino, Hernan Abaunza, and P Castillo. Quadrotor quaternion control. In *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 825–831. IEEE, 2015.
- [31] DM Henderson. Shuttle program. euler angles, quaternions, and transformation matrices working relationships. Technical report, 1977.
- [32] Tammaso Bresciani. Modelling, identification and control of a quadrotor helicopter. *MSc theses*, 2008.
- [33] Abdelhamid Tayebi and Stephen McGilvray. Attitude stabilization of a vtol quadrotor aircraft. *IEEE Transactions on control systems technology*, 14(3):562–571, 2006.
- [34] Taki Eddine Lechekhab, Stojadin Manojlovic, Momir Stankovic, Rafal Madonski, and Slobodan Simic. Robust error-based active disturbance rejection control of a quadrotor. *Aircraft Engineering and Aerospace Technology*, 2020.
- [35] Mokhtari Mohammed Rida. Observation et commande de drones miniatures à voilures tournantes. *Université Aboubekr Belkaid Tlemcen Faculté de Technologie*, le, 2015.
- [36] Alberto Castillo, Ricardo Sanz, Pedro Garcia, and Pedro Albertos. A quaternion-based and active disturbance rejection attitude control for quadrotor. In *2016 IEEE International Conference on Information and Automation (ICIA)*, pages 240–245. IEEE, 2016.
- [37] Hongyiping Feng and Bao-Zhu Guo. Active disturbance rejection control: Old and new results. *Annual Reviews in Control*, 44:238–248, 2017.
- [38] Bao-Zhu Guo and Zhi-Liang Zhao. On convergence of tracking differentiator. *International Journal of Control*, 84(4):693–701, 2011.
- [39] Jingqing Han. From pid to active disturbance rejection control. *IEEE transactions on Industrial Electronics*, 56(3):900–906, 2009.
- [40] Amine Abadi, Adnen El Amraoui, Hassen Mekki, and Nacim Ramdani. Robust tracking control of quadrotor based on flatness and active disturbance rejection control. *IET Control Theory & Applications*, 14(8):1057–1068, 2020.
- [41] Xiaoxia Yang and Yi Huang. Capabilities of extended state observer for estimating uncertainties. In *2009 American control conference*, pages 3700–3705. IEEE, 2009.

- [42] Wenchao Xue and Yi Huang. On performance analysis of adrc for nonlinear uncertain systems with unknown dynamics and discontinuous disturbances. In *Proceedings of the 32nd chinese control conference*, pages 1102–1107. IEEE, 2013.
- [43] Yi Huang and Wenchao Xue. Active disturbance rejection control: Methodology and theoretical analysis. *ISA transactions*, 53(4):963–976, 2014.
- [44] Mark L Psiaki and Joanna C Hinks. Numerical solution of a generalized wahba problem for a spinning spacecraft. *Journal of Guidance, Control, and Dynamics*, 35(3):764–773, 2012.
- [45] Malcolm D Shuster. The generalized wahba problem. *The Journal of the Astronautical Sciences*, 54(2):245–259, 2006.
- [46] F Landis Markley and Daniele Mortari. Quaternion attitude estimation using vector observations. *The Journal of the Astronautical Sciences*, 48(2):359–380, 2000.
- [47] Grace Wahba. A least squares estimate of satellite attitude. *SIAM review*, 7(3):409–409, 1965.
- [48] Eric Rangel Enger. Spacecraft attitude determination methods in an educational context, 2019.

